Real World Problem

How do online platforms like Netflix, Spotify, or YouTube recommend you relevant content?

How It Works

Online commercial and entertainment platforms such as Spotify, Netflix, and YouTube serve recommendations to users using a mathematical framework known as a *recommendation engine*. The underlying idea behind a recommendation engine is to collect data on other users who are "similar" to you, and then make recommendations based on content that they have liked.

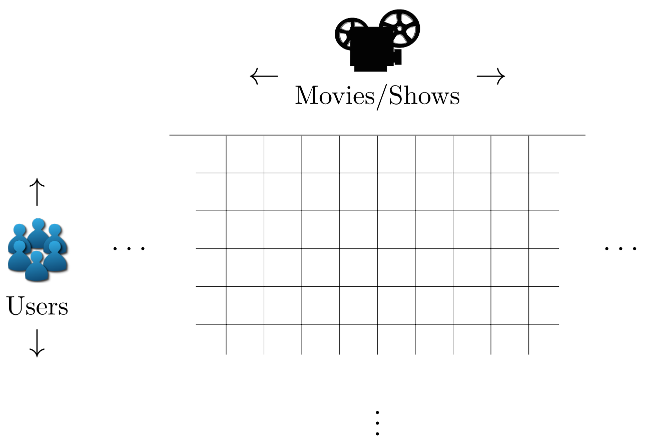
The tricky part is coming up with a way to measure "similarity" in a fast and efficient way. Let's see how a company such as Netflix might design their recommendation system.

Encoding User Preferences

Netflix, at its core, is about **users** and **movies**. For every user that uses Netflix, Netflix keeps track of ratings that they assign to movies. How can we use this to check if two users have similar movie tastes? Netflix does this by representing its user data as a *data matrix*.

Envisioning The Netflix Data Matrix

Imagine a grid. Each column of this grid represents movies or shows in the Netflix database. Each row represents a user in the database. If a user rates a movie, then in the row corresponding to that user in the grid, we fill in the column corresponding to the movie they rated with the rating they gave the movie. This grid forms the Netflix data matrix.



Now, Netflix has about 8000 films and TV shows, and well over 99 million users - can you imagine how big this data matrix might be?

To better understand how this matrix would look, let's look at a smaller example. Suppose that Netflix only had 8 users and 4 movies in its entire system, and that their ratings were either thumbs up (1) or thumbs down (0). The data matrix might look something like this:

User	Movie 1	Movie 2	Movie 3	Movie 4
1	1	0	-	1
2	-	-	1	0
3	1	0	1	-
4	1	0	1	0
5	1	0	1	1
6	0	0	-	1
7	1	0	0	1
8	1	0	1	1

Note that having '-' in a cell means something entirely different than '0'. '-' means that no rating was given (a lack of information), while '0' means the user rated the movie, but did not like it.

How could we figure out whether or not we should recommend Movie 3 to User 1, if this was the only information we had?

The question to ask is: *what makes sense*? It seems pretty reasonable to assume that users similar to User 1 have similar opinions about Movie 3 - if I have the same opinion for most of the movies User 1 has seen, it's not a stretch to believe that User 1 will have the same opinion as me about the movies I have seen. For example, if I like Movie 3, then User 1 will probably like it as well.

However, given that we only know what content a user has liked or disliked, the only choice we have for comparison is to see how many **common** ratings were given. Users with many ratings in common are likely more similar than those who have very little ratings in common.

Using this idea, we develop the following calculation to determine how mathematically similar two users are:

$$\text{Similarity Score} = \frac{\text{Number of Ratings in Common}}{\text{Number of Movies Rated Combined}}$$

This score is also known as *Jaccard Similarity*, named after French biologist Paul Jaccard.

Now that we have developed a measure of similarity, how do we use this to determine a recommendation for Movie 3 for User 1? Remember the original plan: find users similar to User 1, then consider the ratings of these specific users to determine the recommendation.

Example

Use Jaccard Similarity to find User 1's similarity scores with the other users.

User	Movie 1	Movie 2	Movie 3	Movie 4
1	1	0	-	1
2	-	-	1	0
3	1	0	1	-
4	1	0	1	0
5	1	0	1	1
6	0	0	-	1
7	1	0	0	1
8	1	0	1	1

$$\text{Similarity Score} = \frac{\text{Number of ratings in common}}{\text{Number of Movies Rated Combined}}$$

Solution

User 2: They have no ratings in common, so the numerator will be 0. Hence they have 0 similarity.

User 3:

$$\text{Movie Ratings in Common} = \{(\text{Movie } 1, 1), (\text{Movie } 2, 0)\}$$

and

$$\text{All Ratings} = \{(\text{Movie } 1, 1), (\text{Movie } 2, 0), (\text{Movie } 3, 1), (\text{Movie } 4, 1)\}$$

Hence the similarity score is $2/4 = 0.5$.

User 4:

$$\text{Movie Ratings in Common} = \{(\text{Movie } 1, 1), (\text{Movie } 2, 0)\}$$

and

$$\text{All Ratings} = \{(\text{Movie } 1, 1), (\text{Movie } 2, 0), (\text{Movie } 3, 1), (\text{Movie } 4, 1), (\text{Movie } 4, 0)\}$$

Hence the similarity score is $2/5 = 0.4$.

Note that in the collection of all ratings, Movie 4 is included twice. This is because each user gave a different rating. While the numerator remains the same for both User 3 and User 4, User 4's similarity score has a larger denominator, and so the score is lower. This makes intuitive sense: User 3 and User 4 both *agree* on the same number of things as User 1, but User 4 also *disagrees* on something with User 1 - hence, they are less similar than User 3.

User 5:

$$\text{Movie Ratings in Common} = \{(\text{Movie } 1, 1), (\text{Movie } 2, 0), (\text{Movie } 3, 0)\}$$

and

$$\text{All Ratings} = \{(\text{Movie } 1, 1), (\text{Movie } 2, 0), (\text{Movie } 3, 1), (\text{Movie } 4, 1)\}$$

Hence the similarity score is $3/4 = 0.75$.

User 6: Following the same reasoning, the similarity score is 0.5.

User 7: Following the same reasoning, the similarity score is 0.75.

User 8: Following the same reasoning, the similarity score is 0.75.

Thus users 5, 7, and 8 are the most similar to User 1, using the Jaccard score.

Now, what are their preferences for Movie 3? Users 5 and 8 like it, while User 7 does not. Since the majority (66%) liked Movie 3, the recommendation engine would recommend Movie 3 to User 1.

Some Things to Think About

1. Mike thinks he can come up with a better approach for recommendations - recommend movies that are similar to movies that a user has already searched.
   (a) What are some problems with this method?
   (b) If a movie is not similar to movies I've searched, does that mean I won't like it?
2. What if User 8 did not give a rating for Movie 3? Then it would be a 50-50 split – should we recommend Movie 3 to User 1 in this case?
3. What if none of the most similar users watched Movie 3?
4. How many blank ('-') entries do you think the real Netflix data matrix would contain? Think about how many movies a user might actually *rate* (not just watch)
   (a) What do you think the Jaccard similarity score would be, on average, between users?
   (b) How confident do you think you can be in your recommendations if the similarity scores are all really low?
5. How might you adapt this recommendation engine to produce recommendations for Spotify (a music, not movie/TV, streaming service)?

Conclusion

The recommendation engine we just described is commonly known as *collaborative filtering* - to make a recommendation, we look for other users who watch similar things (collaborators), and ignore those who don't (filtering). The key point is that we are looking at the behaviour of *other* users, to judge how a *specific* user will behave.

This approach only works, however, if there are other users who have something in common with us! If there aren't many (or even any), then many of the scores we calculate will be very close to 0 - **can you explain why**? What problems, if any, could this cause?

In the Real World

Netflix has spent much time and effort on finding ways to overcome these challenges and provide better recommendations - even offering a substantial cash prize, called the *Netflix Challenge*, to anyone who could come up with a better recommendation engine than their existing one (you can read more about this challenge on Wikipedia, at `https://en.wikipedia.org/wiki/Netflix_Prize`)

The Netflix Challenge is just one example illustrating the excitement and the challenge of doing data science - using and dealing with real world data, and how in demand these skills really are. Here are some other ways that data science is being used right now in the real world:

- predicting credit card fraud,
- recommending great movies,
- optimizing stock market trading,
- finding flaws in utility systems,
- classifying cancerous tumours,
- understanding patterns in organized crime...

... and the list goes on. Right now, there is an ever increasing demand for talented data scientists in society - we need smart, motivated, and interested students like you to innovate and make a difference in the world.

And it all starts with mathematics.

Exercise

Suppose that there is a startup music company called Sparsify. Right now, their library contains only 8 loyal users and 16 songs, spanning 9 artists and 5 genres in total.

- User 1: Likes songs 1, 2, 7, 8, 9; dislikes songs 3, 4, 5, 11, 15
- User 2: Likes songs 2, 4, 7, 8, 13; dislikes song 1
- User 3: Likes songs 6, 10; dislikes songs 8, 9, 11, 13, 14
- User 4: Likes songs 4, 5, 6, 10, 11, 12, 14, 16
- User 5: Likes songs 11, 12, 13, 14, 15; dislikes songs 1, 2, 3, 4, 5
- User 6: Likes all songs except for 7 and 13
- User 7: Doesn't rate any songs
- User 8: Like songs 2, 3, 4, 7, 8, 9, 11, 13, 15, 16; dislikes 1, 5, 6, 10, 12, 14

- Song 1: Genre 2, features artists 2
- Song 2: Genre 2, features artists 3
- Song 3: Genre 3, features artists 4 and 7
- Song 4: Genre 1, features artists 1
- Song 5: Genre 4, features artists 8
- Song 6: Genre 4, features artists 8
- Song 7: Genre 2, features artists 5 and 6
- Song 8: Genre 3, features artists 5
- Song 9: Genre 3, features artists 6
- Song 10: Genre 1, features artists 1 and 6
- Song 11: Genre 1, features artists 2 and 9
- Song 12: Genre 1, features artists 9
- Song 13: Genre 5, features artists 7
- Song 14: Genre 2, features artists 4
- Song 15: Genre 5, features artists 3
- Song 16: Genre 3, features artists 1, and 4

1. Use a collaborative filtering approach to determine if song 6 should be recommended to User 1.

2. Why can't we use collaborative filtering to recommend songs to User 7?

   (a) If you absolutely had to recommend something to User 7, what would you choose, and why?

   (b) Can you think of other types of data we might be able to collect on User 7? How could we use that to give them recommendations?