Recommender systems for e-shops

"We are leaving the age of information and entering the age of recommendation"
Chris Anderson

Business Mathematics and Informatics paper
Manisha Hiralall (1663100)
Vrije Universiteit, Amsterdam
Faculty of Sciences
Supervisor: Wojtek Kowalczyk
2011

Abstract

Selling a wide variety of products has become easier since the coming of online stores, online retailers are able to sell more products than a physical store. The disadvantage is that the customer is not able to find products anymore, because they have to browse in many different categories and sub-categories in order to find the products they are looking for. That is why the recommender systems are increasingly used on e-commerce websites. It learns from the knowledge about the customers and products and gives suitable personalized recommendations to every single customer.
Helping customers to find products easier will increase the loyalty of the customers, this is very important since the competition is just one click away. Because the recommender system encourages customers to buy products they did not plan to buy, the system provides a strategic advantage over businesses without them. And the recommender system helps to build a “value-added relationship” between the website and the user, because the more a user uses a website and purchases items, the more the recommender system learns about the user and the better the recommendations get.
Although recommender systems help discovering products, they do have some advantages that have to be considered. Some examples are the lack of personalization, inaccurate recommendations or no recommendations at all when there is not enough information about the customer or about the product. In this paper I will look at six different approaches that can be used on different e-commerce websites.

1. Non-personalized
2. Demographic-based
3. Collaborative filtering
4. Content-based
5. Knowledge-based
6. Hybrid

For every approach I will state the data source and the advantages and disadvantages.
Lastly I will look at four websites that give recommendations to their customers and try to figure out what data and approach they use to give recommendations:

- Tripbase.com, a travel recommendation website
- Jinni.com, a movie recommendation website
- Wehkamp.nl, online department stores were recommendations are given
- Mybuys.com, a provider of recommender systems

The main question in this paper is what approach will give the best recommendations for an e-shop. My answer to this question is that it depends on the concrete use case. No matter how clever an approach is, there is no single approach that will give the best recommendations for every retailer, because online retailers and shoppers vary way too much. The retailer has to investigate their business and understand what data they can or want to retrieve about the customer and/or products. According to this information the online retailer will be able to find a suitable approach for giving proper recommendations.

Preface

Writing a paper that emphasizes the business-oriented, mathematics and informatics component of the study is a required part of the Masters program of Business Mathematics and Informatics.
During my study I followed the course Business Intelligence and became interested in recommender systems. While shopping online I noticed that the high amount of product offers discouraged me from purchasing any products. I do not want to waste time searching on the web for the products I want to purchase. Wouldn't it be a lot easier if the web knew what I would like to buy? In this paper I will research different methods to give recommendations for different kinds of e-commerce businesses.

I want to thank Dr. Frans Feldberg for helping me find a subject for my paper. And I would like to thank Dr. Wojtek Kowalczyk who has supported me during the process of writing the paper.

Manisha Hiralall
Amsterdam 2011

Contents

1 Introduction

Since the coming of the World Wide Web, in 1991, a lot of terabytes of information has gone online. More than 1.9 billion people around the world now have access to this global information resource[1]. With this development you can imagine that the difficulty of finding relevant information is rapidly increasing. Internet users are constantly presented with situations in which they have many options to choose from, they need assistance in exploring or winnowing down the possibilities. When a user enters some keywords into a search engine, the engine provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text[2]. These search engines are not good enough for winnowing down the possibilities, because there still may be thousands of relevant results too pick from. And users do not always know in advance where he or she is looking for and might not be able to specify the query he needs. For example you want to watch a movie, but you do not know how to specify this in a query.
A system is needed that supports the user in finding and selecting products, services or information when there are too many items to consider or the user has a lack of knowledge about the topic or domain. This system is called a recommender system; it uses knowledge about the user and products to give suitable recommendations. This system has many approaches and every approach uses a different data source.

The main question of this paper is:

Find a recommender system approach that gives the best recommendations for an arbitrary e-commerce website.

To find the best approach I will investigate the following:

1. What data can be obtained for the recommender system?
2. What recommender system approaches exists?
3. What are the advantages and disadvantages of the different recommender system approaches?

The paper starts in chapter 2 with a definition of the recommender system and why recommender systems are useful for a customer and for the online retailer.
Chapter 3 contains the different data sources that are needed for the recommender systems. Chapter 4 describes the different approaches of the recommender systems with the data sources and the advantages and disadvantages of the approach. In chapter 5 you can find some examples of e-commerce websites that use recommender systems. Finally in chapter 6 I will give an answer to the main question of this paper.

[1] http://www.internetworldstats.com/stats.htm
[2] http://computer.howstuffworks.com/internet/basics/search-engine.htm

2 Recommender systems in E-commerce

2.1 Definition

It is obvious online shoppers are discouraged from buying products online when they have to browse in many different categories and sub-categories of an e-store in order to find the products they are looking for. To overcome this obstacle, recommender systems are created. The system attempts to filter products of an e-store according to the preference of the customer. The difference with traditional search engines is the need for a keyword input. The recommender system is able to present matching products without the obligation to type any keywords. The system bases the presented products on a profile of the customer and therefore can enhance the product discovery.
The most popular example of a recommender system is the e-store Amazon.com, where book recommendations are given. This system takes the buying behavior, opinions and tastes of a large community of users into account and thus constitutes a social or collaborative recommendation approach. In contrast, content-based approaches rely on product features and textual item descriptions; the demographic- and knowledge-based approach, finally, generate product recommendations based on explicit knowledge of the customer.

There are different types of websites that are using these recommender systems:

- Content sites: Last.fm, stumbleUpon
- Advertisement: google adsense, doubleclick
- E-commerce sites: Amazon, Netflix

In figure 2.1 you can find an example of a personal recommendation on Amazon.com.

Figure 2.1 - Amazon recommendations

In this paper the focus will be only on recommendations for e-commerce websites like Amazon and Netflix.

2.2 Why vendors need recommender system

In [1] Schafer et al. discussed three ways how recommender systems enhance e-shop sales.
Good recommender systems present customers products they likely are interested in but did not plan to buy, making them purchase more items. These unplanned purchases are not yet happening as often in online stores as in traditional stores.
Recommender engines can help to gain consumers' loyalty, which is an essential business strategy in e-commerce as the competitor is always just "one click away". Because the recommender system makes it easier and faster to find new products, customers will come back more often.
The more a user uses a website and purchases items, the more the recommender system learns about the user and the better the recommendations get. This helps to build a "value-added relationship" between the website and the user. Recommender systems are also a way to promote older or low-demand items, such as niche products.

2.3 Recommendation process

In general, every recommendation system follows a specific process in order to produce product recommendations, see figure 2.2. The recommendation approaches can be classified based on the information sources they use. Three possible sources of information can be identified as input for the recommendation process. The available sources are the user data (demographics), the item data (keywords, genres) and the user-item ratings (obtained by transaction data, explicit ratings).These sources will be discussed in the next chapter. The recommender system approaches will be discussed in chapter 4.

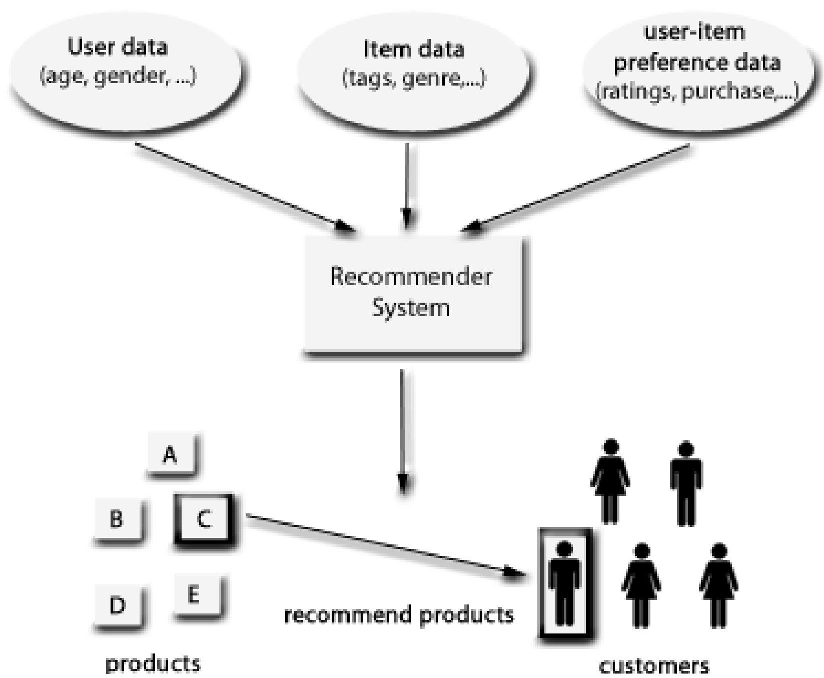

Figure 2.2 – Recommendation process

3 Data

As discussed in the previous chapter, the recommender systems would not be functional without data. Some recommender systems need data about the user, some need data about the products and some recommender systems need both. Data can be provided by the customer explicitly or implicitly. There are many ways to acquire data, which will be discussed here.

3.1 Explicit user data

Explicit data is given by a customer. A rating can be given on a particular scale, this is often a scale of five stars, where one star represents the lowest ranking and five stars the highest ranking. Scales of more than five stars are possible as well, see figure 3.1. Amazon.com uses the following meaning to the stars: **1 star:** I hate it; **2 stars:** I don't like it; **3 stars:** It's OK; **4 stars:** I like it; **5 stars:** I love it

Figure 3.1 - Star rating

Figure 3.2 - Thumbs up/Thumbs down

Alternatively, the user can give thumbs up if he/she liked an item or thumbs down for not liking the item (figure 3.2). All these explicit ratings can be stored in a user-item matrix, see table 3.1.

item / user	**Item 1**	**Item 2**	**Item 3**	**..**	**Item j**		**Item m**
User 1	$R_{1,1}$	$R_{1,2}$	$R_{1,3}$	..	$R_{1,j}$	..	$R_{1,m}$
User 2	$R_{2,1}$	$R_{2,2}$	$R_{2,3}$	..	$R_{2,j}$	..	$R_{2,m}$
User 3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$	..	$R_{3,j}$	..	$R_{3,m}$
..	..	..	..	..	..	..	..
User k	$R_{k,1}$	$R_{k,2}$	$R_{k,3}$	..	$R_{k,j}$	..	$R_{k,m}$
..	..	..	..	..	..	..	..
User n	$R_{n,1}$	$R_{n,1}$	$R_{n,1}$	..	$R_{n,j}$	..	$R_{n,m}$

Table 3.1 - User-item rating matrix

Where u*ser 1, user 2, .., user n* are the *n* users that use the particular e-shop. *Item 1, item 2,... item m* are the *m* items that can be rated by the users. And $R_{k,j}$ represent the rating of user *k* for item j.

There are a lot of other methods to obtain explicit ratings, like asking the user to pick the most favorite items from a list of items [2].

3.2 Implicit user data

Not all users rate all the items they have bought or viewed, because they just do not want to spend their time rating the items or do not see the point of doing so. And not all customers register to the shop and want to give all their personal information. For these users another information source is needed to

overcome the lack of ratings. One approach to this problem is to use implicit ratings: watching the behavior of the user.
Nichols et al. produced a table in [3] that shows the usage data for a Digital Library. The usage data also applies for any arbitrary e-shop, see table 3.2. The actions are listed in an approximate ordering reflecting the importance of the type of data; a purchase of an item says more than a simple inspection. Some of the data sources have additional information, for example *Purchase* has a price and *Repeated Use* has a number.

Action	**Example**
Purchase (Price)	buys item
Assess	evaluates or recommends item
Repeated Use (Number)	buying item multiple times
Save / Print	saves item to personal storage (e.g. product image/ information page)
Delete	deletes an item (e.g. from favorites)
Refer	cites or otherwise refers to item
Reply (Time)	replies to item
Mark	add to a 'marked' or 'interesting' list
Examine / Read (Time)	looks at whole item
Consider (Time)	looks at description
Glimpse	sees title
Associate	returns in search but never glimpses
Query	association of terms from queries

Table 3.2 - Potential types of implicit rating information

With collected *Purchase* information the purchase pattern of a user can be obtained. For example with the gathered data about the types and combination of goods bought by a customer, a pattern can be created. A pattern can be an association, e.g. a customer who buys a camera also buys a memory card. When using this association rule, if a customer buys a camera, the system should give memory cards a high priority. The user-item matrix would look the same as table 3.1 except the ratings are replaced with binary numbers, 1 for a purchase and a 0 for a non-purchase.
It is also possible to gather implicit data from an explicit rating scenario. The *Assess* category distinguishes those events when an evaluator chooses *not* to rate an item when they could have done so. Hence this category would not contain any reference to the actual value of a rating only the fact that a rating had, or had not occurred.
The Repeated Use category can be described as follows: when items are bought by a lot of users, the item is apparently popular and the rating of the item can be considered to be high.
Items that a user wishes to preserve for some purpose are often *Saved* to personal file space or *Printed*. For example a product picture can be saved to personal file space or a product information page can be printed.
The *Delete* category differs from the others in that it expresses a negative judgement. This may occur when a user deletes a product from for example a favorites list or wish list. This might suggest that the user is not interested in the item anymore
When a user likes an item, he can *Refer* to it by, for example send an email to a friend with the item or posting it on a blog or forum with the hyperlink of the items page.
E-shops might support some interactive environments where users can *Reply* to items they encounter, either back to the sender or via a public forum. The *Time* taken to compose this reply may also be available.

In many environments a user will *Mark* certain items as being of particular interest so that they can easily return to them, e.g. Web browsers enable hotlists or bookmarks to be recorded.
The next three categories, *Examine*, *Consider* and *Glimpse*, all refer to the same action: the user reading a document (or document surrogate). Systems usually allow users to read a short description of an item. E-shops often provide their items in a list with a short description or title, the selection of an individual item for further examination provides the first clue about a user's interest.
At the bottom of the list the action *Associate* refers to items which are closely connected to those that are examined.
The action *Query* refers to query terms which have been used by searchers and can then be reused by subsequent searchers who use related terms.
Some of these types of data, such as purchasing, repeated use and explicit referencing are possible to record in physical systems, even then it may be exorbitantly expensive. Most actions, such as examining, considering or querying are not. They can however be captured by transaction logging systems, this has been discussed by Borgman et al. in [4].

3.3 Demographical data

Demographical data can be obtained explicitly and implicitly. Based on the demographical data a user profile can be created. With a demographic profile, certain matching items can be recommended. Data such as *age, gender, social class, education, location*, etc. can be used. For instance teenagers prefer different products than the elderly and the rich may want different products than middle and lower classes and are willing to pay more. In most web shops these different user demographics are asked during the registration or they can be captured by transaction logging systems [4]. See table 3.3 for an example of how the data matrix looks like.

demographic / user	age	gender	location	Etc.
User 1	22	M	Amsterdam	
User 2	32	M	Barcelona	
User 3	15	F	Antwerp	
..	..			
User k	23	F	Seoul	
..				
User n	45	M	Rome	

Table 3.3 - demographic data

3.4 Product data

Data about products is the most commonly used in a recommender system. This information is easy to get as products are mostly provided with all sorts of data, tags or features. In the case of movies the genres like science-fiction, action or comedy are examples of product data. The data has to be provided by the e-shop owner or the users can provide the product properties by giving suitable tags to the products. Or the data can be extracted from documents using an extracting tool. See table 3.4 for an example of the product matrix of a movie.

feature / product	genre	Year	director	etc
movie 1	Science-fiction	2001	James Cameron	
movie 2	Comedy	1998	Edgar Wright	
movie 3	Thriller	1990	Christopher Nolan	
..	..			
movie k	Action	1991	Gore Verbinski	
..				
movie n	Adventure	2009	Peter Jackson	

Table 3.4 - product data matrix

4 Different approaches of recommender systems

The data captured is used by the recommender system to eventually provide the recommendations to the customer. Recommender systems can be present in all sorts of systems and situations, and thus can be implemented in many different ways. In this paper six categories of recommendation approaches will be investigated:

1. Non-personalized
2. Demographic-based
3. Collaborative filtering
4. Content-based
5. Knowledge-based
6. Hybrid

4.1 Non-personalized

The first most simple recommendation approach is the non-personalized one. The recommendations are identical for each customer. The recommendations are either manually selected by the online retailer, based on the popularity of items (average ratings, sales data, total visits, see figure 4.1) or the recommendations can be the top-N new products of the e-shop.

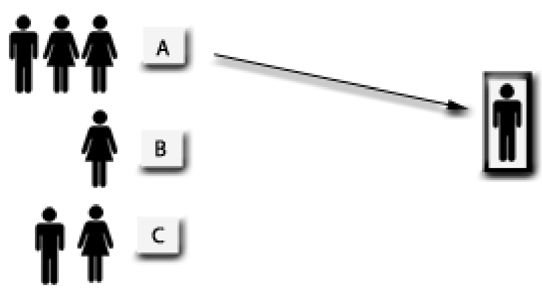

Figure 4.1 - Non-personalized recommendation based on popularity

Advantages and disadvantages
The advantage of this method is that it is easy to realize for the shop owner, the data is easy to collect. However, the recommendations in this system are the same to all users and lack personalization and might not appeal to everyone [5].

4.2 Demographic-based

In [6] Pazzani researched the demographic-based recommendation approach. Demographic data can be used to identify the type of users that like a certain object and create "stereotypes". For example, Table 4.1 shows information on the age, gender, education, etc. of people that rated a certain restaurant together with their rating of the restaurant. With this data one might learn the type of person that likes a certain restaurant, or any kind of product.

	gender	age	education	employed	Restaurant A
Alan	M	15	HS	F	+
Julie	F	17	HS	F	-
Steve	M	35	C	T	+
Emma	F	10	E	F	?

Table 4.1 - demographic data

Demographic recommender systems aim to categorize the user based on personal attributes and make recommendations based on demographic classes, see figure 4.2 for a simple example. This approach can be used for online retailer who sell event tickets or for a restaurant website.

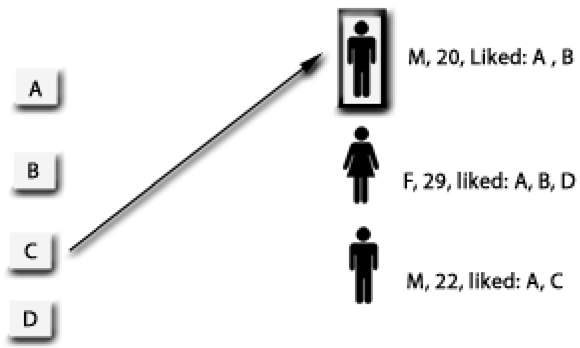

Figure 4.2 - Demographic-based recommendation based on popularity

Advantages and disadvantages
The advantage of a demographic approach is that the user-item ratings are not used, so new users can get recommendations before they have rated any item. And knowledge about the items and their features is not needed; therefore the technique is domain independent.
The disadvantage of the demographic approach is that gathering the required demographic data leads to privacy issues [7]. Demographic classification is also too crude for highly personalized recommendations. For instance not all 20-year old females who are employed enjoy the same movies [8]. And customers with different opinions or an unusual taste result in low correlation coefficient with other customers. Recommendations for them are very difficult to find and they also cause odd recommendations for their correlated users, this problem is called the *grey sheep* problem and is discussed in [9]. Another challenge is the difficulty to change a created profile of a customer once the taste of the customer changes. This is called the stability vs. plasticity problem [10].

4.3 Collaborative filtering

The Collaborative Filtering (CF) approach is widely used in recommender systems. *Filtering* stands for filtering of information, selecting the right information from a big collection. *Collaborative* covers the fact that the information that is being used to filter the collection is being supplied by all the users of the system, see figure 4.3 for a simple example of the approach. The active user likes A and B, the approach then compares the preferences of the active users to the other users and finds a similar user. In this case that is the second user that likes A, B and D, because they both like A and B. The second user likes D as well, so it is likely that the active user will like it to.

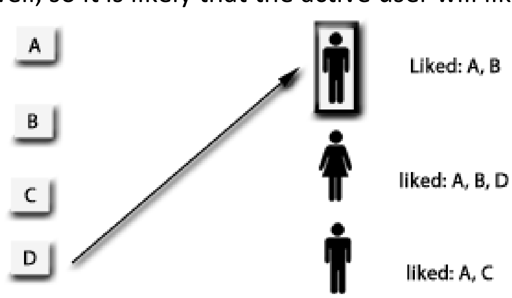

Figure 4.3 - Collaborative filtering approach

The collaborative filtering approach can be divided in two categories[3]:

1. memory-based
2. model-based

4.3.1 Memory-based

Memory-based Collaborative Filtering algorithms use the entire or a sample of the user-item rating matrix to generate a prediction [11]. Every user is part of a group of people with similar interests. By identifying the so-called neighbors of an active user, a prediction of preferences on items for him or her can be produced.

In the *neighborhood-based algorithm (User-based Collaborative Filtering)*, a subset of users are chosen based on their similarity to the active user, and a weighted combination of their ratings is used to produce predictions for this user. Most of these approaches can be generalized by the algorithm summarized in the following steps

1. Assign a weight to all users with respect to similarity with the active user.
2. Select *k* users that have the highest similarity with the active user (neighbors)
3. Compute a prediction from a weighted combination of the selected neighbors' ratings. The *s*imilarity computation.

Step 1 is a critical step in memory-based CF algorithms. The weight $w_{a,u}$ is a measure of similarity between the user *u* and the active user *a*. There are many different methods to compute similarity between users. The most commonly used measure of similarity is the Pearson correlation coefficient between the ratings of the two users:

$$w_{a,u} = \frac{\sum_{i\in I}(r_{a,i}-\bar{r}_a)(r_{u,i}-\bar{r}_u)}{\sqrt{\sum_{i\in I}(r_{a,i}-\bar{r}_a)^2\sum_{i\in I}(r_{u,i}-\bar{r}_u)^2}}$$

Where *I* is the set of items rated by both users, $r_{a,i}$ is the rating given to item i by the active user a, $r_{u,i}$ is the rating given to item *i* by user *u*, and $\bar{r}_u$ is the mean rating given by user.

In step 3 predictions are generally computed as the weighted average of deviations from the neighbor's mean, as in:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u\in K}(r_{u,i}-\bar{r}_u)\times w_{a,u}}{\sum_{u\in K} w_{a,u}}$$

Where $p_{a,i}$ is the prediction for the active user a for item *i*, $w_{a,u}$ is the similarity between users a and u, and *K* is the set of most similar users.

The *vector-based similarity* is another way to compute similarity. It is originally used to calculate the similarity between two documents by treating each document as a vector of word frequencies and computing the cosine of the angle formed by the frequency vectors [12]. This formalism can be adopted in collaborative filtering, which uses the ratings of two users as a vector in an m-dimensional space, and compute similarity based on the cosine of the angle between them, given by:

[3] Breese et al. (1998) introduced the categorization: memory-based and model-based

$$w_{a,u} = \cos(\vec{r}_a, \vec{r}_u) = \frac{\vec{r}_a \cdot \vec{r}_u}{\|\vec{r}_a\|_2 \times \|\vec{r}_u\|_2} = \frac{\sum_{i=1}^{m} r_{a,i} r_{u,i}}{\sqrt{\sum_{i=1}^{m} r_{a,i}^2}\sqrt{\sum_{i=1}^{m} r_{u,i}^2}}$$

There have been several other similarity measures used in the literature, including Spearman rank correlation, Kendall's τ correlation, mean squared differences, entropy, and adjusted cosine similarity [13].

The *User-based CF* approach does not scale well when it is applies to millions of users and items, because of the computational complexity of the search for similar users. As an alternative, Linden et al. [14] proposed *Item-based Collaborative Filtering* where they match a users' rated items to similar items, rather than matching similar users. In this approach, similarities between pairs of items *i* and *j* are computed off-line using Pearson correlation, given by:

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2}\sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}},$$

where $r_{u,i}$ is the rating of user *u* on item *i*, *U* is the set of all users who rated both items *i* and *j*, $\bar{r}_i$ is the average rating of the *i*th item by those users. See figure 4.4, where you can see that user 2, l and n co-rated the same items *i* and *j* [11].

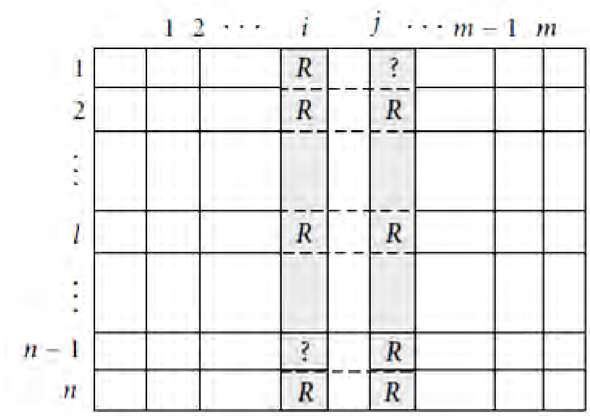

Figure 4.4 item-based similarity ($w_{i,j}$) calculation

Now, the rating for item i for user a can be predicted using a simple weighted average, as in:

$$p_{a,i} = \frac{\sum_{j \in K} r_{a,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|}$$

Where *K* is the neighborhood set of the *k* items rated by *a* that are most similar to *i*.
For item-based collaborative filtering too, one may use alternative similarity metrics such as adjusted cosine similarity; these are discussed in [11].

4.3.2 Model-based

Memory-based recommendation systems are not always fast and scalable, especially in the context of actual systems that generate real-time recommendations on the basis of very large datasets. Model-based recommendation systems involve building a model based on the dataset of ratings. In other words, information has to be extracted from the dataset, and can be used a "model" to make recommendations without having to use the complete dataset every time. This approach potentially offers the benefits of both speed and scalability.
Model-based collaborative filtering algorithms include Bayesian models (probabilistic), clustering models and recently, a new method based on matrix factorization has been successfully applied in the Netflix competition[4] and is the most promising approach now.

From a probabilistic perspective, the collaborative filtering can be viewed as calculating the expected value of a rating, given the profile of the user or the previous ratings. Assume that the ratings are integers with a range for 0 to m, the probability that the active user will have a particular rating for item *j* given the previously observed ratings is:

$$P_{a,j} = E(v_{a,j}) = \sum_{i=0}^{m} \Pr(v_{a,j} = i \mid v_{a,k}, k \in I_a) \cdot i$$

Cluster Models: Based on the idea that there are certain groups or types of users capturing a common set of preferences and tastes, Breese, et.al [15], proposed a cluster method, in which like-minded users are classified into the same group. Given a user's class membership, the user's ratings are assumed to be independent, then the joint probability of class and ratings could be calculated by the "naïve" Bayes formulation,

$$\Pr(C = c, v_1, ..., v_n) = \Pr(C = c) \prod_{i=1}^{n} \Pr(v_i \mid C = c)$$

Once the probability of observing an individual of a class with a set of votes is known, the expectation of the future vote could be easily calculated. Since the classes and number of class are unknown, Expectation Maximization algorithm is [16] used to find the model structure with maximum likelihood.

Bayesian Network Models: An alternative model formulation for probabilistic collaborative filtering is a Bayesian belief network with a node corresponding to each product in the database. The missing data can be represented by a "no rating" value. After applying an algorithm to train the belief network, in the resulting network, each item will have a set of parent items that are the best predictors of its votes. A decision tree could be used to represent the conditional probability table. [17]

Neural Network Models: Similar as the Bayesian Network models, collaborative filtering can be seen as a classification task. Based on a set of ratings from users for products, a model could be induced for each user that allows us to classify unseen products into two or more classes, for example *like* and *dislike*. An example of this method is given in Billsus' paper [18].

Singular Value Decomposition: in [19] Sarwar et al. investigated the use of singular value decomposition to reduce the dimensionality of recommender systems databases. It turns out, that using a smaller number of dimensions can actually improve prediction accuracy. For example, suppose two users both like science fiction movies. If one user has rated *Star Wars* highly and the other has rated *Empire Strikes*

[4] http://www.netflixprize.com/

Back highly, then it makes sense to say the users are similar. If we compare the users based on individual movies, however, only those movies that both users have rated will affect their similarity. This is an extreme example, but one can certainly imagine that there are various classes of movies that should be compared.

SVD is a well-known matrix factorization technique that factors an *m x n* matrix *R*, with *m* users and *n* items:

$$R = [r_{i,j}]$$

Where $r_{i,j}$ is the rating of user *i* for item *j*.

SVD factors the *R* matrix into three matrices as follows:

$$R = USV^T \quad ,$$

Where *U* and *V* are orthogonal matrices of size *m x r* and *n x r* respectively; *r* is the rank of the matrix R. And the S is a diagonal matrix of size *r x r* containing the singular values of the matrix R. U is representative of the response of each user to certain features. V is representative of the amount of each feature present in each product. S is a matrix related to the feature importance in overall determination of the rating.

Sarwar et al. [19] use SVD in recommender systems to perform two different tasks: First, they use SVD to capture latent relationships between customers and products that allows to compute the predicted likeliness of a certain product by a customer. Second, they use SVD to produce a *low-dimensional* representation of the original customer-product space and then compute neighborhood in the reduced space. They then used that to generate a list of *top-N* product recommendations for customers.

Advantages and disadvantages

The advantage of the collaborative approach is just like the demographic approach that no knowledge is needed about the products, so it is also domain independent. Collaborative filtering techniques are able to make recommendations “outside the box” because they look outside the preferences of the individual user [1]. The main advantage of the model-based approach is low memory and CPU- time requirements.

There still are several disadvantages, like the size of the data set influences the quality of the recommendations. And when there are new users and new products with no rating, the approach is not able to give a recommendation, because it is not able to identify users with the same preference, since there are no preferences of the user available. This is called the cold start problem [9]. Also the gray sheep and the stability vs. plasticity are problems here [9, 10].

The memory-based approach requires lots of memory and CPU-time, so it is applicable only to relative small number of users and items.

4.4 Content-based

Collaborative filtering uses the assumption that people with similar tastes will rate things similarly. Content-based filtering uses the assumption that items with similar features will be rated similarly. While collaborative and demographic information filtering methods do not require any additional product information, the content-based filtering approach depends on the availability of (manually created or automatically extracted) item descriptions and a user profile that assigns relevance to these characteristics. Those items in the catalog that are most similar to a query or to the user’s profile are

then recommended [20]. The relevance to the characteristics can be obtained by examining the ratings provided by the user, transactional data or website activity.
For example, in a movie recommendation application, in order to recommend movies to user *u*, the content-based recommendation system looks for the similarities among the books user *u* has rated highly (explicitly rated, purchased or viewed) in the past (specific actors, genres). Only the movies that have a high degree of similarity to whatever the user's preferences are would be recommended, see figure 4.5 for a simple example.
The user profile of preferences is stored as a vector of keywords. These profiles are obtained by analyzing the content of the items previously seen and rated by the user and are usually constructed using keyword analysis techniques from information retrieval. Information retrieval involves allocating various weights to keywords by use of algorithms such as the *Winnow* [21] and *Rocchio* [22] algorithms.

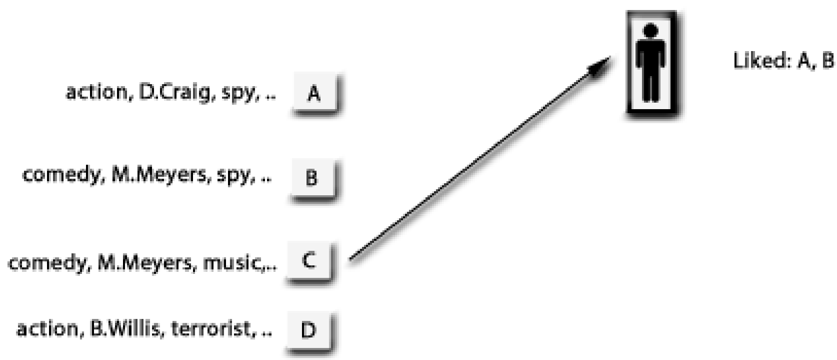

Figure 4.5 - Content-based approach

Advantages and disadvantages
Because the content-based approach uses item features and compares it with other items it does not matter what the item is to give recommendations, the approach does not need knowledge about the domain. The approach works well if the items can be properly represented as a set of features.
There are several disadvantages. When there are no item features available, the shop owner has to manually supply the items' features, which can be an unpleasant task. And the approach depends on the quality of the item metadata but also on the analogy of the stock, so items can be categorized. When the shop contains all unique products with all different product features, the approach will not be able to make any proper recommendations. The similarity computation is limited to the item features. The technique also suffers from the cold start problem and the stability vs. plasticity problem.

4.5 Knowledge-based

Knowledge-based recommendation has been discussed in [23]. It utilizes the knowledge about users and products and reasons out what products meet the user's requirements. The system offers for example a dialog that effectively walks the user down a discrimination tree of product attributes or there are systems that use a quantitative decision support tool for this task. Constraint-based and Case-based recommenders are examples of such systems. Let's look at the movie recommender system again. If a customer has stated that he or she likes movies with the genre comedy and with music, the system will look for movies that match these preferences. In figure 4.6 the movie that matches the users' preference best is movie C.

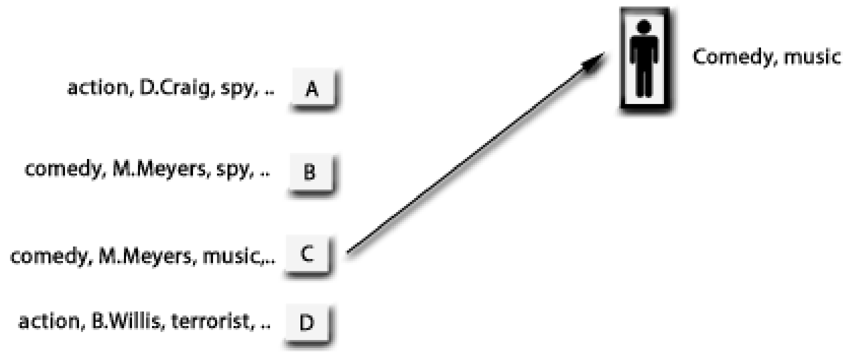

Figure 4.6 - Knowledge-based approach

Advantages and disadvantages
The advantage of the knowledge-based approach is that is does not rely on historical data. The system does not have to store any information about the user on a long term. So every user who uses the system provides their preference and can get their recommendations immediately. If the preference of the customer changes, it is easy to adjust it.
The disadvantage is that the system does not learn from your preferences, every time you will use the system you will have to state your preference. In order to give the right recommendations domain knowledge is needed. For example if you have chosen a set of preferences for a product, the system has to figure out which products will fit the preferences best, this requires some knowledge engineering.

4.6 Hybrid

Burke did a survey on several hybrid approaches [10]. Hybrid recommender system is another category of recommender systems that tries to overcome the limitations of the other approaches. A Hybrid recommender system combines two or more recommendation techniques to gain better system optimization and fewer of the weaknesses of any individual ones. The most popular hybrid approaches are those of content-based and collaborative filtering**.**
There are different strategies by which hybridization can be achieved and they are broadly classified into seven categories that are summarized in table 4.4.

Hybridization method	**description**
Weighted	The ratings of several recommendation techniques are combined together to produce a single recommendation
switching	The system switches between recommendation techniques depending on the current situation
Mixed	Recommendations from several different recommenders are presented at the same time
Feature combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm
Cascade	One recommender refines the recommendations given by another
Feature augmentation	Output from one technique is used as an input feature to another
Meta-level	The model learned by one recommender is used as input to another

Table 4.2 - Hybrid method categories

4.7 Overview

Below a summary can be found of the approaches given in this paper. Including the data source, the recommendation process and the pros and cons of the particular approach.

Approach	**item data**	**User data**	**Recommend**	**pros**	**cons**
Non-personalized	new / best sold / most visited etc.	x	new / best sold / most visited item etc.	- easy to realize - simple data needed	lack personalization
Demographic	x	Demographics	Item liked by a user with similar demographic	-no user-item rating needed - no item data needed - domain independent	- user data needed - privacy issues - not accurate
Collaborative	x	profile of items the user has liked	Item liked by a user with similar user data	- no item data needed -domain independent - outside the box - model-based: low CPU-time	-need enough data - cold start - gray sheep -stability vs. plasticity - memory-based: lots of memory and high CPU-time
Content	Keywords/ description	profile of items the user has liked	item with most similar description to one of the items in the user's profile	domain independent	- manual entering description - depends of quality of description - cold start -gray sheep -stability vs. plasticity
Knowledge	Keywords / description	Profile of users' requirements	Recommend the items that matches the users' requirements	- no historical data - no cold start - can handle changing preferences	- user has to give preference (burden) - does not learn

Table 4.3 - Overview

5 Examples

In this chapter we will look at four different e-shops that use a recommender system. For each e-shop, a brief description of the features of the system will be given. Because we cannot see the data and algorithms they use we can only assume the approaches they use for the recommendations.

5.1 Tripbase.com

The first website is Tripbase.com. The website helps users find the vacation destination that best matches their personal preferences. Travelers no longer have to waste time combing the web for the perfect personalized vacation. Tripbase works as follows[5]:

Data Aggregation
Tripbase has developed search technology that constantly scans the web for the most relevant, current and highest quality travel information. Tripbase has aggregated over 11 million points of data from sources that include traveler reviews, expert blogs and websites.

Data Analysis
Tripbase.com's proprietary artificial intelligence technology breaks down and analyzes that data quickly and precisely so that a traveler's preferences are matched with appropriate travel information from across the web.

Travel Recommendations
Tripbase.com reassembles all of the data that relates to a specific user and then gives unbiased recommendations for a personalized vacation, including the most appropriate destinations, activities, flights, hotels and more.

Figure 5.1 shows a screenshot of the website. The user has the option to *fly* or to *drive* to the destination. Tripbase asks to state the preferable *departure location*, the *number of travelers*, the *depart data* and *return date*.
Subsequently you must state how much of *Nightlife, Dining, Shopping, Nature* and *Attraction* you want to have in your vacation, the more you push the bar to the right the more you would like this particular characteristic to be present in your vacation.

Test
I want to fly from Amsterdam and want to depart on 02/23/2011 and want to come back on 03/02/2011
If I only want to shop on my vacation, I will have to put the bar *Shopping* to the maximum, see figure 5.1.

[5] http://www.tripbase.com/release2.do

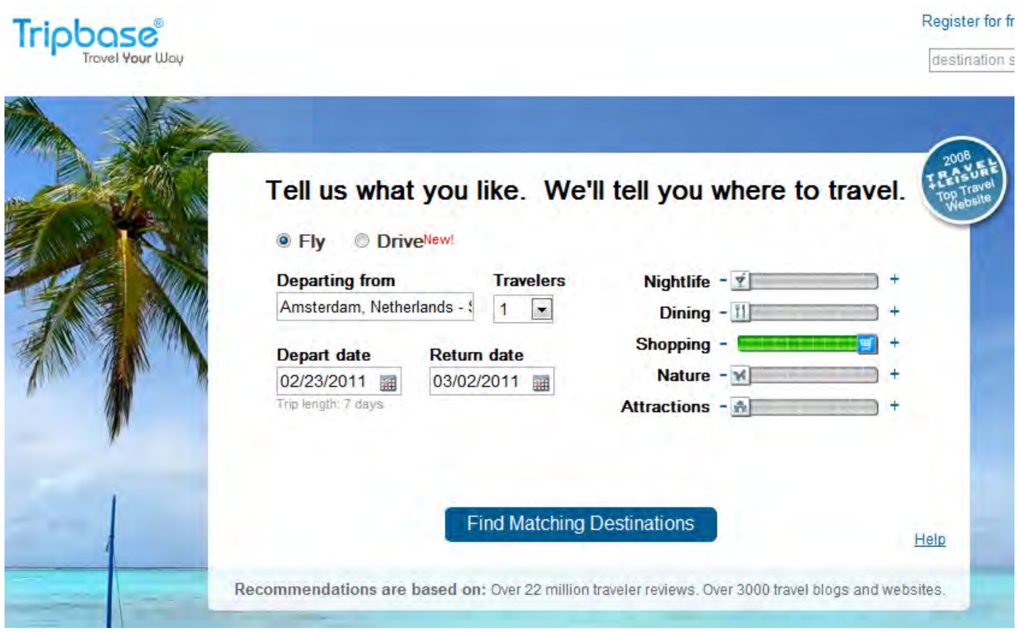

Fig 5.1 - Tripbase website

When clicking on the *Find Matching Destinations* button, Tripbase produced the following recommendations:

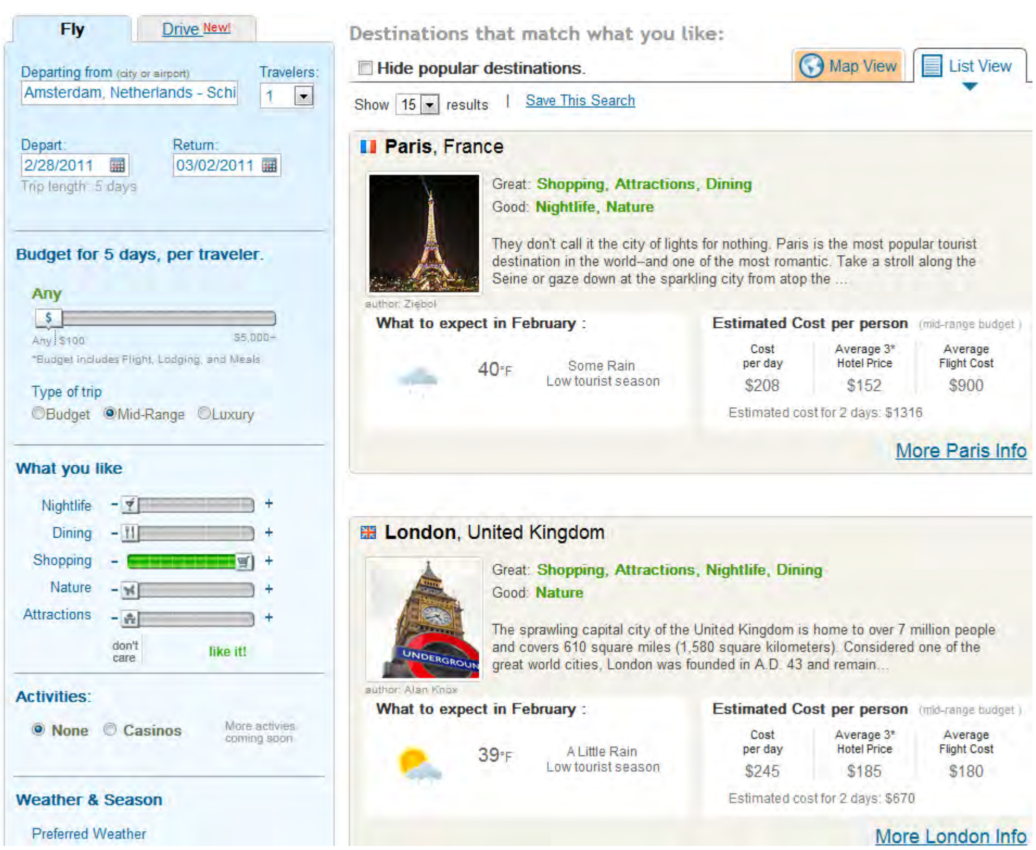

Fig 5.2 - Tripbase website recommendations

If I want to shop, the system recommends me to go to Paris or London. It also recommended Milan, Barcelona and Madrid as respectively third and fourth option.
As you can see, you have other possibilities to adjust your preferences and find the vacation that suits you best. You can adjust the budget, the type of trip, activities, weather and you can change the bars you have set earlier, etc. these options will all help to narrow down your search and find the best vacation for you.
For instance we will look at the situation when I will narrow the search. I want the trip to be of the type *Luxury* (figure 5.3). Tripbase give the following recommendations based on the extra preference. It implies that Milan is more luxurious than Paris or London.

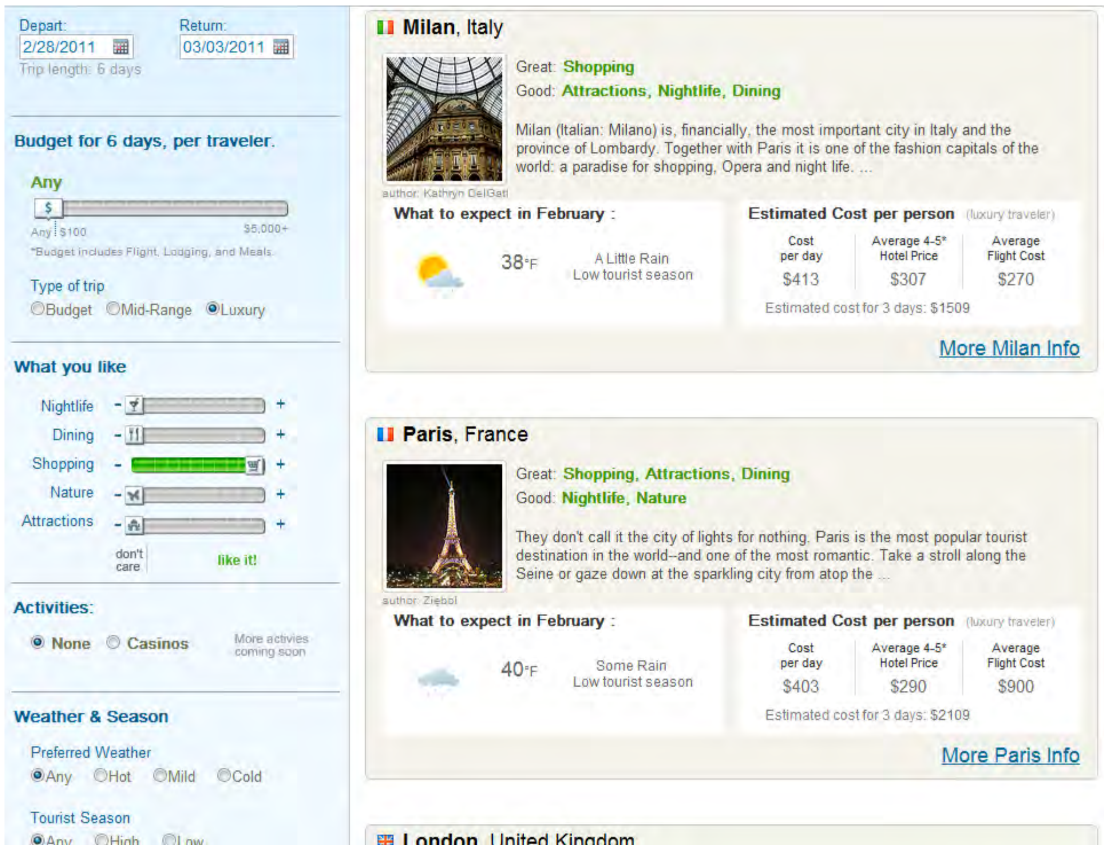

Fig 5.3 - Tripbase website recommendations - luxury

Used data

Explicit user data, the user has to give his preference in order to get recommendations from the Tripbase website.

Presumable approach

Looking at table 4.4 we can conclude that the recommendation approach is presumably the content-based approach or the knowledge-based approach. Because the system does not store any ratings or purchasing data from the user, this cannot be a content-based approach. So Tripbase.com uses the knowledge-based approach to give recommendations to its customers.

5.2 Jinni.com

Taste in movies is complex and individual. Yet the usual way of cataloging movies, by titles, people, and genres, flattens all this - as if you'd like a movie just because it's a Drama or stars Vince Vaughn. That's why the Movie Genome [6]was created, an ambitious, ongoing project with the Jinni community to map more aspects of movies, shows, and semi-professional videos than ever before - so that all different viewers can match their personal tastes and moods, and find what they really want to watch next. It is an internet application designed to fit how people relate to movies and TV. They have created tools to meet people with shared tastes, compare preferences, and review and rank titles.

Inside, the Genome is broadly divided in two:

1. Experience - the mood and tone of the content
2. Story - plot elements (One man army, Battle of the sexes), structures (Nonlinear, Story-within-a-story), flags (Violence, Nudity) and more.

The Genome also includes many external aspects like awards.

Data

The starting point of the Movie Genome is manual tagging by the team of film professionals. Each title has around fifty genes, among thousands of possibilities. Then, using advanced machine-learning technology and Natural Language Processing, Jinni's system indexes new titles automatically by analyzing user reviews and metadata. It also incorporates multiple perspectives (from reviews) rather than just one person's opinion. Everyone who votes on genes, as well as the Jinni team, constantly checks and improves the machine tagging. So Jinni uses product data and user data.

Recommendations

According to Jinni the best recommendations use man and machine. A machine can deeply analyze the type of content you like to learn about your unique taste. People can share their personal favorites and opinions about what they've seen (in a way no machine can do, as yet). Jinni isn't a social network, it is a service meant to fit how people experience media - and they have included dialogue about movies and shows as part of that.

Test

To test the recommender system on Jinni.com I had to sign up, where I first was asked to give a username, email address, password and country. Afterwards it is optional to give your gender, year of birth, the kind movie genres and plots you like. Obviously when you give information about which genre and plot you like, the recommendations will be more accurate. When you are finished with selecting the plots and genres you like and don't like you have the option to share Jinni with your friends by sending them an email or trough a social network platform. Then to get started you have to start rating movies. There are twelve types of M*ovie Watchers* (figure 5.4) and the intention is that you have to determine which type of movie watcher you are and rate the movies from these types (figure 5.5).

[6] http://www.jinni.com/movie-genome.html

Fig. 5.4 - Type of Movie Watchers

Fig. 5.5 - Type of Movie Watchers

Once you have rated enough movies, Jinni is able to create a *Movie Personality sketch* and give recommendations, see figure 5.6. You are able to rate these if you have already seen them, remove the movie if you don't like it, watch the movie (e.g. on Netflix or buy DVD on Amazon.com), put it on a wish list, put in on a favorites list or find more.
Another feature of Jinni.com is the ability to see what movies your neighbors like, users that have rated movies similar to you, see figure 5.7.

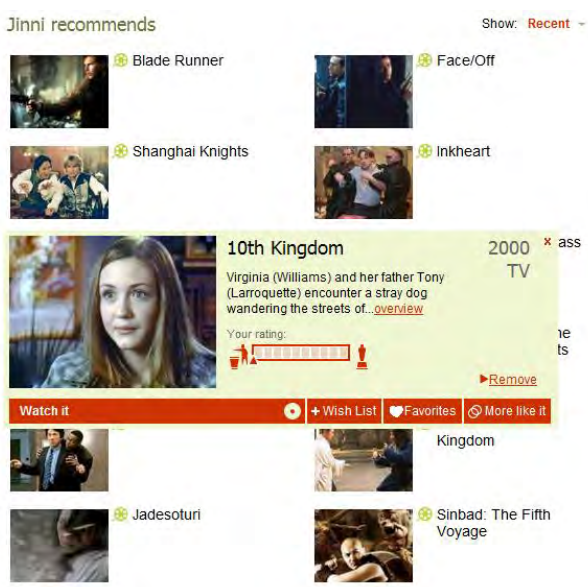

Figure 5.6 - Movie recommendations Jinni.com

Figure - 5.7 Neighbors

Data
Jinni.com uses customer ratings and movie features, such as movie plots and genres.

Presumably used approach
Jinni.com presumably uses a hybrid approach. With the content-based approach, because they use movie description like genre and plots. The collaborative filtering approach, because according to the ratings you give they are able to compare your ratings to other Jinni users and give surprising recommendations and give a list of your neighbors. And lastly they use the knowledge-based approach, because you are able to state your preference to a certain movie genre or plot and get recommendations accordingly.

5.3 Wehkamp

Wehkamp is an online department store for Dutch consumers. Wehkamp uses a lot of ways to recommend items to its customers. They use different recommendation approaches for the different separate blocks on a product page. For example I am interested in buying a Playstation 3 controller, see figure 5.8 for the different recommendation blocks give on the page.

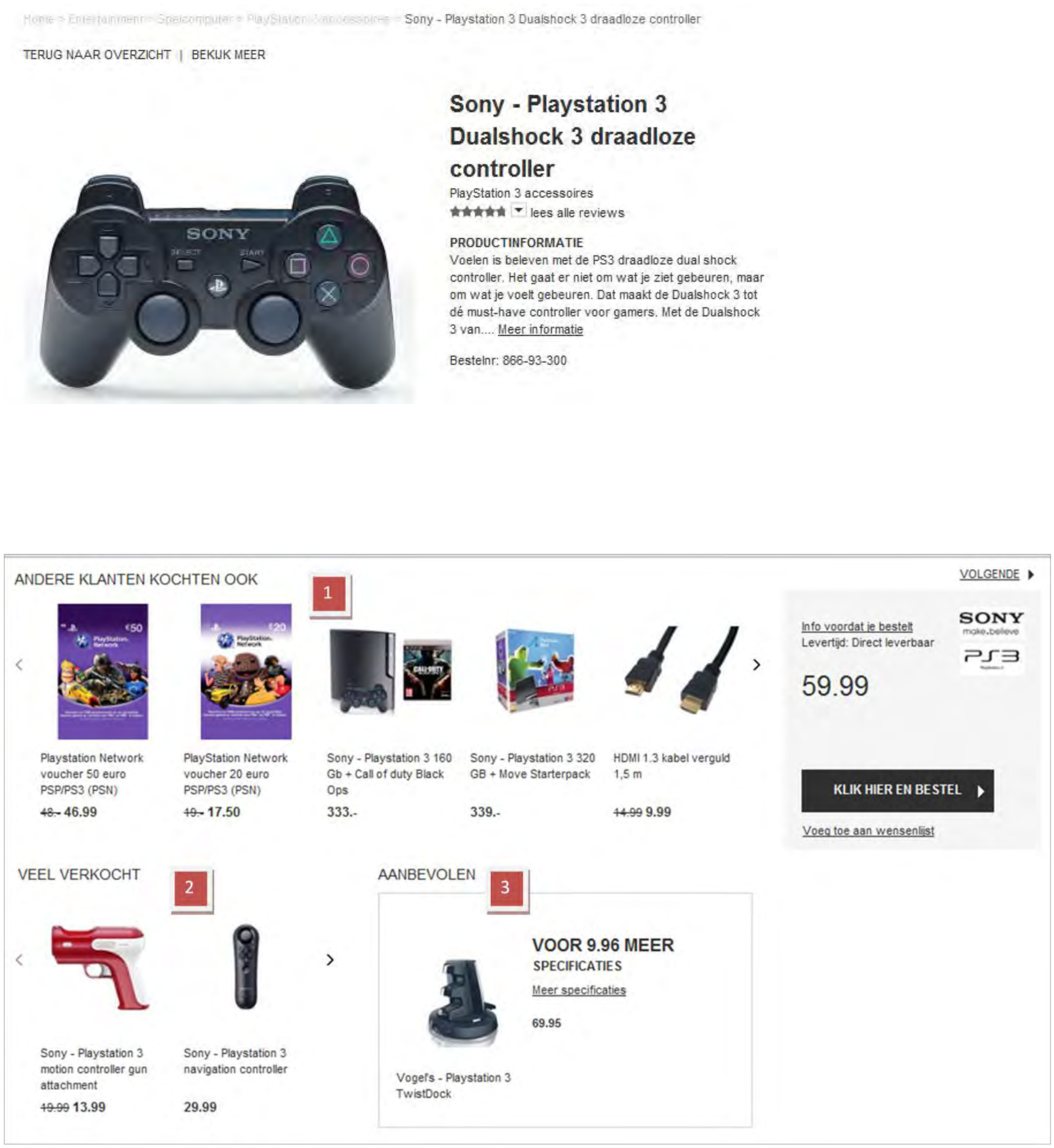

Fig 5.8 - Wehkamp recommendation blocks

Recommendations

Andere klanten kochten ook
The first block *Andere klanten kochten ook* shown in figure 5.8 is a nice example of a collaborative filtering approach (item-based). Item-based CF looks at the target user's chosen item and finds other content in the choice set that it deems similar to that item.

Veel verkocht
The second block *Veel verkocht* shows the items that are top sellers in the given category. They offer products of the same brand and type as the item you are interested in. This might be a combination of the non-personalized and content-based approach, where they look at the features of the current viewed product and give the most sold items that match the features of the viewed product

Aanbevolen
Block number three *Aanbevolen* is for up-selling. Wehkamp tries to sell a (better/completer) product for a higher price.

Reviews & Vragen
On every page there is also a block for customer comments, it allows customers to receive text recommendations based on the opinions of other customers. Located on the information page, the customer can either ask a question about the product or give a review. Customers have the option of incorporating these recommendations into their purchase decision. Furthermore, customers can "rate the comments." With each comment is the question "Did this comment help you?" ("Was dit antwoord nuttig?"). Customers may indicate yes or no. This can be seen as a form of a non-personalized recommendation approach, all the reviews are the same to every customer viewing the page.

Voordeel combinatie
On another product page for a digital camera, another block is added, the *Voordeel Combinatie* block see figure 5.9. Wehkamp tries to cross-sell products with a discount. The memory card is a well fitting product to the camera. These products are probably set manually by Wehkamp in order to give the right discount combination.

Combinatie
The last block *Combinatie* is actually the same as the "voordeel combinatie" block, they only offer more products that fit well with the product you are viewing. And you do not get a discount for buying them all together. These recommendations are probably provided with a content-based approach, where watching the page provides a positive implicit rating and together with the product features like the brand and the type of product the recommendations are given.

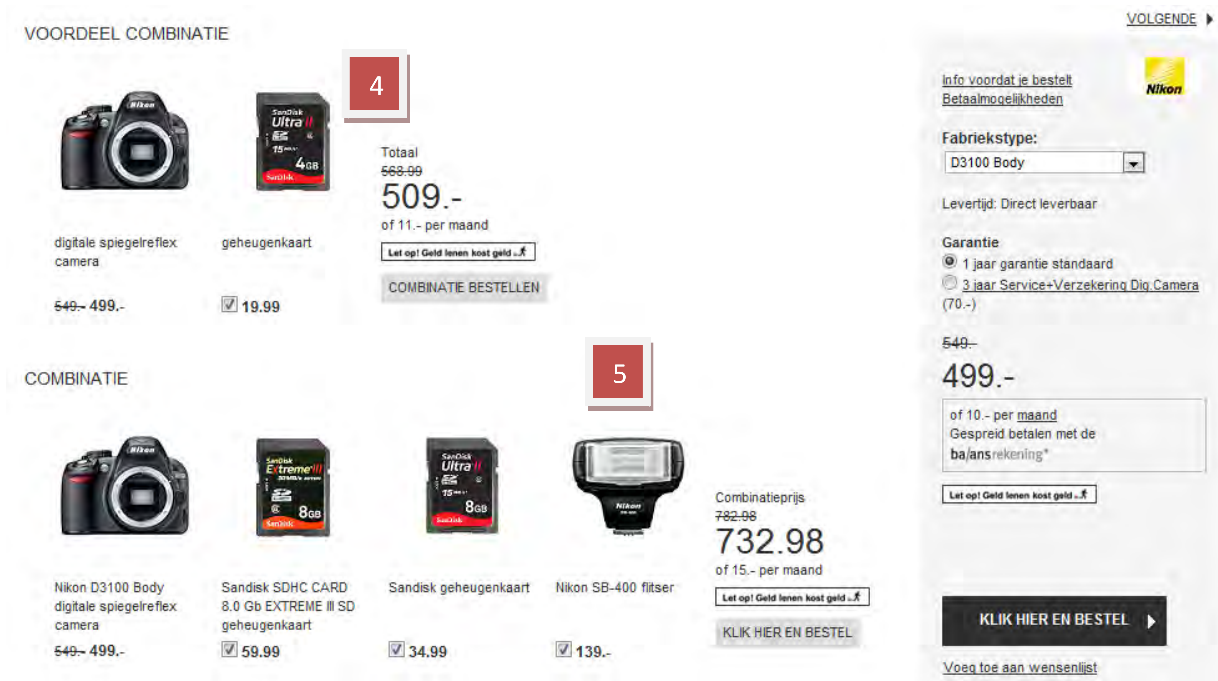

Fig 5.9 - Wehkamp recommendation blocks

Email

Wehkamp also gives recommendations through email. These recommendations probably has been achieved with the content-based approach where viewing a page indicates a positive rating. Probably because I viewed the Playstation 3 controller page, Wehkamp gave the recommendations as you can see in figure 5.10

Fig 5.10 - Wehkamp email recommendations

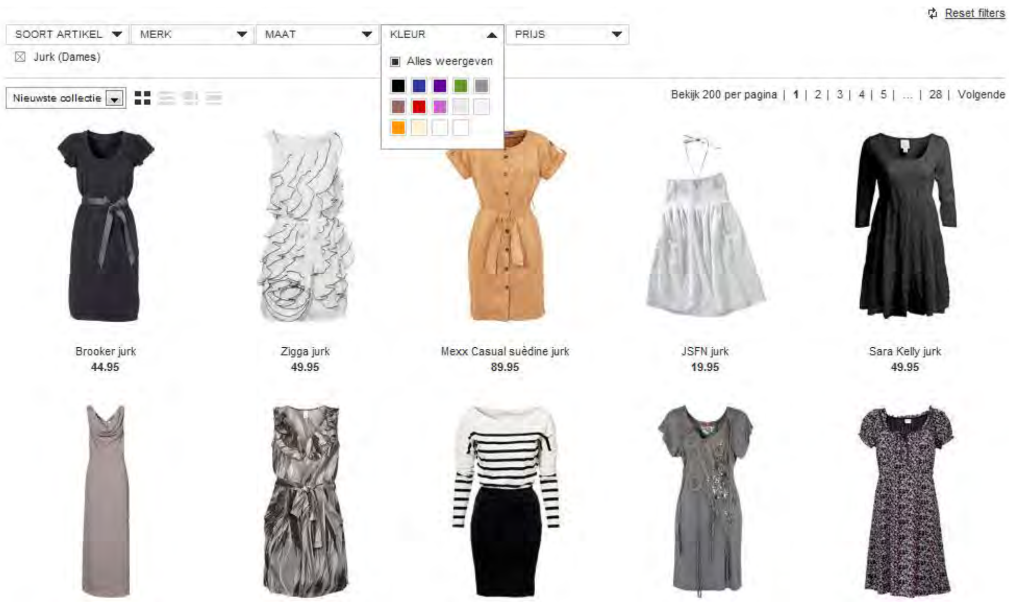

Fig 5.11 - Wehkamp email recommendations

Filters

When we want to buy clothing on Wehkamp, you are able to use filters to narrow down your search. As you can see in figure 5.11 I already decided that I want a dress, so Wehkamp only shows me dresses. I can also provide the brand, size, color and my budget to narrow down the search. If I decide I do not want a dress but a t-shirt I can reset the filter and start again by applying other filters. This way of recommending products correspond to the knowledge-based approach.

Presumably used approach

Wehkamp uses a lot of recommendation blocks on different product pages. They presumably use the hybrid method. For every product type they use a suitable approach, this corresponds to the switching hybrid approach.

5.4 Mybuys

MyBuys is a provider of personalization for multi-channel retailers, some examples can be found in figure 5.11. The company builds deep profiles based on each individual shopper's behavior, and then uses a patented portfolio of algorithms and real-time optimization to deliver the most engaging recommendations.

Fig 5.11 Mybuys clients

Data

Mybuys get insight by bringing together four key data sources:

Transaction history
Past transactions from the web, catalog and stores.

Product catalog
Examine and index everything about every product, including category, brand, price point, promotions, size, gender, age, fabric, color, durability and keywords.

Website activity
Track each click, with corresponding product and price. What a shopper searches for, where and how each shopper browses, what gets put into their shopping cart, and what's been abandoned or purchased.

Personal choices
Explicitly tell about products and categories of interest, so that it can be used for the recommendations.

Presumably used approach
According to the data they use to give recommendations to clients, Mybuys presumably use a hybrid approach. *Transaction history*, website *activity* and *personal choices* can be used in the content-based and collaborative filtering approach.
Product catalog information can be used in the content-based and knowledge-based approach.

6 Conclusion

Recommender systems are a powerful technology for overcoming the information overload and achieve personalization. Used in the right way, they can benefit both consumers and businesses. Consumers benefit by discovering new interesting products, they had not found otherwise. And e-shop owners can increase their sales.

There are many application fields for recommender systems and many have their own requirements that need to be fulfilled by different approaches. So which recommendation approach works best always depends on the concrete use case. No matter how clever an approach is, there is no single algorithm that will give the best recommendations for every retailer, online retailers and shoppers vary way too much.

If the online retailer does not want to collect any information about the user and does not have much information about the products, he can use the *non-personalized approach*; this however will be the same for every user and will lack personalization.

The *demographic-based approach* can be used if the online retailer has information about the customers and only wants to sell products to a particular group, like young or old, men or women, educated or non-educated, etc. this approach will not always be successful, because not all customers can be stereotyped.

When the online retailer has enough information about the users' ratings for products in the store and wants to recommend out of the box products, the *collaborative filtering approach* will be the right one. However the retailer has to keep in mind that new unrated products are not recommended and new customers cannot get recommendations until they start rating products. It is also hard to give recommendations to users who don't have rated products similarly to other customers. And it is probably expensive to run the approach on a big database.

The *content-based approach* is useful when the features of the product are available, and if not the retailer does not mind to provide the products features manually. A feature has to return in multiple products or else no recommendations can be made, because of the lack of similarity. So the approach will not work if the retailer sells unique products. And the system must be able to extract the users' preference for a product (e.g. through ratings, transactional data or website activity).

If the historical data of the users is not stored or available. And if the retailer sells products where the customers' interest can change depending on for example the occasion, for example on an online clothing store, where a customer one day might be interested in a blue skirt and another day in black pants. The *knowledge-based approach* is suitable.

The *hybrid approach* is suitable if one of the previous approaches does not give the recommendations as accurate as you would like them to be. The hybrid approach tries to overcome the limitations of the other approaches and combine the advantages of the other approaches by using two or more approaches.

Every single online retailer has to investigate their business and understand what data they can/want to retrieve about the customer and about the products. According to this information the online retailer will be able to find a suitable approach for the recommender system.

7 Literature

[1] J. B. Schafer, J. A. Konstan, J. Riedl, E-Commerce Recommendation Applications, *Data Mining and Knowledge Discovery*, Vol. 5 Issue 1-2, 2001, p. 115–153

[2] S. Perez, Etsy Experiments with New Recommendation Engine, on ReadWriteWeb 2010

[3] D. M. Nichols, M. B. Twidale and C. D. Paice , Recommendation and Usage in the Digital Library, *Technical Report - CSEG/2/97,* 1997

[4] C. L. Borgman, S. G. Hirsh and J. Hiller, Rethinking online monitoring methods for infomation retrieval system: from search product to search process, *Journal of the American Society for Information Science*, Volume 47 Issue7, 1996, p. 568-583

[5] Y. Fan, Y. Shen, J. Mai, Study of the Model of E-commerce Personalized Recommendation System Based on Data Mining, *Proceedings of the 2008 International Symposium on Electronic Commerce and Security*, 2008, p. 647-651

[6] M. J. Pazzani, A Framework for Collaborative, Content-Based and Demographic Filtering, *Artificial Intelligence Review* – Special issue on data mining on the Internet, Volume 13 Issue 5-6, 1999, p. 393-408

[7] S. Lam, D. Frankowski and J. Riedl, Do You Trust Your Recommendations? An Exploration of Security and Privacy Issues in Recommender Systems, GroupLens Research *In Proceedings of the 2006 International Conference on Emerging Trends in Information and Communication Security (ETRICS),* 2006

[8] C. Thompson, If You Liked This, You're Sure to Love That, *The New York Times Magazine, 2008*

[9] M. Claypool, A. Gokhale, T.Miranda, P. Murnikov, D. Netes, and M. Sartin, Combining Content-Based and Collaborative Filters in an Online Newspaper, *In Proceedings of ACM SIGIR Workshop on Recommender Systems,* 1999

[10] R. Burke, Hybrid Web Recommender Systems: Survey and Experiments, *User Modeling and User-Adapted Interaction*, Volume 12, Issue 4, 2002, p. 331–370

[11] B. M. Sarwar, G. Karypis, J. A. Konstan, J. T. Riedl, Item-based Collaborative Filtering Recommendation Algorithms, In *Proceedings of the 10th international conference on World Wide Web*, 2001 p. 285-295

[12] G. Salton, M. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983

[13] J. L. Herlocker, J. A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999

[14] G. Linden, B. Smith and J. York, Amazon.com Recommendations, Industry report: IEEE Internet Computing, 2003, p. 76-80

[15] J. Breese, D. Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI '98)*, 1988

[16] A. P. Dempster, N. M. Laird and D.B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal Of The Royal Statistical Society, Series B,* 1977*,* p. 1-38

[17] Y-H Chen and E. I. George, A Bayesian model for Collaborative Filtering. *In Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*, 1999

[18] D. Billsus and M. Pazzani, Learning Collaborative Information Filters, In *Proceedings of the 15th International Conference on Machine Learning*, 1998

[19] B. M. Sarwar, G. Karypis, J. A. Konstan, J. T. Riedl, Application of Dimensionality Reduction in Recommender System - A Case Study, In *ACM WebKDD Workshop,* 2000*,* p. 285-295

[20] M. Balabanovic and Y. Shoham, Fab: Content-based, collaborative recommendation, *Communications of the ACM*, vol. 40(3), 1997, p. 66–72

[21] N. Littlestone, , Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm, Machine Learning, Vol. 2 (4), 1988, p. 285-318

[22] http://nlp.stanford.edu/IR-book/html/htmledition/the-rocchio71-algorithm-1.html

[23] R. Burke, Knowledge-based recommender systems, In *Encyclopedia of Library and Information Systems*, Vol. 69 (32), 2000, p. 180–200