Cinematcha: A (Better) Movie Recommendation System

Ignacio Amezcua: *ignacia@uci.edu*
Aditya Koppula: *akoppula@uci.edu*
Zeeshan Hanif: *zhanif@uci.edu*

University of California, Irvine

ABSTRACT

With the rise of media streaming services in the 21st century, movies have become exponentially more popular as their accessibility increases. Despite the explosive growth of key industry players such as Netflix and Hulu, movie recommendations have long been a hit-or-miss feature in services that offer them. Simplistic matching algorithms, typically based only on past watched movies, deliver a mixed bag of results. Users are left to sift through vaguely relevant suggestions manually; Thorin Klosowski, a reporter on media and entertainment, simply puts it as "not only a waste of time, [but also a process that] makes the already time-consuming act of watching a movie more drawn out."[1] The vast number of other factors that could be used to form an effective personalized recommendation are typically ignored. To remedy this, we propose a new system for generating movie recommendations for users. This system, dubbed **Cinematcha**, will aggregate data from several disparate data streams which could influence a movie-watcher's preference to create the optimal set of recommendations for any particular user at a given time.

I. INTRODUCTION & PROBLEM DEFINITION

The purpose of Cinematcha is to be a movie recommendation solution that not only is useful for users to find relevant movies for their desires at a given time but also to receive relevant recommendations based on genre preferences and viewing history. Our goal in this undertaking is to give our users a way to explore their interests in movies, in a tailored fashion that fits each user individually. With access to a plethora of movie data and armed with knowledge of movie recommendations from our own experiences as well as the experiences of fellow movie watchers, Cinematcha was born.

Cinematcha is a movie search and recommendation system with the explicit focus of catering to each user's needs. Users should expect to have a customized movie discovery experience based on inputs such as personal movie ratings and the current weather. Cinematcha will do its best to rank movies in accordance to the inputs given and provide the user with relevant information on each movie. It is almost like having a personal chef who knows your allergies, your favorite foods, your cravings and puts all of these things together to cook up the perfect meal for you.

Cinematcha does not stop there. Aside from being a movie recommendation program, it doubles as a database for movies. Users are able to search up movies on the system to find out more information about each movie. This feature is incredibly useful when users are seeking detailed information about a certain movie.

It can be questionable to choose a movie to watch solely on the fact that it is highly rated. There are certainly a wide selection of good movies, measured by a composite viewer score of 80% or higher but that does not really tell a user much. Cinematcha looks through all possible results, and looks through your interests and some other factors to narrow all the movies down into something that you would want to watch. A critic score of 90 is useless if the movie's content and topic do not interest the user. Cinematcha aims to sift through the vast collection of 'good movies' and recommend those that have a personal appeal to you.

Users will be presented with information about movies, trailers, general information about that movie, movies to watch by category and more. We plan on giving the user as much information as we can so that they can make the best educated decision on whether what we offer is relevant in the end. After all, the users will be watching and experiencing the movie, not us.

Overall, users can expect Cinematcha to aid in the process of movie selection. Looking for a good movie to watch should not be something that takes longer than watching the movie itself. We hope that Cinematcha will utterly outperform and make obsolete the existing ineffective search systems that Netflix and Hulu and the like implement.

II. CURRENT STATE OF ART

When we began to develop our program, we had in mind some software that is already accessible to everyone. We have found that MALgraph and IMDb were the direction we wanted to follow while being able to recommend movies to a user like the pre-existing recommendation systems in Hulu and Netflix.

i. Netflix

One of the recommendation systems we can directly compare ourselves to is the one that **Netflix** already uses. Netflix is a popular movie and TV show streaming service that exists many households across the US. Users who have access to Netflix know that Netflix tries to recommend you movies and TV Shows but most of the time they're impacted by things users don't really care about and that seem irrelevant.

Taking a look at how Netflix handles their recommendations, we can see that they take things into consideration such as "the genres of TV shows and movies available, ... streaming history and previously made ratings ... and a combined rating of all Netflix members ..."[8] One of the flaws that we see with this type of recommendation is that users of Netflix are treated as people who like watching the same things. It isn't always the case that what one user likes might be what some other user likes. This is because each form of media isn't one dimensional. There could be many reasons why a user likes a particular

show or movie. This could be as abstract as watching a film "because Chris Pratt" is in it or liking a film "because I love bobcats and they appeared in this documentary". These reasons are of course made up but it isn't hard to think of someone who is a diehard fan of an actor that thinks of an actor as a god therefore watching certain movies and rating them higher. We must consider the fact that everyone is unique and Cinematcha takes this into account.

ii. Hulu

Hulu[9], another popular streaming service, also takes a similar approach to Netflix but puts a spin on it. Hulu decided that their streaming service would offer a dynamic list that changes every time a user logs on. A user can populate their Hulu account with a watch list, favorites, and queue of what to watch and Hulu tries its best to recommend you things that it knows to be similar to what you are watching. One of the unique things that Hulu does is recommend you things that aren't on its service yet. This includes recommending you new shows that are going to come out, recommending you future episodes of a series you like to watch so that you are aware of what's coming, and all of this along with the static content recommendations takes a step in the right direction. Having the ability to utilize what they have and will have makes Hulu recommendations seem more vivid.

However, it is not specified exactly how movies and TV shows are tied together. Our best assumption is that they take the same approach as Netflix with treating users as mostly the same.
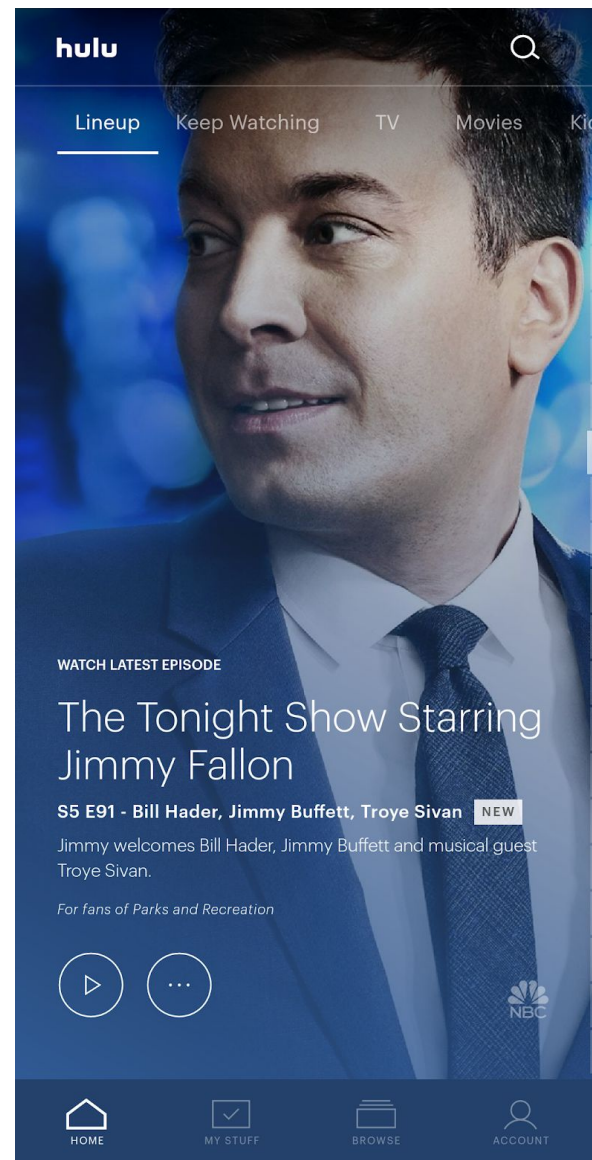

Figure 2.1 (Hulu recommending what to watch based on what they think fans of a show like.)

Here, in Figure 2.1, we can see some of the process that Hulu provides to its
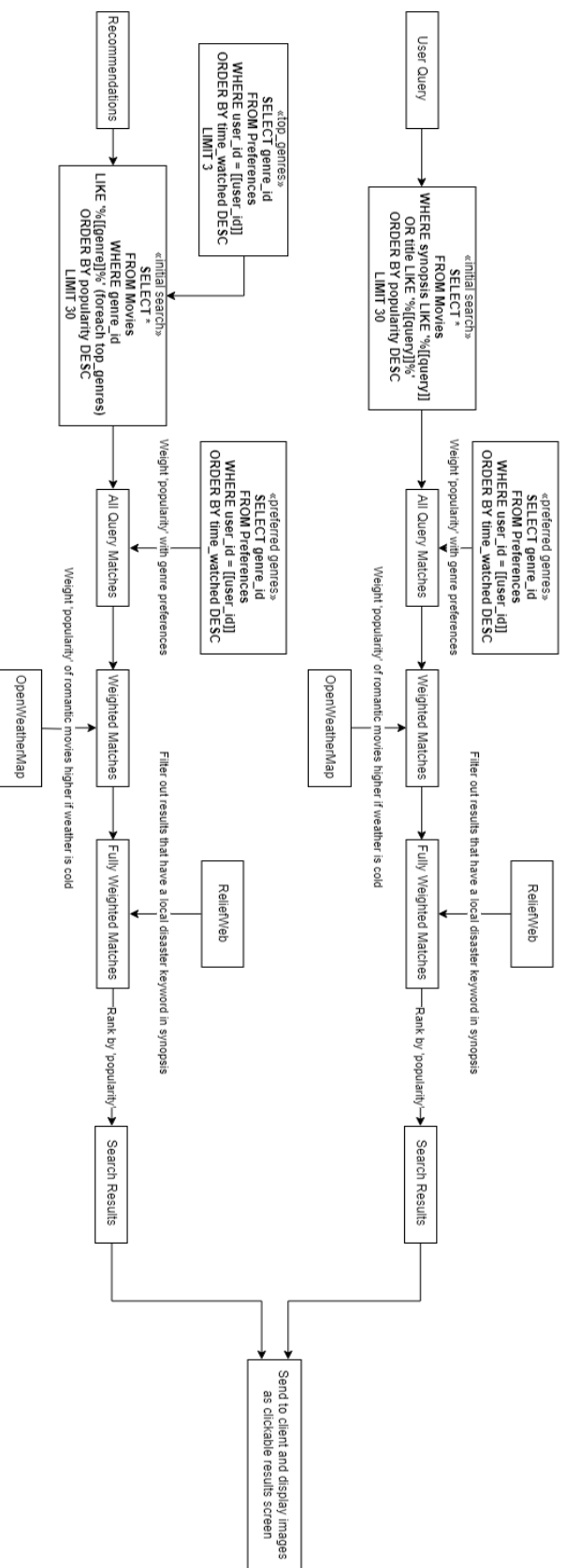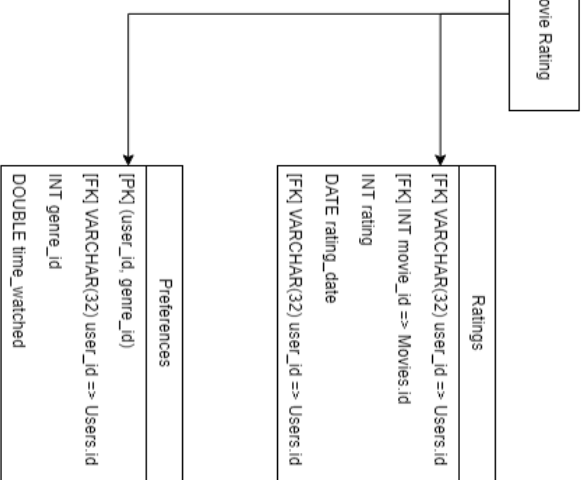
users. Recently, after finishing "Parks and Recreation", the Hulu app started recommending "The Tonight Show Starring Jimmy Fallon." Hulu makes this assumption without knowing what either show is about. One is a comedic sitcom and the other a late night comedic talk show. This is a hard assumption that Hulu makes, and in this case we can personally say that watching a sitcom leads to no interest in watching a late night talk show. We made it our goal to have our application give users better results than this. We don't want to give users recommendations based on the actual content they watched but rather on a more personal level which is also something Hulu didn't take into account. The user of this Hulu profile hasn't even given a rating on the show Hulu is giving a recommendation off of. It could've been the case that the user did not like "Parks and Recreation" and Hulu tried to give recommendations anyway. These types of recommendations make the user feel categorized which isn't something a good recommendation system should do.

III. ARCHITECTURE

When designing our recommendation system, we wanted the users to do less and receive more. By connecting their MovieDb accounts, users of Cinematcha are able to import pre-existing data and take comfort in a recommendation system that works in the blink of an eye.

We didn't stop there, we wanted the user's recommendations to feel a little more special by adding the input of the weather. In a study on the effect of the weather versus movie category likeability, researchers discovered that the weather had a direct correlation with how much a user enjoyed films[2]. In particular, when the weather was raining outside, we decided to use that information and recommend our users romantic movies to complement the mood.

We have detailed below the data that we receive from MovieDB and the other sources of input we use - namely, user input, OpenWeatherMap, and ReliefWeb. OpenWeatherMap and user preferences are used to weight a general popularity score from MovieDB, while ReliefWeb is used to filter out results the users may not want to see (since related disasters recently occurred near them, possibly causing mental trauma if they were to witness a similar scenario in a movie. The weather is primarily used to boost the likelihood of showing romance movies in cold weather[2]. Of course, if the user does not like romance movies, their personal preferences will outweigh the weather bias to result in a suggestion they would still consider. The process flows for search and recommendation are very similar, with the main difference being that recommendations are primarily genre based. We may add a past movie keyword component to improve recommendation quality.

MovieDB Movie Listings → (Retrieved 16,335 movies)

Movies
- [PK] INT id
- VARCHAR(256) title
- VARCHAR(8) original_language
- VARCHAR(256) original_title
- VARCHAR(128) genre_ids
- INT vote_count
- DOUBLE vote_average
- DOUBLE popularity
- VARCHAR(5096) synopsis
- DATE release_date
- VARCHAR(128) poster_path
- VARCHAR(128) backdrop_path

User Profile Information

Users
- [PK] VARCHAR(32) user_id
- VARCHAR(256) password
- INT zip_code
- VARCHAR(256) account_id
- VARCHAR(256) access_token

Indices
- Clustered Index on Movies.genre_ids (initial filter)
- Index on Movies.title (for actual search term lookup)
- Index on Movies.synopsis (for actual term lookup)
- Index on Preferences.user_id (for quick lookup in preferences)

User Movie Rating

Ratings
- [FK] VARCHAR(32) user_id => Users.id
- [FK] INT movie_id => Movies.id
- INT rating
- DATE rating_date
- [FK] VARCHAR(32) user_id => Users.id

Preferences
- [PK] (user_id, genre_id)
- [FK] VARCHAR(32) user_id => Users.id
- INT genre_id
- DOUBLE time_watched

User Query

«initial search»
```
SELECT *
FROM Movies
WHERE synopsis LIKE "%[[query]]%"
OR title LIKE "%[[query]]%"
ORDER BY popularity DESC
LIMIT 30
```
→ All Query Matches

«preferred genres»
```
SELECT genre_id
FROM Preferences
WHERE user_id = [[user_id]]
ORDER BY time_watched DESC
```
(Weight 'popularity' with genre preferences)
→ Weighted Matches

OpenWeatherMap — Weight 'popularity' of romantic movies higher if weather is cold
→ Fully Weighted Matches

ReliefWeb — Filter out results that have a local disaster keyword in synopsis
→ Rank by 'popularity' → Search Results

Recommendations

«top_genres»
```
SELECT genre_id
FROM Preferences
WHERE user_id = [[user_id]]
ORDER BY time_watched DESC
LIMIT 3
```

«initial search»
```
SELECT *
FROM Movies
WHERE genre_id LIKE "%[[genre]]%" (foreach top_genres)
ORDER BY popularity DESC
LIMIT 30
```
→ All Query Matches

«preferred genres»
```
SELECT genre_id
FROM Preferences
WHERE user_id = [[user_id]]
ORDER BY time_watched DESC
```
(Weight 'popularity' with genre preferences)
→ Weighted Matches

OpenWeatherMap — Weight 'popularity' of romantic movies higher if weather is cold
→ Fully Weighted Matches

ReliefWeb — Filter out results that have a local disaster keyword in synopsis
→ Rank by 'popularity' → Search Results

Send to client and display images as clickable results screen

IV. QUERY/INPUT ENVIRONMENT

The project was executed on a Digitalocean server running Ubuntu 16.04. Queries were executed using a LAMP Stack, combining the practicality of MySQL's ability and the flexibility of PHP. Through MySQL we were able to efficiently retrieve our data, and we primarily used PHP for query construction and ranking.

In order to provide our users with the best possible precision and results, we first execute an initial analysis on the movies rated by a user. We calculate a user's favorite genres by time spent watching and average score. We then execute our query to retrieve movies with synopsis and titles which fit our keywords. Using our preemptively fetched information on the user's viewing habits, we then weigh results featuring user's favorite genres more heavily. We then further ranked items by taking into account the weather, and removed items in accordance with our ReliefWeb proposal.

From these filtering and ranking algorithms, we hope to deliver the greatest movie recommendation experience to our users.

V. PRESENTATION/OUTPUT

We send detailed recommendations to our users in JSON format, which we use to display the posters of the movies in a grid format. Hovering over one of the posters causes the background of the main page to change to the backdrop of the movie's poster. Clicking on the movie's poster will redirect the user to the trailer of the movie.
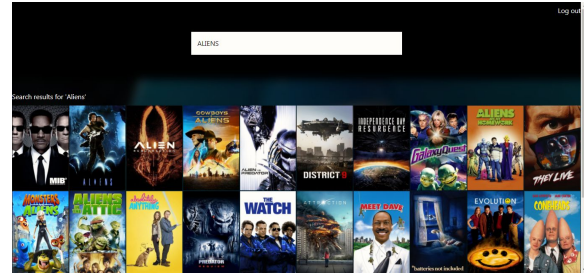

Figure 5.1 - Default view of cinematcha's search functionality, with the example search term, "Alien".

VI. DETAILS OF COMPONENTS & ALGORITHMS

Our system currently uses a standard query on the database with either the query term or the user's top 3 genre preferences, then applies the aforementioned weights and filters to each result's popularity scores. This is a preliminary weighting system, and still needs fine-tuning to adjust how much each factor (i.e. weather, user preference, search term) influences the final ranking of the results. Multi-word queries are also yet to be implemented due to a shortage of time.

An experimental version of improving the search system would be with using the Euclidean distance of how movies rank in comparison with each other with dimensions stemming from topics and information based off the movies themselves[5]. However, using the Euclidean distance to measure compute how relative two movies are to one

another is not the smart way to go. To better improve this method, we would need to compute a cosine similarity after normalizing the vectors which each movies makes. From here, we could use the smaller Euclidean distances to determine how relevant a movie is to a seperate move. By using these cosine similarities, we can see how users choose what to watch and get greater insight on what makes a horror movie as interesting as a family-fun anime movie for a user.

One of the ways we could improve our system in the future would be to implement a support vector machine algorithm to aid in choosing what kind of movies matchup on a closer level[4]. By being able to use this technology, we will build a set of movies that come close to a previously watched movie, using technology that we already use in Cinematcha and get movies related to those movies, whether it be by topic, actor, genre, essentially whatever our algorithm learns that our user likes, and populate a bigger list where we can narrow down from. Being able to have a variety of movies to choose from will help our algorithm make the best choice for our users. We plan on making that a permanent implementation once we learn how people react to the choices Cinematcha already makes.

Another implementation we will add in future use of our program's algorithm is one that adapts naturally to a users preferences. The research article "Preference-based search with adaptive recommendations" talks about how we can implement an variables into our algorithm that build from saving user preferences that the system learns about a user[7]. Given the fact that we can store a preference like "loves dinosaur movies" we can provide a plethora of movies from various genres that the user might be interested in. These will give the user movies like *Jurassic Park* and *The Land Before Time*, movies that are about dinosaurs, and vastly different including the style of the film and target audience. Users who "love dinosaur movies" will probably want to watch these movies rather than watching a thriller like *Jurassic Park* but without dinosaurs. Our algorithm would learn more about the user and provide movies with common details that the user looks for in film.

Choosing what variable to recommend off of, aside from the movie genre and statistically liked movies by fans and critics, we will be able to use ontologies created from machine learning to factor in other examples like producers, art directors, and costume designers[3]. Advanced and better computer learning techniques will allow for the system to analyze the film and add sub-categories based on every film. Having a way to sub-categorize films would help our database find films that are similar based on categories that users prefer, therefore making the process of finding new films easier for the system.

VII. USER STUDIES & EVALUATION

The purpose of our system is to recommend users media they are going to be interested in. With this in mind, we decided to compare and contrast results between Hulu's recommendation system and our recommendation system. We added a rating to our recommendation system and simply watched the movie, *Alien Psychosis*, to see what we would be recommended on the Hulu application. Here were the first 5 results we saw on both recommendation platforms dealing with the movie at hand:

Hulu

Movie Title	Type of Show	Likeliness to watch (1 - 10)
American Exorcism	Ghost/Undead horror	2
The Suffering	Undead Horror	4
Drinksgiving	Comedy	1
Black Ghosts	Mystery Horror	4
Apocalypse Cult	Mystery Horror	3

Average Likeliness to watch: **2.8**

Cinematcha

Men in Black	Adventure, Comedy	6
Men in Black 2	Adventure, Comedy	4
Spiderman 3	Action Adventure	5
Alien	Sci-Fi Horror	7
District 9	Sci-Fi Thriller	8

Average Likeliness to watch: **6**

Media given to recommend off of

Alien Psychosis	Sci-Fi Alien horror	N/A

From the results, we can see a big difference in the likeliness of watching the movies given by both recommendation systems. Without trying to be biased, Cinematcha proved to give back more relevant results catered to what we were trying to watch. We believe that this is because Hulu tried to classify us as a user who wanted to watch horror films when that is not the case. Being more interested in space and aliens gave titles like *Alien* and *District 9* a higher score. Those were the type of movies that we as a user wanted to see; movies that explore alien life.

One study that we came upon talked about how users preferences evolve with what they watch to interest them in new forms of entertainment to watch[6]. We have seen evolution happen in our day to day lives because movies can grab someone's interest in a way that exposes them into movies that are branched all over the place. Take for example watching a movie like *Iron Man*, a movie about a superhero that fights

bad guys. *Iron Man* can interest people into technology and academia films which can be nothing like superhero movies. Because of this, movie recommendation systems need to learn how to adapt and see where users could roam to for different movie options that have these newly found interests.

VIII. CONCLUSION

While Cinematcha is still very much in its infancy, we have a clear vision of the end goal of our system and the revolution it could bring to the on-demand consumer digital media industry. We will continue to revise our methods and user experience to create the best possible experience for our users within our project development timeline.

IX. REFERENCES

[1] Klosowski, Thorin. "How to Stop Flicking Through a Million Movie Options and Pick Something to Watch." *LifeHacker*, 28 Mar. 2012.

[2] H. Jiewen and S. U. N. Yacheng, "Warm It Up with Love: The Effect of Physical Coldness on Liking of Romance Movies," Journal of Consumer Research, Article vol. 39, no. 2, pp. 293-306, 2012.

[3] M. N. Moreno, S. Segrera, V. F. López, M. D. Muñoz, and Á. L. Sánchez, "Web mining based framework for solving usual problems in recommender systems. A case study for movies'

recommendation," Neurocomputing, Article vol. 176, pp. 72-80, 2016.

[4] X. Wang, F. Luo, Y. Qian, and G. Ranzi, "A Personalized Electronic Movie Recommendation System Based on Support Vector Machine and Improved Particle Swarm Optimization," PLoS ONE, Article vol. 11, no. 11, pp. 1-17, 2016.

[5] Z. Wang, X. Yu, N. Feng, and Z. Wang, "An improved collaborative movie recommendation system using computational intelligence," Journal of Visual Languages & Computing, Article vol. 25, no. 6, pp. 667-675, 2014.

[6] S. Zhang, Z. Jin, and J. Zhang, "The dynamical modeling and simulation analysis of the recommendation on the user–movie network," Physica A, Article vol. 463, pp. 310-319, 2016.

[7] Viappiani, Paolo, et al. "Preference-Based Search with Adaptive Recommendations." *AI Communications*, vol. 21, no. 2/3, June 2008, pp. 155-175. EBSCO*host*, doi:10.3233/AIC-2008-0433.

[8] Netflix. "Netflix Ratings & Recommendations." *Help Center*, help.netflix.com/en/node/9898.

[9] Hulu. "Introducing Watchlist." *Stream TV and Movies Live and Online*, www.hulu.com/press/introducing-watchlist/.