Chapter 13

Recommendations Worth a Million

"Data! Data! Data!" he cried impatiently. "I can't make bricks without clay."

– Sir Arthur Conan Doyle in *The Adventures of Sherlock Holmes*

Contents

In October 2006, Netflix announced that they would award a one million dollar prize to the first developer of a recommendation algorithm that could do a better job than their current system at recommending movies to users. The executives at Netflix believed in the value of great recommendations so much that they were willing to award a large sum of money for a better algorithm. Three years after the prize was announced, a seven person team of statisticians, machine-learning experts, and computer engineers were finally awarded the prize.

The Netflix contest was closely followed by researchers around the world because recommendation systems are used for much more than suggesting movies. Amazon, the world's largest online retailer, uses an intricate recommendation system for almost any item imaginable. eHarmony, an online dating site, uses a recommendation system to recommend compatible matches to single men and women. With the large number of choices available, recommendation systems help users to quickly find what they might be interested in.

In this chapter, we discuss how three popular companies – Netflix, Amazon, and eHarmony – have implemented recommendation systems to accurately predict their users' interests. An effective recommendation system provides an edge over competitors by increasing sales and improving the satisfaction of customers. The best recommendation systems use massive datasets efficiently and appropriately.

13.1 Getting Advice from Data

When a customer walks into a store, a sales representative often asks, "Can I help you find something?" This question is targeted toward customers who know what they want; they have come into the store looking for a specific item, and they just need to locate it. Similarly, almost every online store has a search tool that can be used to find specific items in the store. But what about the customer who has just come to the store to browse? They are interested in buying something, but they do not have a specific item in mind. Ideally, a worker at a store could instantly make recommendations for someone based on their previous purchases and the previous purchases of similar customers. This is hard to do without the help of a computer, since it requires massive amounts of knowledge about customers, their previous purchases, and purchases by other customers with similar interests.

Online recommendation systems develop recommendations by utilizing massive datasets, which many online retailers or service providers now have at their disposal. A recommendation system uses an information filtering device to recommend items or other services to users. This allows companies to search through large databases of items or options to pick out the ones that a user will most likely be interested in. This is beneficial to the company and to customers; the company is able to identify potential customers for its products, and customers can more easily find items that they may find sufficiently interesting to buy.

Creating a good recommendation system has many challenges in addition to collecting the huge amount of data needed. Often, the recommendations have to be available in real-time. This means that if a user gives a rating or preference for an item, the system should be able to immediately use that information to select other candidates which are of interest to a buyer. The system must produce recommendations that the user will find helpful. If a user is constantly getting recommendations for things that they are not interested in, they might get frustrated and the company could lose a customer. Additionally, different customers have different amounts of information available. The system should work with users for whom the system has very little information on ratings or purchases, as well as high activity users.

13.2 Types of Recommendation Systems

There are two main types of recommendation systems: collaborative filtering and content-based filtering. Collaborative filtering systems make recommendations based on the attributes of users, while content-based filtering systems make recommendations based on the attributes of items. While some recommendation systems are one type or the other, there are also many systems that combine these approaches to improve functionality. In this section, we will discuss basic collaborative filtering and content-based filtering systems as an introduction to later sections, where we will discuss some of the most successful recommendation algorithms in use today.

Collaborative Filtering

In collaborative filtering systems, each user is represented by a vector of items. The length of the vector is the number of distinct items that can be offered to, rated by, or purchased by the user. Each entry in the vector gives the customer's rating for the item, if the customer has rated the item. This means that the vectors often have many empty entries; most customers will have rated only a small fraction of the total number of items available.

Collaborative filtering uses a measure of similarity between users to make rating predictions. There are many different ways of measuring similarity, but one popular method is to use the correlation between the item vectors of the users. The similarities are used as weights in predicting ratings for a user. If we predict that a user will like an item, we will recommend it to him or her.

As an example, suppose we have five different customers (Amy, Bob, Carl, Dan, and Eva) and we are looking at their ratings (out of 5) for four different movies. Table 13.1 gives the ratings, where users and movies are denoted by letters. The blank entries indicate that the person has not rated the movie.

Let $r_{i,j}$ be the rating that user i gives to movie j. Thus, $r_{A,F} = 5$ since Amy (A) gave a rating of 5 to the movie *Forrest Gump* (F). Let μ_i be the average rating of user i, and σ_i be the standard deviation of the ratings of customer i. In our example, the average rating for Amy would be

Table 13.1: Example user ratings.

User	Forrest Gump (F)	Godfather (G)	Inception (I)	Jaws (J)
Amy (A)	5		4	3
Bob (B)	3	5	2	5
Carl (C)		3	5	4
Dan (D)	4	5	4	
Eva (E)	4	4		3

Table 13.2: User means and standard deviations for ratings.

User	Mean Rating	Std Dev Rating
Amy (A)	4.00	0.82
Bob (B)	3.75	1.30
Carl (C)	4.00	0.82
Dan (D)	4.33	0.47
Eva (E)	3.67	0.47

$$\mu_A = \tfrac{5+4+3}{3} = 4,$$

and the standard deviation for Amy would be

$$\sigma_A = \sqrt{\tfrac{(5-4)^2+(4-4)^2+(3-4)^2}{3}} = 0.82.$$

Table 13.2 gives the mean and standard deviation of the ratings for each of the five users. (Note that, for simplicity, we are dividing by n in the computation of standard deviation, instead of $(n-1)$ as is done when computing an unbiased estimate of the population standard deviation.[1])

We now want to compute the similarity between any two users. In this example, we will use correlation as a measure of similarity, but different recommendation systems use different measures of similarity. Using the definition of correlation, we define the similarity between users A and B as:

$$s_{A,B} = \frac{\frac{1}{n}\displaystyle\sum_{i=1}^{n}(r_{A,i}-\mu_A)(r_{B,i}-\mu_B)}{\sigma_A\sigma_B},$$

[1]When calculating the standard deviation of a sample, $(n-1)$ is often used in the denominator instead of n, to produce what is called an "unbiased estimate of the population standard deviation." For this application, either approach is valid. Note that the choice of n or $(n-1)$ in the denominator will only make a material difference when n is small.

where n is the number of items (in this case, n = 4). We have to adjust this definition slightly to only average over the items that both users have rated, since the rating for an item that a user has never rated is undefined. Thus, we will just replace n with number of items that both users have rated. Note that this may lead to similarities between users that are larger than 1 or smaller than -1 (unlike standard correlation). However, the interpretation remains the same. Two users with a similarity close to 1 have positively correlated (similar) preferences, two users with a similarity close to -1 have negatively correlated (opposite) preferences, and two users with a similarity close to zero do not have very correlated preferences at all.

As an example, the similarity between Amy (A) and Bob (B) in our example case is:

$$s_{A,B} = \frac{\frac{1}{3}[(5-4)(3-3.75)+(4-4)(2-3.75)+(3-4)(5-3.75)]}{0.82 \times 1.30} = -0.63.$$

Note that there are only three terms in the numerator since Amy has never rated *The Godfather*, and so they have both rated only three movies.

Table 13.3 gives the similarities between all pairs of users. Each entry in the table is the similarity between the user corresponding to the row and the user corresponding to the column. Note that each user has a similarity of one with themselves.

Once we know the similarities between all users, we can predict a rating for a user. The prediction for user A and item i is given by:

$$p_{A,i} = \mu_A + \frac{\sum_{u \in U_i} s_{A,u}(r_{u,i} - \mu_u)}{\sum_{u \in U_i} s_{A,u}},$$

where each sum is over all other users u who have rated item i (denoted by the set U_i). This takes the average rating for user A, and adjusts it based on the ratings of similar users. The numerator of the fraction adds up the ratings of all other users for item i (relative to their average rating), weighted by their similarity with user A. Thus, if a similar user liked the movie more than their average rating or if a dissimilar user liked the movie less than their average rating, we would add a positive factor to user A's rating. On the other hand, if a similar user liked the movie less than their average rating or if a dissimilar user liked the movie more than their average rating, we would add a negative factor to user A's rating. The denominator of the fraction normalizes this sum by the total similarity between user A and all other users who have rated item i. So the prediction for a selected user can be thought of as an average of other user's ratings, weighted by the similarities of these users to the selected user.

As an example, we will predict Eva's rating for *Inception*:

$$p_{E,I} = 3.67 + \frac{1.30\times(4-4)-0.37\times(2-3.75)-0.43\times(5-4)+0.25\times(4-4.33)}{1.3-0.37-0.43+0.25} = 3.85.$$

Table 13.3: Similarities between all users.

	Amy (A)	Bob (B)	Carl (C)	Dan (D)	Eva (E)
Amy (A)	1	-0.63	0.00	-0.43	1.30
Bob (B)	-0.63	1	-0.94	0.91	-0.37
Carl (C)	0.00	-0.94	1	-1.30	-0.43
Dan (D)	-0.43	0.91	-1.30	1	0.25
Eva (E)	1.30	-0.37	-0.43	0.25	1

Therefore, we predict that Eva will like *Inception* more than her average rating, but not by much. Depending on our criteria for recommending movies, we may or may not recommend *Inception* to Eva.

This method works very well, but involves a significant amount of computation when there are many users and items. Different ways of addressing this problem exist, such as randomly sampling users, ignoring unpopular items, and clustering the items by type first.

It can also be challenging to get accurate results with collaborative filtering when the number of items is significantly larger than the number of users, due to the fact that it is highly unlikely that multiple users will have rated the same items. More generally, this is referred to as a *sparsity* problem. When the available data is very sparse, content-based filtering, which we discuss next, often performs better.

Content-based Filtering

Just as collaborative filtering made recommendations by finding similar users, content-based filtering works by finding "similar" items; given a pair of similar items, if the user likes one of the items, then the other item should be recommended to that user. Similarity can be defined in different ways, but it is typically determined by analyzing attributes of the items, such as the genre of a movie, keywords on a webpage, or harmony of a song. This differs from collaborative filtering methods since we are not looking at whether or not other users like these items; we only consider the information about the items themselves and the user we are making recommendations to. For example, if a customer buys the book *Moneyball* by Michael Lewis, the recommendation system might suggest other books by Michael Lewis or other books about statistics and sports.

Computing similarity between items is often very similar to how we computed the similarity between users in the previous section, but the vectors are of item attributes instead of user ratings. If we are trying to find the similarity between two movies, we might define a vector for each of them that describes things like the genre, director, main actors and actresses, year, etc.

This requires someone to define the attributes for each item, which could be subject to human judgment.

Once the similarity between the items is computed, an item is recommended to a user if it is similar to an item that they indicated they liked. For example, if content-based filtering is used for a movie system, it might recommended *Toy Story 2* to someone who indicated that they liked *Toy Story*.

Content-based filtering often works better than collaborative filtering when the user has not purchased or rated very many items. In content-based filtering, items can be recommended as soon as the user indicates that they like a single item, whereas in collaborative filtering, the user needs to be linked to other similar users, which requires significantly more data. But when the user has rated or purchased thousands of items, content-based filtering has a hard time making recommendations that use all of the known information. This is because each item that the user likes has many similar items, and it becomes difficult to know which ones to recommended at a given time. Additionally, it is difficult to make recommendations in an area not present in the user's previous history. In collaborative filtering, we could recommend an item to someone that might be very different from what they have previously purchased or rated because a similar user liked it. But in content-based filtering, the users themselves have to purchase or rate similar items for an item to ever get recommended. This limits the extent to which the recommendation algorithm could get customers interested in buying items that they might not think to purchase on their own.

13.3 Netflix

Being able to rent movies is a nice luxury. You can watch the entire movie in the comfort of your own home at a fraction of the price of buying the movie or going to the movie theater. Before Netflix was founded in 1997, the only places to rent movies were rental stores. A customer could rent the movie for a fixed number of days, usually somewhere between 1 day and 5 days. If they returned the movie late, they would be charged late fees. The founders of Netflix, Reed Hastings and Marc Randolph, felt that this business model was flawed. DVDs are easily lost under stacks of magazines, piles of mail, or even forgotten in the DVD player. Before long, late fees have accumulated and the cost of the rental has become more than the cost of just buying the DVD in the first place. This inspired Hastings and Randolph to create a new business model, and a new way of renting movies.

The website netflix.com was launched in April 1998 and started as an online version of the traditional pay-per-rental model used in movie rental stores. In September 1999, they introduced the monthly subscription concept still in use today; for a flat monthly fee, customers could get unlimited rentals without due dates, late fees, or shipping and handling fees. As soon as a customer mailed back the movie he currently had, he would be sent a new one.

Table 13.4: Financial Growth of Netflix.

Year	Revenue (in mil. $)	Net Income (in mil. $)	Market Cap (in mil. $)
2002	153	-22	267
2003	272	7	741
2004	506	22	1,260
2005	682	42	996
2006	997	49	1,770
2007	1,210	67	1,420
2008	1,360	83	1,590
2009	1,670	116	1,940
2010	2,160	161	3,060
2011	3,200	226	10,430
2012	3,610	17	5,280
2013	4,370	112	6,290
2014	5,505	267	22,340

Netflix now offers on-demand internet streaming of many titles, as well as over 100,000 titles on DVD.

In September 2002, four years after launching their website, Netflix had 670,000 subscribers. By September 2006, they had reached 5.6 million subscribers, and in July 2014 they announced over 50 million subscribers in 40 different countries. Table 13.4 lists the revenue, net income, and market capitalization of Netflix for the years 2002-2014. After taking a hit to Net Income in 2012 (attributed to a change in the subscription policy), Netflix is on the rise again.

Cinematch

Netflix introduced its recommendation system, called Cinematch, in 2000. Users of Netflix are asked to give a rating, from one to five stars, for any movie they have watched. These ratings are used to power the recommendations.

The recommendation system helps to keep more of the movie library in circulation since people receive recommendations from the entire library of movie titles, not just recent or mainstream movies. This gives independent films and movies that were not very successful in the theaters a wider distribution. Netflix has reported that 60% of subscribers add recommended movies to their list of movies to watch.

According to Netflix, the original Cinematch algorithm used "straightforward statistical linear models with a lot of data conditioning." They claimed that the predictions from Cinematch were accurate within half a star 75% of the time, and half of Netflix users gave five stars (the best possible rating) to movies that they rented after a recommendation.

The Netflix Prize

In October 2006, Netflix announced that it would award a one-million-dollar prize to the first developer of a recommendation algorithm that could beat Cinematch at predicting customer ratings by more than 10% (in terms of root mean squared error) on a data set. Netflix provided the competitors with a training set of over 100 million ratings that 480,189 customers gave to 17,770 different movies. Each entry included the user ID, movie ID, date of the rating, and the rating (from one to five stars). This data set was much larger than previous recommendation data sets that had been made available to researchers in the field of collaborative filtering and recommendation systems.

The contestants had to improve the predictive accuracy on a qualifying set of over two million ratings, for which the true results were only known to the judges of the competition. This contest was intriguing to both Netflix and recommendation system researchers; if someone claimed the prize, Netflix would have a much improved recommendation system and increased publicity, and the winner would win one million dollars and become well known in the field of recommendation algorithms.

Six days after the contest went live, a team had already submitted an algorithm that beat Cinematch. By June 2007, eight months after the contest started, over 20,000 teams had registered from over 150 different countries. At this point, teams started joining forces to improve their algorithms even more.

On June 26, 2009, almost three years after the competition started, the team "BellKor's Pragmatic Chaos" submitted a 10.05% improvement over Cinematch, signaling the end of the contest. However, Netflix decided to announce a "last call," which meant that teams had 30 days to make additional submissions. With one day left, the team "The Ensemble" submitted a 10.09% improvement, and by the time the competition officially ended, "BellKor's Pragmatic Chaos" had also submitted a 10.09% improvement. Netflix then tested the algorithms on a private test set, which the competitors had never seen before.

On September 18, 2009 the million dollar prize was awarded to "BellKor's Pragmatic Chaos." This team was composed of two researchers from AT&T labs, two Austrian researchers from Commendo Research & Consulting, one researcher from Yahoo!, and two Canadian researchers from Pragmatic Theory. This team combined many different recommendation strategies in their final algorithm, including nearest neighbors and matrix factorization, which we will focus on here. They also discovered unusual characteristics of users, including the fact that people tend to rate movies differently on Fridays versus Mondays, and that ratings can be very different depending on how long ago the user saw the movie.

It was the combination of many different ideas that made the winning algorithm successful. This is a general principle of analytics: successful systems often use sophisticated infrastructure to combine a variety of approaches.

Nearest Neighbors

The Nearest Neighbors method is similar to content-based filtering in that ratings are predicted based on the similarity between items.

We can again define the rating of user A for item i by $r_{A,i}$ and the similarity between items i and j by $s_{i,j}$. The similarity between items can be computed in many different ways, and typically depends on the type of item. For example, if we are rating movies, then two movies would probably have a higher similarity score if they are in the same genre, or have the same director.

Of all of the items that a user has rated, a certain number of the most similar items are selected and used to predict the user's rating for a new item. The group of items that are considered the most similar to an item are called the *neighborhood* of that item. Thus, predicted ratings do not consider other users, just similar items that a particular user has rated. Using this approach, the predicted rating for user A and item k is:

$$r_{A,k} = \frac{\sum\limits_{j \in N(A;k)} s_{j,k} r_{A,j}}{\sum\limits_{j \in N(A;k)} s_{j,k}},$$

where $N(A;k)$ is the set of neighbors of item k that have been rated by user A. The neighbors depend on the particular recommendation system. For example, the neighbors of the movie *The Shawshank Redemption* probably would include other movies based on books by Stephen King, other movies directed by Frank Darabont, or other movies starring Morgan Freeman.

As in collaborative filtering, there are various methods that can be used to define the similarity between two items. There are also variations of this method that weight the neighbors differently. If two neighbors of an item are very similar themselves, it is almost like the same movie is being counted twice. For example, the three movies in *The Lord of the Rings* trilogy are all considered similar to *The Chronicles of Narnia: The Lion, The Witch, and The Wardrobe*, and would probably all be considered neighbors with very similar similarity ratings. Additionally, most people probably give similar ratings to the three movies in the trilogy (when compared to all other movies in the database), and so it is like there are three identical terms in the numerator for the predicted rating of *The Chronicles of Narnia* movie. This gives the trilogy a weight that is three times larger than the weight of any other neighbor. Since there are three movies, some people could argue that this is accurate, but since the movies all belong together, other people could argue that it is unfairly weighted. Situations like this produce different variations of the Nearest Neighbors method.

Matrix Factorization

Many submissions for the Netflix Prize used the method of *matrix factorization*, including the team that ultimately won the million dollar prize. This method represents both items and users by vectors of a fixed length. The length of these vectors is much smaller than either the number of items (like in collaborative filtering) or the number of users. Instead, each entry of the vector represents some possible characteristic of the item, like in content-based filtering. For a movie, the entries in the vector might be whether the movie is a drama or comedy, the amount of violence, the rating, etc. The values in the vectors for users are weights that represent how much they like high values in that component of the vector. Assume for now that the user's weights are numbers between 0 and 1, where 0 means that they do not like movies with high numbers in that entry at all, and 1 means that they love movies with high numbers in that entry.

As a small example, suppose our items are movies and we have vectors with three entries: the amount of violence on a scale of one to five, whether the movie is a drama or comedy (a value between 0 and 1 with 0 representing a dramatic movie with no comedy, and 1 representing a funny movie without drama), and the popularity/experience of the cast (on a scale of 1 to 10, with 1 being a movie with an unknown cast, and 10 being a movie with a very well-known cast). A possible vector for *The King's Speech* could be $(1, 0.1, 8)$, since it had very little violence, was mostly a drama, and mostly had a very well-known cast. The movie *Pulp Fiction* could have a vector $(5, 0.5, 9)$ since it is a violent movie with comedy and drama, and also had a very popular cast. Similarly, we can imagine different vectors for users. A person who likes violent dramatic movies and thinks a better cast indicates a good movie might have a vector $(0.9, 0.1, 0.75)$, whereas a person who prefers comedies and doesn't really care about the cast might have a vector $(0.2, 0.9, 0.2)$.

The amount a user might like an item is predicted by taking the sum of the corresponding terms of the vectors multiplied together (the inner product of the vectors). Thus, we predict that a person with vector $(0.9, 0.1, 0.75)$ will give a rating of $(0.9 \times 1) + (0.1 \times 0.1) + (0.75 \times 8) = 6.91$ to *The King's Speech* and a rating of $(0.9 \times 5) + (0.1 \times 0.5) + (0.75 \times 9) = 11.3$ to *Pulp Fiction*. However, these ratings don't necessarily match on the scale of the ratings that users are giving to movies. If this was for Netflix, the predicted ratings should be between one and five, since a user gives a one to five star rating for a movie. To fix this, we estimate the item and user vectors using ratings that we already know.

Suppose we have a user who gave a rating of 5 to *The King's Speech*, 4 to *Pulp Fiction*, and 3 to *The Godfather*. We need to find the user's vector (u_1, u_2, u_3), the movie vector for *The King's Speech*, (a_1, a_2, a_3), the movie vector for *Pulp Fiction*, (b_1, b_2, b_3), and the movie vector for *The Godfather*, (c_1, c_2, c_3). From the ratings, we know that

$$(u_1 \times a_1) + (u_2 \times a_2) + (u_3 \times a_3) = 5$$

$$(u_1 \times b_1) + (u_2 \times b_2) + (u_3 \times b_3) = 4$$
$$(u_1 \times c_1) + (u_2 \times c_2) + (u_3 \times c_3) = 3.$$

Using all known ratings by users for items, we would like to estimate the item and user vectors. This can be done using one of several different matrix computation methods (for more details, see the references in Section 13.7). This technique gives us vectors for all items that have ever been rated and for all users that have ever rated an item. This allows us to predict ratings for users and items that we did not originally have a rating for.

Several extensions of the matrix factorization technique have been implemented. One idea used by the winners of the Netflix prize is that users often select which items to rate, and so missing ratings can also provide information. It could be that a user tends to rate movies that they love or hate, and so we can assume that if a user watched a movie and didn't rate it, they probably neither loved nor hated it.

13.4 Amazon

Named after the largest river in the world, Amazon.com has become the world's largest online retailer. The company was founded in 1994 by Jeff Bezos and is headquartered in Seattle, Washington. It started as an online bookstore, but now sells almost any item imaginable, including electronics, furniture, food, and toys. It has an established presence in over ten countries in addition to the United States.

Amazon.com, Inc. issued its initial public offering of stock in May of 1997. The initial business plan was to not expect a profit for four to five years, which provoked stockholder complaints. But when the dot-com bubble burst, Amazon did not go out of business like many of its competitors, and finally turned its first profit in the fourth quarter of 2001, seven years after the company was launched. Table 13.5 gives the revenue, net income, and market capitalization of Amazon for the years 2001-2014.

Amazon's Recommendation System

In 1998, Amazon developed a recommendation system called "item-to-item collaborative filtering," which is still in use today. The basic idea of item-to-item collaborative filtering is to use the methods of collaborative filtering on the items themselves. Unlike content-based systems where similarity is only defined based on attributes of the items, this method also considers items to be similar if customers tend to purchase them together. Thus, the attributes of the items themselves and the behavior of the entire customer population are taken into account. One of the benefits of this approach is that there is no need for the customers to rate items to receive good recommendations.

This method uses a mapping between items and similar items. Occasionally, the previous purchases of customers are analyzed and items are deemed similar if enough customers purchased them together. For example, if many customers purchased an Origami book together with extra Origami paper, these items would be considered very similar. The higher the number of customers who purchased the items together, the more similar the items are considered to be. Note that this method does not require all pairs of items to be considered; only those that customers actually purchased together. So if an Origami book and Calcium supplements are never purchased together, the algorithm doesn't even need to compute the similarity between these two items. Items can also be considered similar based on their attributes, as in content-based systems, to enhance the accuracy of this method. So the Origami book and the extra Origami paper might be considered similar, even without a customer having purchased them together.

To generate a set of recommendations for the user, the system retrieves the items that were considered similar to the items that we know the user likes (either by the user giving a high rating to the item, or by the user putting the item in their online shopping cart). These similar items are combined into a list that is shown to the user. Based on how similar the items are, they might be shown in a particular order, such as the most similar to the least similar, or the similar items to the customer's most recent purchases might be shown before the similar items to older purchases. These strategies are meant to increase the likelihood that the customer will actually purchase items in the

Table 13.5: Financial Growth of Amazon.com. Note that the net income of Amazon in 2012 and 2014 was negative, partly due to an increasing trend of spending by Amazon on digital content.

Year	Revenue (in mil. $)	Net Income (in mil. $)	Market Cap (in mil. $)
2001	3,120	-567	4,550
2002	3,930	-149	6,390
2003	5,260	35	15,220
2004	6,920	588	17,850
2005	8,490	359	16,600
2006	10,710	190	14,870
2007	14,840	476	27,970
2008	19,170	645	29,910
2009	24,510	902	27,360
2010	34,200	1,150	58,760
2011	48,080	631	82,580
2012	61,090	-39	83,880
2013	74,450	274	123,190
2014	88,990	-241	179,720

recommendation list.

One of the advantages of item-to-item collaborative filtering is that by analyzing the recent purchases of customers, it can detect buying trends of customers. The method also is able to compute recommendations on the spot since the similarity mapping is previously calculated.

In 2010, the online video sharing website YouTube revealed that it is using a new system that is a variation on Amazon's item-to-item collaborative filtering technique. Even though Amazon's recommendation method was developed over ten years ago, Google still considered it to be so effective that they wanted to implement a version of it themselves.

13.5 eHarmony

Founded in 2000, eHarmony is an online dating site that matches single men and women for long-term relationships. The company is privately-held, and has over 20 million registered online users. In 2009, it exceeded one billion dollars in cumulative revenue over its ten years of existence.

It was started by Dr. Neil Clark Warren, a clinical psychologist, who used his 35 years of experience counseling married couples to define the traits that often lead to relationship success. According to Dr. Warren, "opposites attract, then they attack." Their strategy seems to be working, since an average of 542 eHarmony members marry every day in the United States as a result of being matched on the site, and nearly 4% of marriages in the United States in 2012 were a result of eHarmony.

Prospective eHarmony members complete a questionnaire with over 400 different questions about their characteristics, beliefs, values, emotional health, and skills. They are then matched by eHarmony using algorithms that the company has developed and refined. While the details of these algorithms are not publicly known, the idea is very similar to recommendation systems used by Netflix and Amazon. However, because of the two-way nature of dating recommendation systems, the process can be much trickier. For both Netflix and Amazon, the customer just has to like the movie or item. But in online dating, not only does the customer have to like the match, but the match has to like the customer back. It is not enough to make recommendations to a user of other people that they might be interested in. The system has to recommend people that the user is *compatible* with; people who the user is interested in and who are also interested in the user. Using information on known successful matches, the previous history of users can be used to find other matches that are predicted to be successful.

Unlike many other online dating sites, the users at eHarmony are not allowed to browse photos and profiles of all members, only the ones that the algorithms match them with. This system serves two purposes: it tries to increase the number of long-term relationships resulting from the site, and it reduces the number of profiles that users have to search through. With over 20 million members on eHarmony, it can be very useful for the site to provide recommendations. Other online dating sites are also increasingly using recommendation systems. Since there are many different online dating sites to choose from, companies are looking for ways to make their recommendations more complex and personalized. This leads to more and more systems like those resulting from the Netflix prize.

Chief Technology Officer of eHarmony Joseph Essas, previously at Yahoo!, developed matching algorithms that are based on Yahoo's system for targeting advertising based on specific users' preferences and behaviors. These algorithms have been shown to be very useful in online dating, because what people say they want and what they actually want are often two different things. These new algorithms not only take questionnaire responses into account, but also how people behave and communicate. Software innovation has become a key priority for eHarmony, and they have already seen a 10% increase in users communicating with their matches due to more sophisticated recommendation systems.

13.6 The Edge of Recommendation Systems

This chapter has shown that many companies are willing to invest large amounts of resources into creating strong recommendation systems, since helpful recommendations lead to satisfied customers and increased sales. While recommendation systems are very appealing, they also need to be efficient and appropriate. We have seen through the examples in this chapter that there are many factors that contribute to a successful recommendation system:

1. Depending on the specific business, the appropriate type of recommendation system needs to be used. The recommendation system that was constructed as a result of the Netflix prize was designed to predict movie ratings and help customers find interesting movies. Similarly, Amazon's recommendation system is designed to perform well on many different types of items, while eHarmony's recommendation algorithm is built to recommend compatible matches.

2. A successful recommendation system often needs large amounts of data. Without a large database of customers and previous purchases or ratings, it is challenging to find appropriate recommendations.

3. The recommendation system should provide fast and accurate results. Customers need the recommendations to appear as soon as the webpage loads. Additionally, customers will become impatient if the recommendation system is not providing recommendations that they are interested in.

Recommendation systems can be used in a variety of fields, including commerce, science, and politics. There are companies that use recommendation algorithms to suggest employers to job seekers and investors to companies seeking capital. Recommendation systems efficiently use massive datasets on customers and purchases to improve the experience of the customer and increase the sales of the company.

13.7 Notes and Sources

The financial information in this chapter is provided by Wikiinvest [151] and Ycharts [155].

13.3. The details about Netflix are from the press and investor pages of netflix.com [101, 100], and the information on the Netflix Prize is from a paper written by the winners of the prize [13].

13.4. The information about Amazon is from amazon.com and from a paper about their recommendation system [85].

13.5. The information about eHarmony is from eHarmony.com and from several news articles [127, 63].