Development of a Hybrid Recommender System

David S. Restrepo, Hernán A. López
Faculty of Electronic Engineering and Telecommunications, University of Cauca
Popayán, Colombia
dsrestrepo@unicauca.edu.co
andresmazorra@unicauca.edu.co

***Abstract*— Recommender Systems (RS) are often helpful to provide content for users based on certain criteria. Thus, RS can be classified by content, context, knowledge, collaborative filtering, semantic, etc; and depending on their strategy, websites like netflix[1] can use one or many of them to develop their own systems. Also, It's possible to combine two or more RS models, to create what 's known as Hybrid RS, so this can provide results with a better performance than those that use single techniques. For a better approach, is presented a summary of the development of a Hybrid RS for movies, based on two popular techniques (Content Based and Collaborative filtering) for the design of RS, in order to provide a kind of guide to build each approach as a module and a hybrid model that integrates these two.**

***Keywords*— Recommender System, Hybrid, Collaborative Filtering, Content-based, Algorithm, Movies.**

I. OVERVIEW

The developed algorithm function is to recommend different movies depending on data obtained from the user like the last movie seen or ratings given. Since this system is a hybrid RS that uses a switching-mixed technique, it uses two RS techniques (that will be explained later in this document) to relate user preferences with similar topics and help it to recommend movies that should get a better match with each user.

Since the system needs a good variety of data to provide adequate recommendations, we decided to use the Movielens dataset [3] which is easy to get access to and contains many types of data metadata so this implementation is easily replicable. anyway, this recommendation system can be easily applied to other types of data such as games, online stores, among others, as long as they comply with the characteristics of having a summary of the item, rating, users, item id and a classification as gender.

Although the modules that build the recommendation system allow eliminating some characteristics as genres or overview, the use of all of them is recommended to have better performance and greater customization and variety resulting in a more reliable algorithm for the system.

II. DATA SELECTION

This is the first step and the most important, as has been said previously, the performance of the recommendation system is better with better quality and quantity of data, so it is very important to take into account the objectives in this case the final objective was to create a Hybrid RS, and deploy it on a web site.

So data as genres, overview, users id or ratings were required this makes some data sets like the papers data set [4] or other datasets explored as a workout data set, or health recommendations dataset, also were important for a web page the presentation, so there are some datasets like amazon e-commerce[5] or the modcloth [6] that don't display names or were very large for a online implementation due to required processing times.

The movie lens data set provides a good data quality, as well as metadata and a small version perfect for an online implementation.

III. TYPES OF RS USED

As was said in the previous section, we used two types of RS to develop the Hybrid RS to provide a better performance. Those techniques are:

A. Collaborative Filtering.

Collaborative filtering helps us to relate information from different users with similar interests to predict some items for a certain user and give to him/her the proper recommendation based on it [7]. In this case, the ratings given by each user to the items that were in the dataset were used as reference to make a profile of each user, those profiles were used to build a matrix of ratings with rows being the total movies and columns the total users, then using a pearson correlation each column (user) of this matrix were related to the ratings of each item given by the target user this returned a value between 1 and -1 for each user, where 1 is the maximum correlation (that means equal values) and -1 an opposite relationship, from this list of correlations the first 10 positive values were taken in order from highest to lowest, to ensure the credibility of the correlation just were selected users that had at least 3 movies in common with the target user; then the ratings of each movie were multiplied by the correlation and divided on the maximum rating giving us a maximum of 1 an minimus of 0, those values where used as weights for each movie in order from highest to lowest, finally from this list the first 10 values are taken.

Having implemented this the user-based collaborative filter is done, this should recommend to the user some movies that the user hasn't rated but which are popular for similar users. This is very helpful to help bring users to the long tail and exploit those niche markets.

But as we can see there is a problem if the user is new to the platform or has not ranked enough movies the results will not be reliable at all, this problem is known as the cold start problem, to solve this the other approach is the content-based recommender system.

B. Content-based.

Content-based techniques help us to relate items with

similar items. So, based on a item information about the user can be obtained as the genres that user likes the most and recommend movies of those genres, or using movies related to overview recommend the continuation of a saga, an example is if the user watches a movie the genres can be used i.e. Action, Drama, Horror to select other movies of the same genres to recommend the preference categories and based on it, give to the user some recommendations from that same category [8].

In this case were used three approaches to relate movies;

1. Items with similar genres: Here were used the word2vec model to feed this model were used the movies each user has seen and his genres, the reason was to build a content based recommender, but with some benefits of a item-based collaborative filter, so it would be easier for the system not only to find movies with similar genres, but also to know which movies were also seen by users who saw those movies allowing to find niches; Another advantage of this approach is that let do operations between genres, so items of various genres can be find and recommended easily
2. Items with similar overview: This approach is just a complement of the word2vec based models, uses the overview of each movie converted to a bag of words using TF-IDF [9], this bag of words of each movie is applied to a cosine similarity to estimate the similarity of those overviews [10] this approach is bad to find niches, but is good to continue sagas.

IV. COMBINATION METHOD

At this point there are two approaches that can be represented as classes as the collaborative filtering and content-based filter as shown in the class diagram [Fig. 1] and the components with the same names in the components diagram [Fig. 2] but is still missing one of the most important components the method that mixes those two approaches.

There are many methods to do a hybrid recommendation system [2], these methods depend on the components, and the data, in this case were used two approaches

1. A switching method: this is important for each new user due to the cold start in this case if the user has not rated 6 movies or more the collaborative filter is not used, just the Content Based.
2. A mixing method: This is used before the user rated 6 different items, then the results of the content-based filter is mixed randomly with the top results of the user-based collaborative filtering, so each time the recommendations of the same items are different.

Thus, by combining the previous models, we obtained a RS that predicts movies using information from similar users and items, so the user can get recommendations more adequate to his/her preferences.

V. DIAGRAM OF COMPONENTS

Here is presented a sketch of the main components of the Hybrid RS.

In order to have a better representation of the system performed there is a class diagram [Fig. 1] and a software components diagram [Fig. 2] the class diagram tell us about the implementation in a web page, with a user who can set rating of movies and do the request for the hybrid recommendation system, in the components diagram we can see the interaction of those software components at a high level.

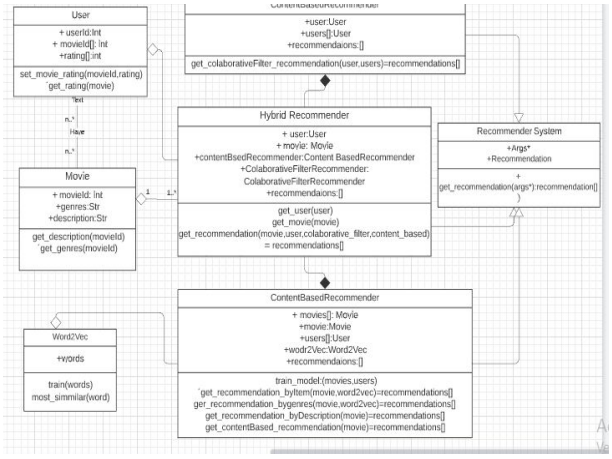

Fig. 1. Class diagram.

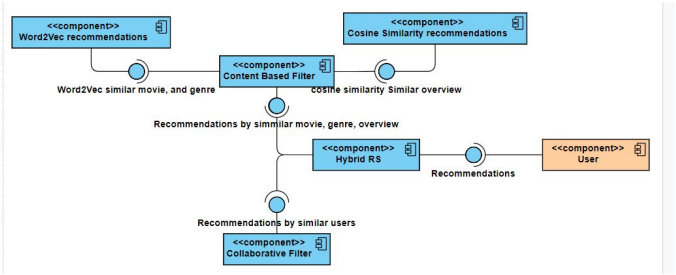

Fig. 2.Software components diagram.

VI. EVALUATION APPROACH

The evaluation approach is not something easy to define and choose in a recommendation system due to the data, and some factors, for example in other fields methods as classifications or regressions, it could be easily evaluated with the results of some evaluation data data different to the data used in train, and verifying the total of hits or misses.

In this case due to the nature of the data, the results and the human evaluations, is very difficult to define which items were selected correctly because in some cases these have not been qualified, but this is also good because it means that it recommends some unexpected items.

Taking this into account, it was decided to measure two things:

1. Precision: The precision were measured by taking only the recommended movies (using training data to recommend) that were viewed by users (using test data to compare) and calculating the true positives and false positives as we can see in the equation [eq. 1], for which was defined a threshold

of 3 stars in the rating; if the rating was higher was a true positive (TP); and if it is lower was defined as a false positive(FP).

2. Serendipity: this was defined as the number of movies in the list recommended that the user has seen in the test data, divide by the number of movies in the recommended list that the user has seen in the test data, added to the number of movies in the list recommended that the user has seen in the test data as we can see in the equation [eq. 2]; in this case the results were high (closer to 1) because the number of total movies is so large, so the probability the user has rated a recommendation in test is small.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Serendipity = \frac{movies\ not\ seen}{movies\ seen + \ movies\ not\ seen} \quad (2)$$

The results of those metrics are the shown in table [Tab. 1]

TP	FP	Precision	Serendipity
218	18	0.9237	0.9903

Tab. 1.Results of the metrics for the Hybrid recommender system.

Other approaches as recall could not be used because the not rated movies could not be defined as false negatives or true negatives.

VII. CONCLUSIONS AND EXPERIENCES

In order to do a good recommender system the data selection and data cleaning are the most important things and it is important to take time to select, analyze and clean the data before start, for this recommender system the data selection and cleaning was what took the longest time and the most complicated part, as a conclusion it is also important to think what is the main object and analyze how the data can help before start developing the recommender, it is not a good experience to have something done and have to redo everything again.

It's necessary to think which will be the last use of the recommender, in the case of a web page there's need to use a model based approach more than a memory based, due to the processing time, if is an offline processing a memory based could be a good approach.

The way how data is mixed for the hybrid recommender system is also important, it is the most important characteristic and where it has all its potential if it is used well. It is important to analyze the proportions of the recommendations; for example, not always recommending based on genres or items. It is important to dive users into varied options, because although it is good to take it to niches in some cases only, recommendations of this type can move users away, and likewise in the opposite direction.

The Evaluation Approaches for a recommender systems are many and can become complex due to the infinity of different characteristics to evaluate, as well as the nature of the data and results, so it is important to research and analyze the possible options.

REFERENCES

[1] Big & personal Xavier Amatriain - Proceedings of the 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining Algorithms, Systems, Programming Models and Applications - BigMine '13 - 2013

[2] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370.

[3] MovieLens," GroupLens, Sep. 06, 2013. https://grouplens.org/datasets/movielens/.

[4] Sugiyama, K., & Kan, M. Y. (2010, June). Scholarly paper recommendation via user's recent research interests. In Proceedings of the 10th annual joint conference on Digital libraries (pp. 29-38).

[5] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science

[6] Decomposing fit semantics for product size recommendation in metric spaces Rishabh Misra, Mengting Wan, Julian McAuley RecSys, 2018

[7] R. Python, "Build a Recommendation Engine With Collaborative Filtering – Real Python." https://realpython.com/build-recommendation-engine-collaborative-filtering/.

[8] "Beginners Guide to learn about Content Based Recommender Engine," Analytics Vidhya, Aug. 11, 2015. https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/.

[9] Trstenjak, B., Mikac, S., & Donko, D. (2014). KNN with TF-IDF based framework for text categorization. Procedia Engineering, 69, 1356-1364.

[10] S. Sondur, A. Chigadani, S. Nayak, "Similarity Measures for Recommender Systems: A Comparative Study", Journal for Research, vol. 20, pp. 76-80, May. 2016.