STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

2020 ECE SUMMER HONORS RESEARCH PROGRAM -

Industrial Recommender Systems in Media & Entertainment

- *By Siddharth Mandgi*
  under Prof Hong Man

WEEK-2

Table of Contents

Recommender Systems in Detail

As we have discussed in our last week's report, recommender systems are essentially AI algorithms used to recommend for similar items to users based on their prior tastes. The goal of implementing Recommender Systems should be to produce recommendations most similar to the product the user has previously purchased or liked. Since our research is based on the Entertainment Industry, we will focus on features such as time taken to view a video or a listen to a music track, likes or dislikes, sentiments for the comments posted and other features including analyzing playlists, most listened to artists, albums or genre. As our research focuses on **Industrial Recommender Systems**, we will also discuss solutions to deal with challenges in Big Data in the coming weeks of our research. But before we discuss our approaches, we need to first understand Recommender Systems in more detail and how we can implement it for a music dataset or a video dataset.

There are many ways to build a recommendation system. Assume that, for a video recommendation, based on videos a user has watched, we can simply suggest same authors videos or same publications videos.

Popularity based

Easiest way to build a recommendation system is popularity based, simply generate recommendations over all the products that are popular. But the question arises as to how we can identify popular products. This can be done by Rating Features (Stars, Likes, thumbs up) and no of users which are all the products that are bought most. Example, in shopping store we can suggest popular dresses by purchase count.

Classification based

In this method we use both features - users as well as products in order to predict whether a video or a song will be liked a user or not. For new users, a ML classifier trained on historic labeled data with a binary class of yes or no (like or dislike), will generate predictions.

Collaborative filtering

These models which are based on assumption that people like things similar to other things they like, and things that are liked by other people with similar taste. They are of two types: -

Nearest neighbors

These type of recommendation systems recommend based on K-nearest neighbors approach used to find out either similar users or similar products. They can be of two types - User Based & Item Based. In user based collaborative recommender systems we find the users who have similar taste of products as the current user, similarity is based on viewing or listening behavior of the user, so based on the neighbor viewing or listening behavior we can recommend items to the current user. For Item based collaborative filters we recommend Items that are similar to the item user has either liked or viewed. similarity is based on co-occurrences of important features.

Matrix factorization

When a user gives feedback to a certain movie they saw (say they can rate from one to five), this collection of feedback can be represented in a form of a matrix. Where each row represents each user, while each column represents different movies. Obviously, the matrix will be sparse since not everyone is going to watch every movie, (we all have different taste when it comes to movies). A benefit of matrix factorization is the fact that it can incorporate implicit feedback, information that are not directly given but can be derived by analyzing user behavior. Using this as an advantage, we can estimate if a user is going to like a movie that (he/she) never saw. And if that estimated rating is high, we can recommend that movie to the user.

Content based filtering

A popular technique of recommendation recommender systems is **content-based filtering.** Content here refers to the content or attributes of the products you like. So, the idea in content-based filtering is to tag products using certain keywords, understand what the user likes, look up those keywords in the database and recommend different products with the same attributes.

Deep Learning based Recommender Systems

Feeding features to Neural Networks to generate recommendations based on different hyper tuned parameters. A case study of how **YouTube** recommends videos was previously discussed which uses neural networks for Candidate Generation, Candidate Sampling and Ranking algorithms is one the best examples of this method.

Sentiment Analysis for Recommender System using NLP

Using Sentiment Analysis to break down comments and predict positive or negative user sentiments can help generate more accurate recommendations.

Research Papers

A few research papers have cited important concepts of AI in Media and Recommender Engines worth mentioning in this week's report.

PolyLens: A Recommender System for Groups of Users

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.127.7633&rep=rep1&type=pdf

This research paper focuses on generating recommendations for a group of users rather than a single user. It is based on Collaborative Filtering Method for a batch of users. This research states that a group recommender is more useful for domains in which several people participate in a single activity, such as viewing a video or listening to a song.

PolyLens is a group recommender extension to the MovieLens recommender system. MovieLens is a free movie recommender site with over 80,000 users and their ratings of over 3,500 movies (with a total of nearly 5 million ratings). MovieLens users rate movies on a five-star scale. The MovieLens front page shows users several lists of recommended films, including movies in theaters and movies recently released on video tape or DVD formats. The front page also provides access to special features, experiments, and a query interface. Users may search for movies by title, retrieving a list of matching movies with predicted ratings, or may select categories of movies by date and genre, retrieving lists of recommendations sorted by prediction. PolyLens was integrated with MovieLens in three places. New links were added to the front page to allow users to create or manage groups; a new field was added to the query interface to allow users to select whether they were looking for group or individual recommendations; and a membership consent interface was added to alert users who were invited to join groups of their pending invitation.

During formation of groups some very important questions were answered such as What is the nature of a group? How do groups form and evolve? How is privacy handled within a group? How are recommendations generated for groups? What interfaces support group recommenders?

Field trial research results show that 95% users were either satisfied or very satisfied with the group recommender engine added along with a single user recommender engine.

A User-Centric Evaluation Framework for Recommender Systems

https://dl.acm.org/doi/pdf/10.1145/2043932.2043962

This paper describes a unifying evaluation framework, called ResQue (Recommender systems' Quality of user experience), which aims at measuring the qualities of the recommended items, the system's usability, usefulness, interface and interaction qualities, users' satisfaction with the systems, and the influence of these qualities on users' behavioral intentions, including their intention to purchase the products recommended to them and return to the system.

Model Development

An evaluation questionnaire consists of a set of constructs, the participating questions for each construct, and the hypotheses relating the constructs. During the development of the model, the researchers compared their model constructs with those used in TAM and SUMI, which are two well-known and widely adopted measurement frameworks.

TAM (Technology Acceptance Model) seeks to understand a set of perceived qualities of a system and users' intention to adopt the system as a result of these qualities, thus explaining not only the desirable outcome of a system, but also users' motivation.

SUMI (Software Usability Measurement Inventory) is a psychometric evaluation model developed by Kirakowski and Corbett to measure the quality of software from the end-user's point of view. The model consists of 5 constructs (efficiency, affect, helpfulness, control, and learnability) and 50 questions. It is widely used to help designers and developers assess the quality of use of a software product or prototype and can assist with the detection of usability flaws and the comparison between software products.

Instead of generating a linear model of sample questions, the researchers structured the question items into four layers of higher-level constructs crossing four dimensions: the perceived system qualities, users' beliefs as a result of these qualities, their subjective attitudes, and their behavioral intentions. Such a topology can more clearly explicate how users' perception of the physical features of a system influences their beliefs, attitudes, and finally behaviors. Finally, the current model was obtained comprising of fifteen constructs and forty-three questions.

Conclusion

This paper presents an overview of recent user experience research in recommender technology. The examination of combined criteria for usability and satisfaction led to the conceptualization of the first balanced measurement framework, ResQue, to assess users' attitudes and acceptance towards a recommender. Most importantly, a Web-based survey confirms that ResQue provides validity and reliability of its structures, and that the proven paths carry meaning causal relationships among the constructs.

A few other papers worth mentioning: -

A Recommendation Model Based on Deep Neural Network
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8247172

Application of Content-Based Approach in Research Paper Recommendation System for a Digital Library
https://pdfs.semanticscholar.org/c9f9/6d22422953625f1f8d9dbee221cba38e6c08.pdf

The Research Development Approach

- As we begin working towards the development side of the project which includes coding and building our recommender engine, we need to focus most on Data Collection and the kind of data we require for our project. As this research focuses on recommender engines for media and entertainment as a domain, we will be exploring approaches to both video and music recommender systems. Thus, we will divide this research into two halves, ***Music Recommender Systems*** and ***Video Recommender Systems***, each having different features and different parameters for generating recommendations.

- In reality the size and kind of data that data scientist work on is not fixed. It could be messy, imbalanced, a data set with very limited information or big data which is very taxing towards computation. Hence collecting data is pivotal to any data science or AI project. A few datasets & APIs I will be working on for this research project are: -

  For Music Recommender Engine: -
  1) API from Spotify
  2) Batches from Million Song Datasets.
  3) Amazon's Music Dataset

  For Video Recommender Engine:
  1) Batches of YouTube 8M datasets
  2) Movie Datasets available on Kaggle

- The algorithms I will be exploring during this research will include basic algorithms such as Matrix Factorization, Classification Based Recommender Systems, etc. and move towards complex algorithms such as Collaborative Filtering, Sentiment Analysis based recommender generation and Deep Learning-based Recommender Systems.

- From this Research I hope to explore the different possibilities of generating the most accurate recommendations to users when it comes to viewing a video or listening to music.

Summary

In this week's report we discussed recommender systems in detail and the different approaches to generate recommendations. We also cited important findings from different research papers and finally spoke about our research development approach.