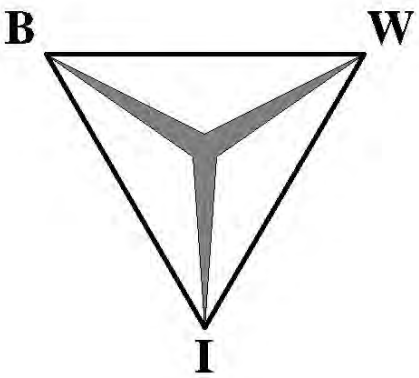
Recommender systems:
An overview



Recommender systems: An overview
Werkstuk Bedrijfswiskunde & Informatica
Guido Jan de Nooij
Student Number: 1516892
Vrije Universiteit Amsterdam
Supervisor: Dr. Wojtek Kowalczyk
November 2008

Abstract

Recommender systems are increasingly used in e-commerce websites, because they give businesses a strategic advantage over businesses without them. Recommender systems can be described as an information filtering technology that is used to present information on items to the user that are in line with the users tastes. These systems involve predictive models, heuristic search, data collection, user interaction and model maintenance. In other words we can see recommender systems as smart search engines which gather information on users and or items in order to give customized recommendations by comparing users and or items with each other.
This smart search is achieved with help of historical data on products and customers. For instance the historical data can consist of ratings given by customers on products or another possibility is that previously purchased items are used. The historical data consists of information on items and customers. Businesses often have a large variety of products that are for sale. As the number of products increase customers will have more difficulty finding what they like. Recommender systems offer a solution as they are designed to provide customers with recommendations. As recommendations reduce the amount of time a customer needs to invest in searching for products of his or her liking, the overall user's experience is positive.
Customers visiting websites are more likely to return on a later date if the overall experience of a website is more pleasant in comparison to other websites. In consequence, sales figures improve over time if recommender systems are integrated in the website. Recommender systems improve customer satisfaction and loyalty; furthermore they increase cross-selling.
Although recommender systems give a strategic advantage they do come with some potential problems that have to be dealt with. For example recommender systems can suffer from cold start problems. New users that are added to the system need time to receive accurate recommendations. The same applies for newly added items. Recommender system will therefore also require some additional maintenance.
In this paper we will present an overview of different algorithms that are in use, business aspects of these recommender systems, the data that is used to make these recommendations and finally other relevant issues like profile injection attacks.

Acknowledgements

In the last stages of the master program Business Mathematics and Informatics it is expected of students to write a paper which describes a certain scientific topic. The target audiences of the paper are people with a scientific degree that wish to learn more about recommender systems.
I want to thank Dr. Wojtek Kowalczyk for helping me narrow down the topic and his help while writing this paper.

Guido de Nooij
Amsterdam 2008

CONTENTS

1 Introduction

Recommender systems are becoming an important business tool in e-commerce, as more and more companies are implementing this feature into their website. Recommender systems were originally designed to overcome the large quantity of data available. However as websites with recommender systems showed an increase in sales figures it became evident that recommender systems also gave a strategic advantage over websites without recommender systems.
E-businesses offer a wide variety of items through the internet, some E-businesses even offer over millions of items. Therefore the customer can have trouble finding products that he or she is looking for. Recommender systems can offer a solution to this problem as customers will get recommendations using a form of smart search.
Recommender systems typically are types of collaborative filtering that involve predictive models, heuristic search, data collection, user interaction and model maintenance. The system usually needs to be updated periodically with newly added ratings, items and users. In other words a recommender system is an information filtering technology designed to determine items that are most likely to the customer's tastes. After the best items have been determined they are recommended to the user. Recommender systems interact with users on their preferences and form a profile of each customer usually based on ratings of items. The different profiles are compared to each other with help of an algorithm and are used to estimate and predict the items that are most likely to the user's tastes. In short recommender systems are a type of heuristic search that uses gathered and stored information of users and or items to predicts and recommend what items users will like.
An example of an E-business that uses recommender systems is Amazon.com. In figure 1 we can see an example recommendation given by Amazon.com while browsing for a book. This recommender system returns the most popular items that customers bought in addition to the selected item at some point in time. In other words: other books customers also purchased/liked. However, this is not the only recommender system that amazon.com uses. For instance customers can also choose to sign up for an e-mail service that sends out e-mails of newly added items or other items the customer might be interested in.
Recommender systems can vary in size and shapes. Some recommender systems compare items to other items whereas others compare customers with other customers. Some require registration or a minimal amount of rated items, others do not. Some are only active when on the website others use a subscription to an e-mail service. Because of the variation in recommender systems this also implies that a lot of research has been done.

Figure 1 Recommendations received while browsing for a book on amazon.com.

Recommender systems often use ratings from customers for their recommendations. An example of such a system is the Netflix video rental service. This company rents out

DVD's through the mail, after customers have registered and rated a minimal number of movies (20). The customer is then compared to other similar customers and based on these similarities a couple of movies will be recommended. The advantage of such a system is that the customer is more likely to rent movies that he or she is interested in. We can compare recommender systems to store salesclerks, but in the case of salesclerks the compatibility of the tastes is unknown.
Although we have to mention that also with recommender systems a small portion of customers which have very unusual tastes could also get inaccurate recommendations. However the majority of customers will receive accurate recommendation.
Furthermore recommender systems can bring customers into contact with movies that he or she otherwise never would have considered, but would like none the less. Netflix uses a highly accurate recommender system, however as they are looking for ways to even improve their recommendations they created the Netflix prize. Other businesses or individuals were challenged to create a recommender system that improved the system. If a 10% increase in accuracy is achieved Netflix awards a million dollars. Furthermore an award of 50.000 is handed out to teams or individuals that have achieved a significant improvement. More information on the Netflix prize can be found at Netflixprize.com.
E-businesses that implemented recommender systems have noticed a considerable increase in sales revenue. Recommender systems affect the overall experience of customers in a positive way. Customers will be more satisfied with the service and this leads to increasingly loyal customers. Recommender systems will furthermore increase sales by improving cross-selling, which is the purchase of additional items with the item the customer is buying.
In this paper we will describe different aspects of recommender systems. In chapter 2 we will start with describing the data that is used for recommender systems followed by several algorithms of recommender systems. We will describe how they work and discuss problems that might occur. In chapter 4 we will describe the business aspects of recommender systems, why are recommender systems so important? What can be expected from the recommender system when implemented? Next we will review different issues of recommender systems, and try to summarise the findings of the different papers on among others profile injection attacks.

2 Business aspects of recommender systems

Companies are always looking for ways to increase their sales figures. The same applies to e-commerce businesses. Businesses typically have large amounts of products for sale. Therefore finding products you like can become difficult. By making customized recommendations we can help customers find product and decrease the burden of going though numerous items. Recommender systems are believed to help E-businesses in the following ways.

- Customers spend less time searching for products (smart search)
- Customer satisfaction is increased
- Customer loyalty is increased
- Cross-selling is increased

The idea of recommender systems is to recommend items to the user and if the item is appealing enough the customer might buy it. It is important for a recommender system to have enough knowledge of the user's interests in order to give accurate recommendations. If we look at a typical street corner store we would not find anything different. Such a store would arrange their window display best suited to the interests of the potential customers passing the store. A good window display could boost the sales, as more customers will enter the store and look around. When the customer is in the store at some point a sales clerk will ask if he can help. The clerk tries to find the interest of the customers and shows items which are best suited for the specific customer. Basically the clerk first has to find the interests of the customer and secondly create the feeling of trust towards a product and or store.
E-commerce sites try to stimulate this process. The recommender system can be seen as an online equivalent of the sales clerk. By showing items in which the user is interested the probability of a customer buying a product increases. Trust in the product is created by having a lot of information on the products capabilities and possibly giving customers the opportunity to give feedback like rating systems and a comment section. Incorporating trust towards the website is usually not a task of a recommender system, but (Tintarev et. Al. 2007) shows that it is important for the recommender system to be properly explained. This could positively influence trust, user satisfaction and loyalty. Trust in the website can also be established through other means, for instance by having reliable payment methods (for instance PayPal) or having a seal for trustworthiness.
In a store before the sale is finalised, a sales clerk will try to persuade a customer to buy an additional item(s). Usually in normal stores this would be an item which has some relevance to the product the customer is about to buy. The selling of additional items, with the item the customer was originally interested in, is called Cross-selling. An example of cross-selling would be a store which is selling mp3 players, a well trained sales clerk would ask: do you want some rechargeable batteries with that? In this case there is a reasonable chance that the customer decides to purchase the batteries along with the mp3 player. Cross-selling techniques are also being used online. Without recommender systems they would have to sell packages. That is the possibility to buy mp3 players with and without rechargeable batteries. Another method could be a discount on a secondary item, for instance 30% discount on the cheapest product.
With recommender systems a wider variety of products become an option for cross-selling. Because items are shown that match the customer's interests, he or she will be more perceptive to buy additional items. This also applies for product of an entirely different category. To summarise, recommender systems improve e-commerce by increasing cross-selling and turning browsers into buyers. Furthermore (Schafer et. al. 2003) stated that recommender systems also improve the loyalty of customers. Customer loyalty is an aspect that is important in business models.

A lot of company's design Business models to be able to have a competitive advantage over other similar businesses.
In (Osterwalder et. al. 2005) a definition is given of a business model:
"A business model is a conceptual tool that contains a big set of elements and their relationships and allows expressing the business logic of a specific firm. It is a description of the value a company offers to one or several segments of customers and of the architecture of the firm and its network of partners for creating, marketing, and delivering this value and relationship capital, to generate profitable and sustainable revenue streams."
Different definitions exist in the literature and often business models are viewed as an equivalent of strategy. And if strategy is not seen as an equivalent of a business model it should at least be seen as a part of the model. This is the case with the loyalty and customer satisfaction models which will be described below.
The customer satisfaction model[1] by prof. N. Kano describes 6 factors that influence customer satisfaction: Basic factors, Excitement factors, performance factors, indifferent factors, doubtful factors and finally reversed factors.
The first three influence the customer satisfaction the most if they are not the only factors. Basic factors are the minimum requirements that customers expect from a product, if they are not met, the customer will be unsatisfied. Excitement factors are factors which create a sense of increased pleasure in the product, but are not of influence when left out. Performance factors are factors which create satisfaction when performance is high and discomfort when low.
Indifferent attributes are attributes in which the customer has no interests and thus will have no or minimal impact when present or left out. Doubtful factors are factors from which it is unsure if it is expected by customer or not. Reversed factors are factors from which the reverse was expected by the customer. In the field of recommender systems we can view the correctness of the recommendations as the performance factor, basic factors would be the timely delivery of the product and. Excitement factors could be an extra feature the product has.
The satisfaction of customers is closely linked with customer loyalty, loyalty is defined as:" the intention of a customer to repurchase products/services through a particular e-service vendor" (Luarn et. al. 2003). Loyalty of customers is becoming a very important aspect in e-commerce; the idea is that if a company can make the customer loyal, the overall revenue of the company will increase over time. Another view would be that a lot of money is spent on acquiring customers, although this money most probably will be spent anyway, inducing loyalty would be cheaper in comparison.(Reichheld and Sasser1990 see Wikipedia link) Furthermore the company will have a strategic advantage over competitors, that is company's that can't induce loyalty in customers.
(Luarn et. al. 2003) researched different aspects which were believed to influence customer loyalty. The conclusion was that trust, customer satisfaction, commitment and Perceived value directly influenced loyalty. Customer satisfaction and perceived value also indirectly influenced loyalty through commitment. Where commitment is defined as: "a customer's psychological attachment to an e-service that develops before a customer would be able to determine that their repeat purchase behaviour was derived from a sense of loyalty" (Beatty et al 1988). And where perceived value is composed of 2 components, the get component and a give component. In e-service context the get component will consist of product, service and website quality, whereas the give component would be time spent on the website by the customer and the money that is spent. From the conclusions of (Luarn et. al. 2003) we now (partially) understand what is important to create loyalty towards eservices.
To summarise a company would be making a good strategic decision to invest in well trained personnel or in this case systems. This would result in increased customer satisfaction and trust. Increased customer satisfaction and trust will result in increased loyalty of the customer, which finally would result in increased profits and revenues.

[1] http://www.12manage.com/methods_kano_customer_satisfaction_model_nl.html

2.1 Sales diversity

Recommender systems are though to have an impact on sales diversity, some studies expect that recommender systems increase the sales diversity, because the customers will discover new products (Brynjolffson et al. 2006). Others believe recommender systems can have the opposite effect a decrease in sales diversity (Mooney et.al. 1999). These opposite views have resulted in 2 papers which try to analyse the effects of recommender systems on sales diversity. (Fleder et. al. 2006) has researched the sales diversity and came to the conclusion that although recommender systems are expected to increase the sales diversity, the use of voting systems could also have an opposite effect. The results furthermore show that recommender systems are dependant on previous outcomes, that is they are highly dependent on the early events in the recommender systems. A secondary paper (Fleder et. al. 2007) researched 2 specific recommender systems using simulation. One recommender system was expected to increase sales diversity and another was expected to have a concentration bias. The conclusion was the same for both systems, in some cases the diversity increased and in other cases it decreased. It was again highly dependent on the path chosen early on. Although sales figures increase it are "mostly" popular items that become even more popular and thus unpopular items will probably even become more unpopular.
It was hoped that recommender systems would also increase the sales diversity, however businesses that implemented recommender systems also noticed this negative effect on sales diversity, most managers tend to think that recommender systems decrease sales diversity instead of increasing it.

2.2 Cost-benefit analysis

Although the recommender system seems to be important for strategic management, we could make an analysis of how beneficial a recommender system is. We use the return on investment metric[2] used in cost benefit analysis.

$$ROI = \frac{gain - \cos t}{\cos t} \tag{1}$$

To be able to use this formula for the analysis of recommender systems we first have to determine the costs. As we want a comparison of a company with and without a recommender system, we need to remember that only extra costs used for installation and maintenance of the recommender system needs to be incorporated in the model. The installation costs(C) are fixed and maintenance(c) is variable, we get

$$\cos t = C + (c \cdot T) \tag{2}$$

where T is time (in years).
The gain of recommender systems is a little trickier as we would have to consider the expected increase in cross-selling, the expected increase of site visitors that decide to buy an item and the increase of customer loyalty. We choose not to include up-selling in our model. Up-selling is a sales technique which is aimed at increasing the profit of a sale by convincing customers to purchase a similar but more expensive item.
Although recommender systems provide recommendations they (mostly) are not aimed at increasing up-selling. Instead they are designed to find products customers are interested in. Therefore it is uncertain whether recommender systems increase up-selling. However, if

[2] http://www.investopedia.com/terms/r/returnoninvestment.asp

desirable up-selling can be included relatively easily, as it will be similar to the calculation of cross-selling.
The expected increase in cross-selling sales is

$$E(\text{increase in cross-selling}) = T \cdot Sales \cdot CS \tag{3}$$

where sales is the number of sold items per year and *CS* is the expected percentage increase in cross-selling success i.e. cross-selling conversion rate. The expected increase of visitors turned into buyers is

$$E(\text{increase visitors into buyers}) = T \cdot V \cdot B \tag{4}$$

where *V* is the total number of visitors of the site and *B* is the conversion rate of visitors (increase in the expected percentage of visitors that turn into buyers) after the installation of the recommender system(percent points). To incorporate the increase in loyalty we assume sales to consist of 2 groups, new customers (*E(new)*) and customers that remained loyal (*L* in a percentage). In other words *L* is a percentage of customers which will buy other items of the website within a time span of a year, we get

$$Sales(t) = E(new) + L \cdot Sales(t-1) \tag{5}$$

The increase in loyalty compared to the situation without a recommender system becomes the sales at time *t* minus the sales at time *0*.

$$E(L \ at \ time \ t) = Sales(t) - Sales(0) \tag{6}$$

We finally come to a total gain of

$$gain = P_{cs} \cdot (T \cdot S \cdot CS) + P \cdot T \cdot V \cdot B + P \cdot \sum_{t=0}^{T} E(L \text{ at time t}) \tag{7}$$

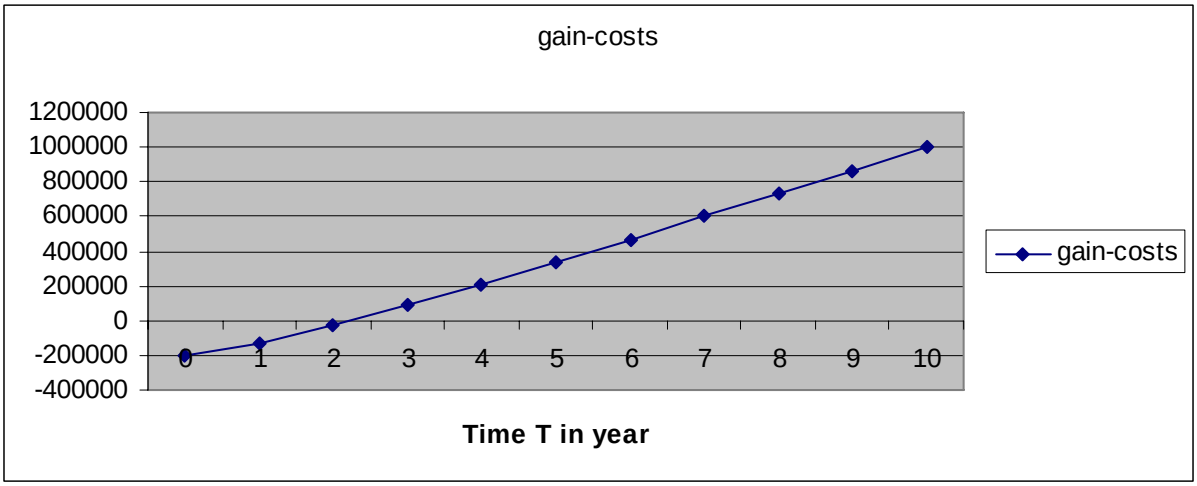
where *P* is the average price of an item and P_{cs} is the average price of a cross-selling item. Finally the ROI becomes

$$ROI = \frac{P_{cs} \cdot (T \cdot S \cdot CS) + P \cdot T \cdot V \cdot B + P \cdot \sum_{t=0}^{T} E(L \text{ at time t}) - (C + (c \cdot T))}{(C + (c \cdot T))} \tag{8}$$

The ROI however does have some assumptions, as we assume the costs of customer acquisition and retention is not altered and thus is of no influence to the ROI. Furthermore we assume *E(new)* to be constant as well as the amount of visitors of the website. Another assumption is the independence of cross-selling with loyalty and the browsers that are turned into buyers. The assumptions probably will cause the ROI to be a little conservative as we would expect the number of visitors to increase along with a decrease in acquisition and retention costs.
Research on effects of recommender systems on sales is mostly limited to statements that they will increase sales and customer loyalty. However some papers do mention increase percentages that were noticed on a particular recommender system. For instance the net perceptions system (Konstan et. al. 2000) achieved an increase of 50 % in cross-selling success and 60% increase in the cross-sell value. Furthermore (Swearingen et al 2002) concluded that users of the Amazon's recommender systems would purchase 20% of the recommended items. (Reichheld et. al. 1990) claimed that a 5% improvement in customer retention could cause an increase in profits somewhere between 25% and 85%. Although

similar effects cannot be guaranteed for every type of E-business it is highly likely that an improvement of sales will occur.
An example of an excel sheet on the ROI calculation is given in the appendix. In figure 2.1 we can see that after only 3 years our investment has turned into a profit with the parameters given in the appendix.

Figure 2.1 Example of gain-costs for ROI calculation.



3 Data sets

Recommender systems make predictions based on data which is mostly provided by users, in this chapter the data format that is used in most recommender systems is described. We can separate the data that is used into 4 categories,

- Data about clients
- Data about products
- Sales data
- Ratings data

The first type of data is not easy to get as most customers are wary or unwilling to give information about themselves. If no registration is present in the system it becomes even more difficult if not impossible to get the necessary data. This method is however the most classical method of recommender systems. An example of information about clients is male or age. The second method is the most commonly used recommender system. This information is easy to get as products are mostly provided with all sorts of data, in the case of movies we can think of movie-type like science-fiction. The third type is also easy to get as the purchased items by customers are mostly stored somewhere by E-businesses. The last type is used by Netflix, this type stores ratings given by customers in a User–Item matrix. The Movielens dataset is also of this type an example is given in Figure 2.1

Figure 2.1 User-Item Ratings Matrix type 4

user \ Item	i1	i2	i3	i4	i5	…	ij	…	im
u1	$R_{1,1}$	$R_{1,2}$	$R_{1,3}$	$R_{1,4}$	$R_{1,5}$	…	$R_{1,j}$	…	$R_{1,m}$
u2	$R_{2,1}$	$R_{2,2}$	$R_{2,3}$	$R_{2,4}$	$R_{2,5}$	…	$R_{2,j}$	…	$R_{2,m}$
…	…	…	…	…	…	…	…	…	…
uk	$R_{k,1}$	$R_{k,2}$	$R_{k,3}$	$R_{k,4}$	$R_{k,5}$	…	$R_{k,j}$	…	$R_{k,m}$
…	…	…	…	…	…	…	…	…	…
un	$R_{n,1}$	$R_{n,2}$	$R_{n,3}$	$R_{n,4}$	$R_{n,5}$	…	$R_{n,j}$	…	$R_{n,m}$

In this matrix *u1,u2,…,un* are the *n* users and *i1,i2,…,im* are *m* items that can be rated by the users. $R_{k,j}$ is used to denote the rating of user *k* on item *j*. Users typically can rate items on a scale from 1 to 5 or from 1 to 10 depending on the preference of the business. A score of 1 would be bad and 5 good if a scale from 1 to 5 was chosen. $\overline{R}_i$ is used to denote the average rating of item *i* and $\overline{R}_u$ is the average of user *u*.
Type 3 is comparable to figure 2.1 except the ratings are replaced with binary numbers which represent bought (usually 1) and not bought (0). Recommender systems that use this type of data mostly also rely on background information on either an item or a user, thus a combination of type 3 and type 1 and or 2. Although some recommender systems make predictions solely on the user-items ratings matrix, recommender systems can also use the information of type 1 and 2 to improve the recommendations. Background information of the user, type 1, is usually also in a matrix form, see figure 2.2.
Here it is important to understand that users might not be entirely truthful about certain information. Hence a bias can occur if the information is based on the user's input.
For items the information is more reliable as it can be verified, the matrix of background information has the same format as figure 2.2, see figure 2.3.

Figure 2.2 Example Background user type 1

user	Age	Gender	city	…
u1	22	M		
u2	30	M		
u3	18	F		
u4	61	M		
u5	55	F		
u6	41	F		
u7	26	M		
…	…	…	…	
uk				
…	…	…	…	
un				

Figure 2.3 Example background Item type 2

item	genre	year	etc
I1	SF	2001	
I2	Action	2004	
I3	comedy	1996	
I4	SF	1980	
I5	Thriller	2006	
I6	SF	2001	
I7	Horror	2006	
…	…	…	…
Ik			
…	…	…	…
Im			

A problem in large datasets like the ones described above is data sparsity, users cannot rate every item as most probably thousands if not millions of products are available. In fact users usually only rate a very limited amount of items, say 1%. The missing data (commonly) is not used in the calculations as these would not add any information if not work contradictory to an algorithm.
Datasets that are mostly used in research are the Netflix dataset and Movielens. As most research has been done on these it is wise to use one of these datasets for future research. Movielens is a movie dataset which contains 6040 users and 3900 movies and is about 95% sparse. The Netflix dataset contains 480189 users and 17770 movies and 99% sparse. As we can see that Movielens is more user friendly when it comes to testing the number of users is much smaller and therefore takes less computational time, Netflix on the other hand is more realistic.

4 Algorithms

Recommender systems can have different setups; they can provide users with only a singular top item or with top n items. Some recommender system can even provide ratings for all items. After a company has decided to implement a recommender system it is just a matter of choosing the recommender system best suited for the intended use. A customer arriving on a website is referred to as an active user. This user will receive recommendations based on the comparison between his or her historical data and historical data of other users, in most cases ratings.
Aspects that also come into play are Accuracy, Sparsity tackling, Scalability and Complexity. The accuracy of the algorithm is important, especially when recommended items are not to the liking of the customer. Sparsity is the lack of information available in the data and the affects this has on the accuracy in the recommender system. In the case of the user-item matrix the data is sparse, because users do not rate all items, only a (small) selection of items. As we will see some recommender systems are better suited for tackling sparsity than others. Scalability is a term which is used to indicate if a system is able to process newly added information or users without (or positively) affecting the reliability, performance and availability. Finally complexity is a measure for the computational time needed to perform the necessary calculations.
In (Breese 1998) it is stated that 2 general approaches exist for recommender systems, Memory-based algorithms and Model-based algorithms. Model-based algorithms build a model when offline using the user-item matrix and background information on users or products as described in chapter 2. The estimated model is used to give predictions. Memory-based algorithms are used online, these algorithms use the full user-item ratings matrix to make predictions, i.e. no further data is required. Thus we can say memory based algorithms searches for similarities between users and or items to give predictions. Models that use memory based methods as well as model based are called hybrid models.
Most recommender systems that are described in the literature use the user-item ratings matrix. This type of recommender system provides the most information on user's tastes; the described recommender systems use this type of data.

4.1 Item to Item

The first Algorithm that we are going to discuss is of the memory-based type and uses an item to item nearest neighbour algorithm. This algorithm searches the user-item ratings matrix described in chapter 2 for similar items. The columns represent the different items and are rated by the users which are the rows. The most similar items are selected to make predictions of items the user most likely is interested in. Only if both items are rated by the same user will it be used in the calculation of the similarity. An example would be the computation of the similarity between items i and m in the user-item ratings matrix. In other words the customer will like items that are similar to items that the customer likes. For the computation of the similarity we first determine the users that rated both items. Different methods exist to compute the similarity between items. Three such methods are discussed in (Sarwar et.al. 2001). The first method is called Cosine based similarity. In cosine based similarity we think of every item as a vector of ratings in the user-items ratings matrix. The angle between vectors is the similarity between two items.

$$sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\left\|\vec{i}\right\|^2 * \left\|\vec{j}\right\|^2} \tag{9}$$

where $\vec{i}$ is vector of *i*. In the example this would be $R_{1,i}*R_{1,m} + R_{20,i}*R_{20,m}+R_{k,i}*R_{k,m}$ divided by the result of the square root of $(R_{1,i})^2 +(R_{20,i})^2 +(R_{k,i})^2$ times the square root of $(R_{1,m})^2 +(R_{20,m})^2+(R_{k,m})^2$
The second method is called correlation based similarity

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \overline{R}_i)(R_{u,j} - \overline{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i} - \overline{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \overline{R}_j)^2}} \tag{10}$$

This method calculates (Pearson-r) correlations between items, similar to the cosine based similarity we only take into account users that have rated both items, the similarity is then computed. The last method is the adjusted cosine similarity.

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \overline{R}_u)(R_{u,j} - \overline{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \overline{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \overline{R}_u)^2}} \tag{11}$$

After the similarity is computed the prediction is calculated. Two methods are discussed the, first the average of predicted rating.

$$P_{a,j} = \frac{\sum_{i \in I} R_{u,i} \cdot sim(i,j)}{\sum_{i \in I} sim(i,j)} \tag{12}$$

Here the predictive rating is computed by summarizing all the multiplications of the rating of every user that rated item *i* with the similarity of that user to active user *a*. We then divide with the sum over all used similarities. The predictions are then returned to the user as recommendations. The data that is used is extremely sparse, in other words users have rated only a small portion of items. These unrated items are not used in the similarity computations as is the case with all recommender systems. Furthermore since the number of items of most e-commerce businesses is big, sometimes over a million, we can state that sometimes it can be difficult to estimate predictions.
The second method uses Linear Regression. This method is very similar to the previous method but differs in the fact that it does not use $R_{u,i}$ rating values but estimated $R'_{u,i}$ from a linear regression model. The linear regression model has the following form

$$\overline{R}'_i = \alpha \overline{R}_n + \beta + \varepsilon \tag{13}$$

R_i is the vector of target item *i* ,R_n are the closest vectors of the most similar items, ε is the error and the parameter α and β are to be determined.
The item to item nearest neighbour algorithm has the advantage that it is scalable and notices little or no effects from the sparsity problem. The complexity of this algorithm is discussed in (Karypis 2000).The upper bound of the complexity for the model building phase is said to be $O(m^2n)$, but the actual complexity is much smaller because of the sparsity in the data. The complexity of an active user is described with $O(kq)$ where *q* is the number of items purchased by a user and *k* most similar items are needed to determine the *N* best recommendations. The accuracy of the algorithm is worse in comparison to other algorithms, however as we stated in chapter 2 the effects of more accuracy does not always constitute more increase in sales. Other factors are also important like transparency.

4.2 User to User

The user to user nearest neighbour algorithm is similar to the item to item algorithm except it searches for users that are similar to an active user. The idea is that an user will like items that other similar users like. The *k* most similar users are selected and used to predict items a user would most likely be interested in. Only the users that have rated at least *k* similar items are used for the calculations. Suppose we are a user a in the ratings matrix in chapter 2, this algorithm would compute the similarity with user *1,2,3…n*. The best similarities are used to rate items for the user that he hasn't rated yet. The user-user nearest neighbour algorithm is very similar to the item-item nearest neighbour algorithm, so the similarity is computed using the same methods, these are the Cosine similarity and Pearson correlation. The Pearson correlation is given below.

$$sim(a,u) = \frac{\sum_{i \in I}(R_{a,i} - \overline{R}_a) \cdot (R_{u,j} - \overline{R}_u)}{\sqrt{\sum_{i \in I}(R_{a,i} - \overline{R}_a)^2} \sqrt{\sum_{i \in I}(R_{u,i} - \overline{R}_u)^2}} \tag{14}$$

The second step is computing the predicted rating of items. This is done with help of the weighted average of neighbour's users ratings

$$P_{a,j} = \overline{R}_a + \frac{\sum_{u \in U}(R_{u,i} - \overline{R}_u) \cdot sim(a,u)}{\sum_{u \in U} sim(a,u)} \tag{15}$$

Figure 4.1 Example of User-Item Ratings Matrix

	i1	i2	i3	i4	i5	i6
u1		4		5	?	
u2	3		1			4
u3	3	4		5	5	
u4						1
u5	1	5	3	2		
u6				4	3	2

To give an example we look at user 1 in figure 4.1 and consider this to be our active user. The goal is to rate the question mark, thus item 5. User 1 is compared with every other user with help of the similarity computation and in this case user 3 is most similar to user 1. The Pearson correlation only uses items that both users have rated (the red ratings are used for the similarity computation between user 1 and 3). The similarity between user 1 and 3 is 1 and if only one similar user is used for predictions the question mark would be rated with a 5. The Pearson correlation will always range between 1 and -1, that is 1 for positive correlation, -1 for negative and 0 for no correlation. In most cases multiple users are taken into consideration, however for illustrative purposes we choose to only consider one user. This recommender system is again a memory based method which can be used in online recommendations. In comparison to the item to item nearest neighbour algorithm this algorithm performs better when considering accuracy. However the algorithm does have problems with the scalability and with data sparsity. Another drawback is that users with peculiar tastes could get bad recommendations using this algorithm. The complexity of this algorithm is similar to the item to item variant, with the exception that *m* and *n* are switched.

4.3 Clustknn

ClustKNN is a hybrid model between a memory and a model-based algorithm. This model uses clustering to tackle the scalability problem of the user-user KNN algorithm. After deciding the number of clusters preferable, a clustering algorithm creates clusters of users which will be used to compute the similarities and predictions for the different clusters (instead of every single user). The similarity and prediction computation is the same as user-user based, thus the only difference is that users have been replaced with surrogate users. Different approaches exist for determining the clusters. (Rashid et. al. 2006) describes the clustering algorithm called Bisecting k-means, the following steps are taken to form clusters:

1. Pick the largest cluster to split
2. Apply the basic k-means which produces 2 sub clusters
3. Repeat step 2 for the number of iteration that is needed and take the split with the highest overall similarity.
4. Repeat steps 1, 2 and 3 until the desired number of clusters is reached

Different clustering algorithms exist and other variants can be used instead, however it is questionable that a significant gain in accuracy will be achieved.
The accuracy of the algorithm is very high. Other methods which result in higher accuracies result only in relatively small differences in the accuracy. The sparsity and scalability problems are reduced if not none existent. Furthermore the complexity for online recommendations is also reduced to *O(m)*, the model building still uses approximately the same amount of time as the User based KNN algorithm.

4.4 Content-Boosted Collaborative Filtering

In (Melville et. al. 2002) another variant of a CF algorithm is discussed. Content-Boosted Collaborative Filtering (CBCF) is a method which reduces the sparsity of the dataset and returns fairly accurate predictions. The first step to take in CBCF algorithm is to change the user item matrix to a pseudo user-item matrix filled with values v_{ui} using the function below.

$$v_{u,i} = \begin{cases} R_{u,i} & :if\ user\ rated\ item\ i \\ c_{u,i} & :otherwise \end{cases} \tag{16}$$

Here R_{ui} is again the actual rating when present and c_{ui} is the prediction by the pure content-based system. In (Melville et. al. 2002) the pure content-based system is a bag of words Naïve Bayesian text classifier. The content would consist of background information on for instance movies. An example is titles, cast, genre etc. After the changing of the user-item matrix we compute the similarity between users using the pseudo matrix. Because the accuracy of a pseudo user is dependent on the number of items the user has rated a confidence measure can be implemented. The harmonic mean weighting is such a method

$$hm_{i,j} = \frac{2m_i m_j}{m_i + m_j} \tag{17}$$

$$m_i = \begin{cases} \frac{n_i}{50} : if \ \ n_i < 50 \\ 1 \quad : otherwise \end{cases} \tag{18}$$

Here n_i refers to the number of items user i has rated. Adding the significance weighting to $hm_{i,j}$ gives us the hybrid correlation weight.

$$hw_{a,u} = hm_{a,u} + sg_{a,u} \tag{19}$$

where

$$sg_{a,u} = \begin{cases} \frac{n}{50} & \text{if} \quad \text{n} \ \leq \ 50 \\ 1 & \text{else} \end{cases} \tag{20}$$

A confidence metric is also included for the active pseudo user's rating vector, the following weights it according to the number of items active user a has rated.

$$sw_a = \begin{cases} \frac{n_a}{50} \times \max & : if \ \ n_a < 50 \\ \max & : otherwise \end{cases} \tag{21}$$

Now that the weighting schemes are defined the adjusted prediction function can be presented,

$$P_{a,i} = \bar{v}_a + \frac{sw_a(c_{a,i} - \bar{v}_a) + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} sim(a,u)(v_{u,i} - \bar{v}_u)}{sw_a + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} sim(a,u)} \tag{22}$$

The accuracy of this algorithm is better than the accuracy of the user KNN algorithm and some other alternatives. Combined with solving of the sparsity problem this method seems to have other advantages. However the scalability remains a problem. The offline complexity of this algorithm becomes the complexity of the naïve Bayes classifier, thus *O(Np),* where *N* is the number of training examples and *p* the number of features. Online the complexity becomes *O(mn)*, since the similarities computation is the same as the user KNN algorithm.

4.5 Unified

A different approach to the memory based algorithms is the unified item and user based hybrid model. In (Wang et. al. 2006) a method for similarity fusion is presented. Three types of similarity are computed, item to item similarity, user to user similarity and a combination of these two. The similarity of user to user and item to item needs no further explanation, the similarity of the combination is calculated using the following

$$sim_{ui}(a,i,u,j) = \frac{1}{\sqrt{(1/sim_u(a,u))^2 + (1/sim_i(i,j))^2}} \tag{23}$$

Where sim_u(a,u) denotes user-user similarity and sim_i(i,j) denotes item-item similarity.
As we now have defined a way to calculate the different similarities, we now can move on to the prediction. To be able to define the accuracy of a prediction, or better stated a way to define better predictions; we need to implement a weighting scheme.

$$W_{a,i}^{u,j} = \begin{cases} \dfrac{sim_u(a,u)}{\sum_{R_{u,i} \in SUR} sim_u(a,u)} \lambda(1-\delta) & \mathrm{R}_{\mathrm{u,i}} \in SUR \\ \dfrac{sim_i(i,j)}{\sum_{R_{u,i} \in SIR} sim_i(i,j)} (1-\lambda)(1-\delta) & \mathrm{R}_{\mathrm{u,i}} \in \mathrm{SIR} \\ \dfrac{sim_{ui}(a,i,u,j)}{\sum_{R_{u,i}} sim_{ui}(a,i,u,j)} \delta & \mathrm{R}_{\mathrm{u,i}} \in \mathrm{SUIR} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

where

$$\begin{aligned} SUR_{a,i} &= \left\{ R_{u,i} \mid u \in S_u(a) \right\} \\ SIR_{a,i} &= \left\{ R_{a,j} \mid j \in S_i(i) \right\} \\ SUIR_{a,i} &= \left\{ R_{u,j} \mid u \in S_u(a), j \in S_i(i), u \neq a, j \neq i \right\} \end{aligned} \tag{25}$$

$S_u(a)$ denotes the top N similar users and $S_i(i)$ denotes the top N similar items
Here it is clear that we can choose to give a combination of user to user and item to item a better rating than a single user to user or item to item. We can choose to give either of the three methods more weight. The predictions is than computed with the following.

$$\hat{R}_{a,i} = \sum_{R_{u,j}} p_{a,i}(R_{u,i}) W_{a,i}^{u,j} \tag{26}$$

where

$$p_{a,i} = R_{u,j} - (\bar{R}_u - \bar{R}_a) - (\bar{R}_j - \bar{R}_i) \tag{27}$$

The complexity of this algorithm is approximately $O(m^2n + n^2m)$ since it uses both the user-user and item-item nearest neighbour algorithm. This method reduces the sparsity problem, as also item-item algorithm is used. The scalability is however still a problem. Accuracy has improved in comparison to the user-user and item-item nearest neighbour algorithms. However this is a good method that blended two memory based algorithms, we will further discuss blending in section 4.7.

4.6 SVD

Singular value decomposition is a model-based algorithm. Model-based algorithms have the advantage of being build offline, thus the complexity is often less when used online. However the building of the model is more time consuming. Further problems are that the model will have to be rebuild from time to time, as data is removed and implemented. To

have an accurate model, a large amount of data needs to be collected. This method is described in (Sarwar 2003).
Singular Value Decomposition is a technique which is used in linear algebra. Recommender systems use a variant of the same principle. We transform the user item ratings matrix (for this subchapter we will refer to it as *M*) into 3 distinct matrices.

$$M = U \sum V^* \tag{28}$$

Where *U* is a $m \times m$ matrix, $\sum$ a $m \times n$ matrix and V^* a $n \times n$ matrix. Since $\sum$ has only *r* non-zero entries on the diagonal, the effective dimensions for the three matrices are $m \times r$, $r \times r$ and $r \times n$.
After transforming *M* into these three matrices we may reduce $\sum$ to a *k x k* matrix, where $k \ll r$. The values on the diagonal in the $\sum$ matrix are sorted from large to small, with larger values having the most influence on the values in the *M* matrix. Therefore we can reduce it to *k* eigenvalues. If we reduces *U* and V^* accordingly, remove right side of *U* and bottom rows of V^*, we can retain an approximation of the original *M* matrix. The idea is that noise in the matrix is removed with help of this technique and that for this reason predictions will be more accurate.
The decomposition can be used to compute similarities between users, this process will be an offline process, and the actual prediction of items will be the online process. Cosine based similarity can now be used to determine similarities between the different users and make predictions accordingly. New users will get recommendations with help of the following,

$$u_k = u^* U_k \Sigma_k^* \tag{29}$$

Now cosine based similarity can be used to determine similarity between different users. This traditional method however has problems with missing values and simply replacing them with 0's will not work. As previously described the user-item ratings matrix mainly consists of these missing values and thus the method described so far is not a very feasible recommender system. However (Funk 2006) used a different method, which is referred to as matrix factorization, to compete in the Netflix prize.
(Takács 2008) also describes this method, although a slight improvement was made. We can think of the ratings matrix as a combination of two matrixes, basically every movie has features on among others type and duration, which we can limit to *k*. Thus we can think of every movie as a vector with *k* features. The same applies for users; every user consists of a vector of *k* preferences on features. A movie rating for a specific user would than be based on how well the vector of the user fits the movie features vector, i.e. the sum of multiplying both vectors. For instance if a user likes Science Fiction movies (3) and doesn't like Horror (-1), and an item has the features Science Fiction (1) and Horror (-1), we will get a rating 4.
In other words the user-item matrix *M* can be separated into two distinct matrixes which can give an approximation of the original matrix *M* when multiplied, thus

$$M \approx UV^* \tag{30}$$

where *U* is a m×*k* matrix and *V* is a k×*n* matrix. *M* consist of integer values, whereas *U* and *V* consist of real values. We should find Matrixes *U* and *V* that minimize the error caused by the differences between the known original ratings and the ratings resulting from matrix multiplication. This allows us to get ratings for every user on every item. We will refer to ratings in the matrix with R_{ai} The error is represented by

$$SE = \sum_{(a,i) \in \mathbb{R}} (R_{ai} - \tilde{R}_{ai})^2 \tag{31}$$

where $\tilde{R}_{ai} = \sum_{k=1}^{K} u_{ak} v_{ki}$. We want to minimize the standard error to obtain the optimal matrixes *U* and *V*.

$$(U,V) = \arg\min_{(U,V)} SE \tag{32}$$

We can than find the minimum with help of the gradient functions,

$$\frac{\partial}{\partial u_{ak}}(R_{ai} - \tilde{R}_{ai})^2 = -2(R_{ai} - \tilde{R}_{ai}) + V_{ki}$$
$$\frac{\partial}{\partial v_{ki}}(R_{ai} - \tilde{R}_{ai})^2 = -2(R_{ai} - \tilde{R}_{ai}) + u_{ak} \tag{33}$$

The weights need to be updated in opposite direction of the gradients, with help of

$$u'_{ak} = u_{ak} + \eta(2(R_{ai} - \tilde{R}_{ai})v_{ki})$$
$$v'_{ak} = v_{ak} + \eta(2(R_{ai} - \tilde{R}_{ai})u_{ki}) \tag{34}$$

where η is the learning rate. We can implement λ for regularization to prevent too large weights.

$$u'_{ak} = u_{ak} + \eta(2(R_{ai} - \tilde{R}_{ai})v_{ki} - \lambda u_{ak})$$
$$v'_{ak} = v_{ak} + \eta(2(R_{ai} - \tilde{R}_{ai})u_{ki} - \lambda v_{ak}) \tag{35}$$

The steps (35) are repeated until (31) has improved.
This method is one of the favourites as this method is highly accurate and the online complexity is only *O(m)* as all ratings are calculated offline and we only have to search through m users. Furthermore singular value decomposition is scalable and thanks to Simon Funk the sparsity problem has also been addressed. When the model has been build this is one of the fastest algorithms.

4.7 Model blending

Blending different methods in a lot of cases result in more accurate recommendations, as is seen in numerous examples in the Netflix prize. However we have to consider that in the Netflix prize the goal is to maximize the accuracy of recommendations, without taking into consideration the complexity or additional costs and gains. It is unlikely that relatively slight improvements will be noticed by customers; therefore blending might not always contribute to an increase in sales.
Furthermore we have to take into consideration that the transparency of the models is also important. If users do not understand the underlying models they are more likely to distrust them. If the choice is made to blend models it is best to blend models produced by different types of algorithms, for example blending SVD and the user-user nearest neighbour algorithm. Similar models would only come to the similar conclusions, whereas different types could contribute to an increase in accuracy.
Blending of models possibly could be done in different ways; however the most commonly is linear regression,

$$R_j = \sum_{i=1}^{m} R_{ij}^* \beta_i + \varepsilon_j \tag{36}$$

where β_i are the parameters that need to be determined, ε_j is the error term and R_{ij}^* is the rating given by recommender system *i* for user *j*.
(Paterek 2007) described this method in more detail and uses linear regression to blend multiple models. The advantage of this blending method is that we combine more than a hundred recommender systems if preferable. To use this method the first step is to draw a

random sample between 1,5 and 15% out of the data and use this as a test set. The remaining data will be in the training set and is used to train the different algorithms. When the error on the training set is minimized the algorithms will calculate the predictions on the test set. These are then combined with the linear regression weights that were found. (Koren 2008) also presented a slightly different model which combines matrix factorization with memory based recommender systems.

4.8 Cold start recommendations

The cold start problem is perhaps the most difficult and important problem to overcome. The cold start problem basically is the problem of having too little information at the starting point to make accurate predictions. To overcome this problem certain steps can be taken. Some of these methods will be discussed in this sub section, but first we must differentiate between 3 different cold start problems. First we have the new system where no information on preferences of users is present (if users are present at all).
The second problem is newly added users and thirdly we have the problem of newly added items. The first problem is the most problematic. If users receive bad recommendations it is very likely that users quit the system. Furthermore the system needs a lot of user (preferences) information to be able to give good recommendations.
Therefore it is highly likely that the system will take a long time to produce good recommendations if the problem is not dealt with appropriately. As the user-item ratings matrix is (almost) empty, this data will not be of any help. Therefore background information on users and items is needed. If information about the user is known this information can be used to classify a user into a category (i.e. a model based recommender system) and make predictions. Thus in case of an entirely new system a company could use its own employees to gather intelligence on the likes and dislikes of different classes of people. A new user can then be compared to the company employees and make predictions. Before putting the system to use it is also possible to gather information about the interests of the different users before implementation.
In other words make the possibility of rating items possible, but do not make predictions yet. Another possibility is to start with a recommender system that uses non-personalized methods. For instance these can return most popular items or most viewed. This way the system will have a better start and be fully operational in a shorter time span. When the system is fully operational cold start problems will still occur. When new items appear they would not be recommended because nobody has rated them yet.
A way to overcome this is to use another non-personalized recommender system for newly added items. This way customers will also come into contact with these items and be able to rate them. Another way is to use specifications of different products in the recommender system.
Newly added users can also experience problems with recommendations. We can again tackle this problem by using other non-personalized recommender systems.
Non-personalised methods are a feasible way to increase accuracy in the earlier stages of the implementation of the recommender system, an example of a Non-personalized recommender system is using top rated items as recommendations.
The recommendation of new items is also of importance, to overcome the problem items can be compared to each other.
Netflix uses retraining of the data; they update their data every week to allow new items and users in their recommender system. New items are implemented into the Netflix system with help of manual suggested similarities to other items. These are automatically removed over time when the number of ratings is sufficient for this item.

5 Other issues

5.1 Malicious users

Recommender systems are hugely dependant on outside information, this dependency creates the opportunity of profile injection attacks. A profile injection attack or shilling attack is an attack in which the goal of the attacker is to create a bias in the recommender system by inserting fake user ratings. (Zhang et. al. 2006) describes 2 types of profile injection attacks: nuke and push attacks. A nuke attack is inserted to decrease the number of times an item is recommended to different users, whereas push attacks increase the number of times an item is recommended, otherwise there is no discrepancy between the two. A general formal framework for profile injection attacks is presented by (Mobasher et.al. 2007) together with ways to reduce effects of profile injection attacks.
Profile injection attacks can be tackled in different ways, namely prevention, removal of injected profiles or reduction of the bias these profiles cause. Often registration to websites or collecting large quantities of data is not preferable for the user; however this does create a viable solution for the problem as this prevents malicious users to quickly insert profiles. If we have to work without registration we have to explore other means of identifying fake user profiles or at least try to control the bias created by the inserted profiles. Fake user profiles can be traced because the distribution of ratings is different in comparison to other users. Even if some real users are removed it could still help improve recommendations. (Zhang et. al. 2006) describes another method of tracking fake profiles.
In this paper fake user profiles were identified by using the time which profiles were added. Usually fake user profiles are added in fast quantities in short period of times. Because of this aspect we could also implement a mechanism which removes these profiles. The bases of the removal would be the difference in the distribution of the newly added profiles and the distribution in the active user database. Together with the time in which the profiles were added it should be a viable way to reduce the problem.
The last solution of the problem would be ways to reduce the bias caused by profile injection attacks. Instead of identifying profiles we could assign weights to user profiles which describes there reliability. If we use the most reliable profiles for recommendations the impact of profile injection attacks is reduced. Another method is to opt for recommender systems that are less vulnerable to these kinds of attacks.

5.2 Privacy

Privacy is considered as a very important aspect in e-commerce as there is legislation on the distribution of private information of users to third parties. To assure privacy of customers and the fact that the company is not liable for any lawsuits, steps need to be taken to assure that privacy related material cannot be subtracted from statistics and other used material in the recommender system. The following papers describe privacy issues in recommender systems (Canny 2002), (Ramakrisnan et. Al. 2001) and (Lee et. al. 2006).

5.3 Evaluation

It is important to be able to measure the performance of different algorithms as this gives us the possibility to asses what recommender system is best in certain types of situations. However we do have to keep in mind that some algorithms might perform better in some situations and less in others. Measures that are usually applied in the case of recommender

systems are MSE, MAE and MAPE. These measures are applied on commonly used datasets for testing the accuracy. The mean square error (MSE) is used the most and is defined as

$$MSE = E((\hat{\theta}_i - \theta_i)^2) \tag{37}$$

where θ_i is the actual rating of the user on an item and $\hat{\theta}_i$ is the estimated rating. An alternative that is used is the Mean Absolute Error (MAE),

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{\theta}_i - \theta_i\right| \tag{38}$$

The final method that is commonly used is the mean average percentage error (MAPE),

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\theta_i - \hat{\theta}_i}{\theta_i}\right| \tag{39}$$

A problem with these error measures is that the response rate is very low in user-item ratings matrix, typically around 1%. This causes the errors to be higher than preferable. Although this is the case these measures are still the best to use when determining the accuracy. A helpful way to determine how well a recommender system predicts items is to compare it to the performance of an average rating recommender system.

6 Discussion

Recommender systems can create a strategic advantage for companies that use them. Customer loyalty is increased along with customer satisfaction. As customers are more likely to return sales figures increase over time. The percentage of cross-selling also increases over time. Therefore businesses without recommender systems are more likely to be forced out of the market. The profitability of recommender systems can be modelled with help of a cost benefit analysis. We presented a simplified model of the ROI suited to some assumptions.

In this paper we discuss different types of recommender systems. It is difficult to state that a particular recommender system is better than others as relatively simple systems can also be cheaper to implement, but give lower accuracy. This however can be of little consequence if sales figures are not affected too much. Earlier research has also shown that the transparency of the system can also be of importance.

From the different recommender systems presented in the paper the method of blending the models gave the best accuracy. However it is not certain that the extra effort also results in additional increase in profits as other aspects are also important.

Although recommender systems are a viable way to increase sales for E-businesses there are some problems that need to be addressed before implementation. Virtually every recommender system will have problems when first implemented as there is only limited data available on users and items. It is important to improve the accuracy when first implementing the recommender system because bad recommendations to users can reduce the effect recommender systems have on sales.

Other potential problems in recommender systems can be caused by malicious users. When large quantities of fake profiles are inserted the accuracy of the recommender system could be affected. This could potentially lead to bad recommendations and thereby possibly contribute to a reduction of the effect recommender systems have on sales.

7 Bibliography

Breese, J., Heckerman, D., Kadie, C., (1998), Empirical analysis of predictive algorithms for collaborative filtering, 14th Conference on Uncertainty in Artificial Intelligence.

Brynjolfsson, E., Hu, Y., Smith, M.D., (2006), From Niches to Riches: The Anatomy of the Long Tail. *Sloan Management Review*, 47(4) 67-71.

Canny, J., (2002), Collaborative Filtering with Privacy, Berkeley

Fleder, D., Hosanagar, K., (2006), Blockbuster culture's next rise or fall:
The effect of recommender systems on sales diversity, University of Pennsylvania

Fleder, D., Hosanager, K., (2007), Recommender systems and their impact on sales diversity, *EC'07*, June 11-15, 2007, San Diego, California, USA., ACM 978-1-59593-653-0/07/0006

Funk, S., (2006), Netflix Try this at home, http://sifter.org/~simon/journal/20061211.html

Kano, N., (1984), Customer Satisfaction Model,
http://www.12manage.com/methods_kano_customer_satisfaction_model_nl.html

Karypis, G.,(2000),Evaluation of Item-Based Top-N recommendation algorithms, University of Minnesota, Minneapolis

Konstan, J.A., Riedl, J., (2000), Recommender systems in E-commerce tutorial notes, University of Minnesota

Koren, Y., (2008), Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model, ACM 978-1-60558-193-4/08/08

Lee, C.H., Hwang, J., (2006),Private Information Shielding Service for Overcoming Privacy Risk in Recommender System, Seoul National University, Seoul, Korea

Loyalty business model: http://en.wikipedia.org/wiki/loyalty_business_model

Luarn, P., Lin, H.H., (2003), A Customer Loyalty Model for E-service Context, Journal of Electronic Commerce Research, Volume 4, No.4

Marlin, B., (2004), Collaborative Filtering: A Machine Learning Perspective, thesis, University of Toronto

Melville, P.,Mooney, R.J., Nagarajan, R. ,(2002),Content-boosted collaborative filtering for improved recommendations , University of Texas, Austin

Mobasher, B., Burke, R., Bhaumik, R., Williams, C., (2007), Towards Trustworthy Recommender systems: An Analysis of Attack Models and Algorithms Robustness, DePaul University, Chicago

Mooney,R.J., Roy,L., (1999), Content-based book recommending using learning for text categorization, In SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation.

Osterwalder, A., Pigneur, Y., Tucci, C.L., (2005), Clarrifying Business Models: Origens, Present, and Future of the Concept, communications of AIS, Volume 15

Paterek, A., (2007), Improving regularized singular value decomposition for collaborative filtering, Warsaw University, Poland, ACM 978-1-59593-834-3/07/0008

Ramakrisnan, N., Keller, B.J., Mirza, B.J., Grama, A.Y., Karypis, G., (2001), Privacy Risks in Recommender Systems, 1089-7801/01 IEEE Internet Computing

Rashid, A.M., Lam, S.K., Karypis,G., Riedl, J., (2006), ClustKNN: A Highly Scalable Hybrid Model & Memory Based CF Algorithm, University of Minnesota, Minneapolis

Reichheld, F., Sasser, W., (1990), Zero defects: quality comes to services, Harvard Business Review, Sept-Oct, 1990, pp 105-111

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

Sarwar, B., Karypis, G., Konstan, J., Riedl, J.,(2001), Item-based Collaborative filtering recommendation algorithms, University of Minnesota, Minneapolis, ACM 1581133480/ 01/0005.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J., (2003), Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems, University of Minnesota, Minneapolis

Schafer, J., Konstan, J.A., Riedl, J., (1999), Recommender systems in e-commerce, In Proceedings of the ACM Conference on Electronic Commerce, p. 158-166

Schafer, B.J., Konstan, J.A., Riedl, J., (2002) Meta-recommendation Systems: User-controlled Integration of Diverse Recommendations, CIKM'02 November 4-9

Schafer, B.J., Konstan, J., Riedl, J., (2003), Recommender Systems in E-commerce, university of Minnesota, Minneapolis

Swearingen, K., Sinha, R., (2002), Interaction design for recommender systems, University of California, Berkeley, CA

Takács, G., Pilászy, I., Németh, B., (2008), Major components of the Gravity Recommendation System, SIGKDD Explorations Volume 9 issue 2

Tintarev, N., Masthoff, J., (2007), A Survey of Explanations in Recommender Systems, University of Aberdeen, Scotland, U.K.

Wang, J., Vries, A.P. de, Reinders M.J.T.,(2006), *unifying User-based and Item-based Collaborative Filtering Approaches by similarity fusion*, in SIGIR06,August 6-11, 2006, Seattle, Washington, ACM 1-59593-369-7/06/0008

Williams, C., Mobasher, B., (2006), Profile Injection Attack Detection for Securing Collaborative Recommender Systems, DePaul University, Chicago

Zhang, S., Chakrabarti, A., Ford, J., Makedon, F., (2006), Attack Detection in Time Series for Recommender Systems, Dartmouth College, ACM 1-59593-339-5/06/0008

Appendix

B1	C1	D1	E1	F1	G1

Parameters:

C	200000		G4
c	10000		G5
V	100.000		G6
v	90000		G7
S	100000		G8
CS	5,00%		G9
B	0,50%		G10
Pcs	5		G11
P	10		G12
T	10		G13
L	55,00%	(previously 50%, thus 5% increase)	G14
E(NEW)	50000		G15

T	Sales	Sales Loyalty	G18
0	100000	50000	G19
1	105000	55000	G20
2	107750	57750	G21
3	109263	59263	G22
4	110094	60094	G23
5	110552	60552	G24
6	110804	60804	G25
7	110942	60942	G26
8	111018	61018	G27
9	111060	61060	G28
10	111083	61083	G29

T	Loyalty	cross-selling	visitors into buyers	G31
0				G32
1	5000	5250	500	G33
2	7750	5388	500	G34
3	9263	5463	500	G35
4	10094	5505	500	G36
5	10552	5528	500	G37
6	10804	5540	500	G38
7	10942	5547	500	G39
8	11018	5551	500	G40
9	11060	5553	500	G41
10	11083	5554	500	G42

T	cost	Gain		G44
0	200000		-200000	G45
1	10000	81250,00	-128750,00	G46
2	10000	109437,50	-29312,50	G47
3	10000	124940,63	85628,13	G48
4	10000	133467,34	209095,47	G49
5	10000	138157,04	337252,51	G50
6	10000	140736,37	467988,88	G51
7	10000	142155,00	600143,88	G52
8	10000	142935,25	733079,14	G53
9	10000	143364,39	866443,52	G54
10	10000	143600,41	1000043,94	G55

Appendix

B1	C1	D1	E1	F1	G1

Parameters:

C	200000		G4
c	10000		G5
V	100000		G6
v	=90000		G7
S	100000		G8
CS	0,05		G9
B	0,005		G10
Pcs	5		G11
P	10		G12
T	10		G13
L	0,55	(previously 50%, thus 5% increase)	G14
E(NEW)	50000		G15

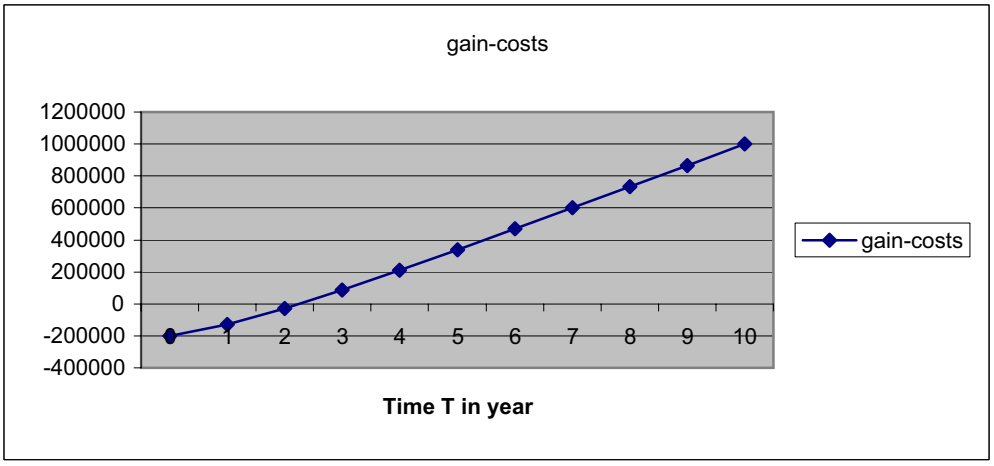
T	Sales	Sales Loyalty	G18
0	=B8	50000	G19
1	=$B$15+C20	=B14*B8	G20
2	=$B$15+C21	=B20*$B$14	G21
3	=$B$15+C22	=B21*$B$14	G22
4	=$B$15+C23	=B22*$B$14	G23
5	=$B$15+C24	=B23*$B$14	G24
6	=$B$15+C25	=B24*$B$14	G25
7	=$B$15+C26	=B25*$B$14	G26
8	=$B$15+C27	=B26*$B$14	G27
9	=$B$15+C28	=B27*$B$14	G28
10	=$B$15+C29	=B28*$B$14	G29

T	Loyalty	cross-selling	visitors into buyers	G31
0				G32
1	=B20-$B$19	=B20*$B$9	=$B$6*$B$10	G33
2	=B21-$B$19	=B21*$B$9	=$B$6*$B$10	G34
3	=B22-$B$19	=B22*$B$9	=$B$6*$B$10	G35
4	=B23-$B$19	=B23*$B$9	=$B$6*$B$10	G36
5	=B24-$B$19	=B24*$B$9	=$B$6*$B$10	G37
6	=B25-$B$19	=B25*$B$9	=$B$6*$B$10	G38
7	=B26-$B$19	=B26*$B$9	=$B$6*$B$10	G39
8	=B27-$B$19	=B27*$B$9	=$B$6*$B$10	G40
9	=B28-$B$19	=B28*$B$9	=$B$6*$B$10	G41
10	=B29-$B$19	=B29*$B$9	=$B$6*$B$10	G42

T	cost	Gain		G44
0	=B4		=C45-B45	G45
1	=$B$5	=B33*$B$12+D33*$B$12+C33*$B$11	=D45+C46-B46	G46
2	=$B$5	=B34*$B$12+D34*$B$12+C34*$B$11	=D46+C47-B47	G47
3	=$B$5	=B35*$B$12+D35*$B$12+C35*$B$11	=D47+C48-B48	G48
4	=$B$5	=B36*$B$12+D36*$B$12+C36*$B$11	=D48+C49-B49	G49
5	=$B$5	=B37*$B$12+D37*$B$12+C37*$B$11	=D49+C50-B50	G50
6	=$B$5	=B38*$B$12+D38*$B$12+C38*$B$11	=D50+C51-B51	G51
7	=$B$5	=B39*$B$12+D39*$B$12+C39*$B$11	=D51+C52-B52	G52
8	=$B$5	=B40*$B$12+D40*$B$12+C40*$B$11	=D52+C53-B53	G53
9	=$B$5	=B41*$B$12+D41*$B$12+C41*$B$11	=D53+C54-B54	G54
10	=$B$5	=B42*$B$12+D42*$B$12+C42*$B$11	=D54+C55-B55	G55

ROI:	=(SOM(C46:C55)-SOM(B45:B55))/SOM(B45:B55)

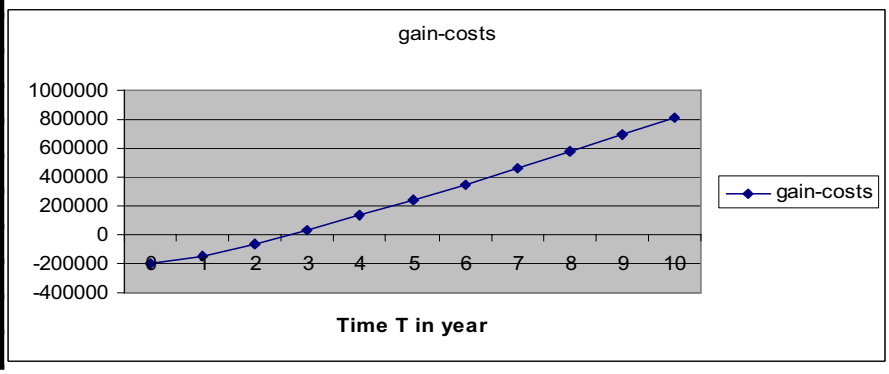
ROI:	3,33



Other scenarios

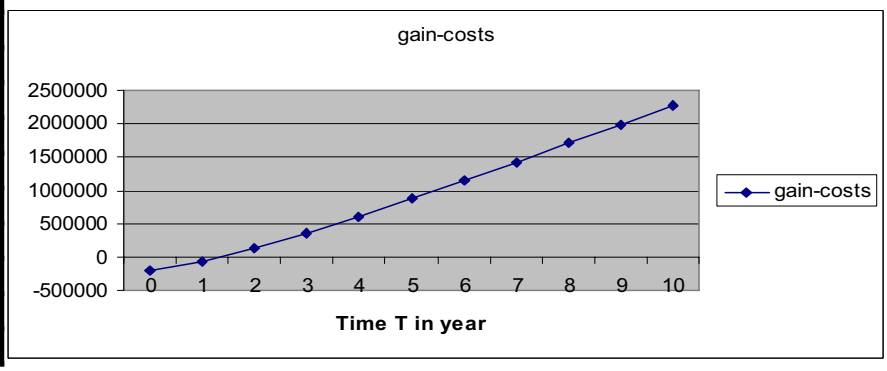
C	200000
c	10000
V	100000
v	90000
S	100000
CS	0,02
B	0,002
Pcs	5
P	10
T	10
L	0,55
E(NEW)	50000

ROI:	2,6846972



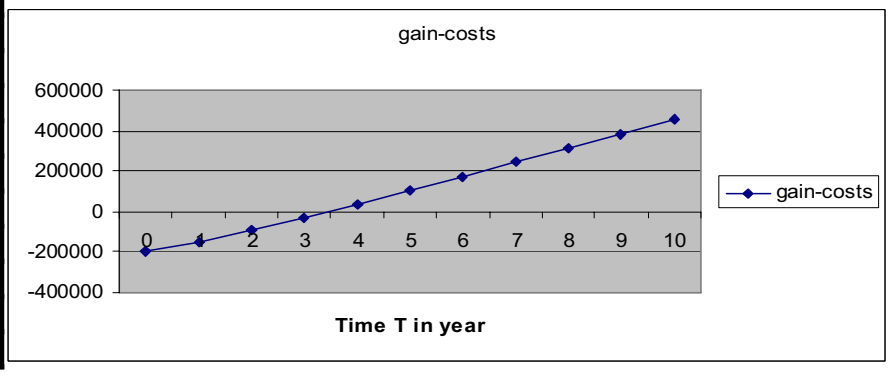
C	200000
c	10000
V	100000
v	90000
S	100000
CS	0,07
B	0,002
Pcs	5
P	10
T	10
L	0,6
E(NEW)	50000

ROI:	7,5724061



C	200000
c	10000
V	100000
v	90000
S	100000
CS	0,07
B	0,002
Pcs	5
P	10
T	10
L	0,52
E(NEW)	50000

ROI:	1,5153293



C	200000
c	10000
V	100000
v	90000
S	100000
CS	0,01
B	0,001
Pcs	5
P	8
T	10
L	0,52
E(NEW)	50000

ROI:	0,1904413