Math 3130 Project Proposal

Quang Tran, Tiffany Tran, Jeffrey Holloway

27 March 2021

As technology increases, so too does our desire to integrate it into our daily lives. As this occurs, old technologies that can’t adapt die off and new technologies fill in their place. One such representation of this is the economic extinction of businesses such as Block-Buster. This business was an apex in movie entertainment for a time but as technology increased, they could not adapt. Shortly after its doors closed, online streaming services began taking over. Now, there are a plethora of online streaming services and they are all fighting for your attention and subscription. But how does a streaming service know what movies catch your attention? These streaming services, such as Netflix, make use of what is called a recommendation algorithm. A recommendation algorithm is an approach made by a business or service that can offer personalized choices to their individual customers. These services collect usage data in a variety of ways. They then compile these data in a specific manner such that they can use mathematics to determine the highest quality of recommendations to the individual user. An overwhelming majority of online sites as well as applications make the use of recommendation algorithms.

In collecting these data and processing them using advanced mathematics, these services are not just guesstimating what they believe to be the most likely choice. They are learning about what categories of movie is the most engaging to the user. They are studying the user’s level of interaction with the service, and learning what the user is most interested in as they use the service provided. These algorithms are exceedingly important to the business as it allows them a competitive edge in the industry. But how could a recommendation algorithm be used to give a competitive edge? As the algorithm learns what the user likes, it tailors the experience of the user in a very personalized manner. If this tailored experience is what the user is looking for, they will spend more time using that service. This implies that the service will make much more money off the continued use of the individual. As the user spends more time on the service, the algorithm will be more fine-tuned to the user’s preferences. All of this can provide an improved experience for the user while giving the service an edge with their competition.

The general idea of creating an algorithm is so that the machine can predict the user’s likes and dislikes, so that essentially other movies that are similar to the ones that the user likes are recommended, and vice versa―movies that the user dislikes are hidden. This is done via *number rating system,* where movies with the highest user preferences are given higher points (on a scale of one to five). This data is collected through the ratings that the user gives a specific movie, where we can use this information to predict the user’s ratings on other movies. This data can also be collected for the number of times someone has watched a certain genre, or the search habits of a user. First, each movie is assigned a set of numbers based on the genre it applies to the most (the assignment is done through the process of matrix factorization, which will be explained later). For example, a movie like “Shrek” could be assigned a 3 in family movies and a 2 in comedy. Given an example Netflix user who does not like family movies but enjoys comedy, this person would be automatically assigned the rating of 2 to Shrek (0\*3+ 1\*2). Conversely, if someone liked family movies but hated comedy, that person would be assigned the rating 3 to Shrek (1\*3 + 0\*2). Essentially, this method can be used for thousands of other movies on Netflix, creating a big collection of data that will be essential in filtering the kinds of movies to recommend.

Moreover, this data can be applied to other users who have reported similar likes and dislikes as those users. This method is known as *collaborative filtering*, which is essentially finding users who have similar watch patterns and preferences, and giving them similar movie recommendations. Collaborative filtering is effective because it is not always certain that users will rate every movie they watch. So, by using linear algebra methods and algorithms, we can deduce and predict what kinds of movies to recommend. Given a matrix of user ratings of certain movies, with the rows being the users and the columns being the movies, there are blanks in some cells of the matrix. These blanks are due to unknown user ratings for certain movies. By developing a system that would “fill in the blanks” using collaborative filtering, matrix factorization, the dot product, and other linear algebra fundamentals, the system can then recommend a movie like “Despicable Me” to fans of “Shrek”, which both have similar genre ratings.

Linear Algebra is very practical and applicable, especially in today's world with computers all around us. In particular, Netflix has a recommendation system that sorts and finds shows and movies that the user will like, all on its own. In general, this wouldn’t be feasible, as humans like different things, and these likes tend to be very subjective. One person could be a polar opposite of another, or more likely, one person could have the same likes as the other. By using matrices, we could record these measurements of how much a user likes the show with values. In our example, we have 4 users, A B C and D with 5 movies, M1, M2, M3, M4, M5. Through Content Filtering, we can construct a 4x5 matrix of 4 Netflix Users (watchers) by 5 Netflix Movies. The resulting values in these matrix elements correspond to how much they enjoy the movie, from a range of 0(dislike) to 4 (love).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Users x Movies | M1 | M2 | M3 | M4 | M5 |
| User 1 (A) | 3 | 1 | 1 | 3 | 1 |
| User 2 (B) | 1 | 2 | 4 | 1 | 3 |
| User 3 (C) | 3 | 1 | 1 | 3 | 1 |
| User 4 (D) | 4 | 3 | 5 | 4 | 4 |

The main benefit to factoring the matrix into smaller submatrices is that the storage is much more efficient. In this example, the original matrix has 20 elements, but the total number of elements in these submatrices is 18. In larger matrices, this is much more apparent. By performing matrix factorization, the matrix above can be broken down into two smaller matrices, a 4x2 matrix about users and their preferences to certain movies. In this matrix however, the values are only 0’s and 1’s. These are boolean(true or false) values where 1 represents that the user likes that movie type, and 0 represents that they dislike it.

|  |  |  |
| --- | --- | --- |
| 4x2 User Matrix | Movie Feature 1 (Comedy) | Movie Feature 2 (Action) |
| User 1 | 1 | 0 |
| User 2 | 0 | 1 |
| User 3 | 1 | 0 |
| User 4 | 1 | 1 |

In the matrix above, user 1 likes Comedy movies, and dislikes Action movies.

The other factorized matrix is a 2x5 is a matrix correlating with the movies and various features(genres) of the movie. The values range from 1 - 4, from least correlated to most correlated. For example, a movie with a 4 in comedy is more comedy oriented, and a movie with 1 in action is the least action oriented.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Movie 1 | Movie 2 | Movie 3 | Movie 4 | Movie 5 |
| Feature 1 (Comedy) | 3 | 1 | 1 | 3 | 1 |
| Feature 2 (Action) | 1 | 2 | 4 | 1 | 3 |

In the matrix above, the values indicate how relevant they are to the feature(genre).

Since these two matrices are factorized from the parent matrix, performing the dot product on these two matrices with these two factorized matrices will recreate the original matrix.

Each element of the matrix can be recreated with the Dot Product. For example, the row 2, col 3 of the matrix above can be found by doing the dot product with the 2nd row of the user matrix, and the 3rd column of the movie matrix.

Dot product for user row 2 and movie col 3:

X · Y = (0 \* 1) + (1 \* 4) = 4

Citations:

Prasad, A. (2020, September 15). *The Mathematics of Recommendation Systems - Level Up Coding*. Medium. <https://levelup.gitconnected.com/the-mathematics-of-recommendation-systems-e8922a50bdea>

*How Netflix Recommendations System Works*. (N.D.). Netflix. <https://help.netflix.com/en/node/100639>

<https://levelup.gitconnected.com/the-mathematics-of-recommendation-systems-e8922a50bdea>

<https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48>

<https://uh.edu/engines/epi2514.htm>

<https://math.berkeley.edu/~sander/fall2016decal/reading10.pdf>

<https://dl.acm.org/doi/10.1145/2843948>

<https://dl.acm.org/doi/10.1109/WI-IAT.2013.140>

<https://towardsdatascience.com/tensorflow-for-recommendation-model-part-1-19f6b6dc207d>

<http://www.mathsjournal.org/Volume3/Issue2/IJMCT-V3I2P1.pdf>

<https://www.youtube.com/watch?v=ZspR5PZemcs>

<https://www.youtube.com/watch?v=n3RKsY2H-NE>

<https://hellanicus.lib.aegean.gr/handle/11610/18038>

[https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf](https://datajobs.com/data-science-repo/Recommender-Systems-%5bNetflix%5d.pdf)

<https://blogs.commons.georgetown.edu/cctp-607-spring2019/2019/03/13/de-blackbox-the-algorithms-of-netflix-recommendation/>