Enhancing serendipity in Recommender systems using User Clustering

Anamika Lochab
Rutgers University
al1522@scarletmail.rutger.edu

ABSTRACT

Recommender systems have become essential tools for navigating vast information spaces, yet traditional methods often emphasize accuracy and relevance at the expense of serendipity. Serendipitous recommendations, those that are both unexpected and valuable, can enhance user satisfaction, engagement, and discovery of novel content. The objective of this project is to propose a novel approach for incorporating serendipity into recommender systems by combining user-user clustering with collaborative filtering (CF) and clustering algorithms like k-means. We initially employ kNN to predict user ratings and generate top-N recommendations. Subsequently, we utilize the updated CF matrix to identify user clusters and extract the most dissimilar cluster's favorite movies. From these, we select the top-N serendipitous items, resulting in recommendations that are both relevant and serendipitous.

KEYWORDS

Keywords

1 INTRODUCTION

Recommender systems play a vital role in the modern information landscape, assisting users in discovering relevant and personalized content amidst an overwhelming volume of information. While traditional recommender systems focus on accuracy and relevance, recent research emphasizes the importance of serendipity - the phenomenon of unexpected yet valuable discoveries - in fostering user satisfaction and engagement.

The primary contributions of this project are as follows:

(1) An approach that combines user-user clustering with CF results and user-genre matrix to generate serendipitous recommendations.
(2) A comprehensive evaluation of our proposed method, demonstrating its effectiveness in providing relevant yet serendipitous recommendations that improve user satisfaction and engagement.

2 RELATED WORK

A brief review of the related work. (Here is an example of how to include citation [1]).

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

3 PRELIMINARIES

Suppose we have a user set $\mathcal{U} = \{u_1, u_2, \cdots, u_n\}$, an item set $\mathcal{I} = \{i_1, i_2, , i_m\}$ and a genre set $\mathcal{G} = \{g_1, g_2, \cdots, g_k\}$. Let $\mathcal{A} \in [0, 5]^{nxm}$ represent the interaction between a user u_i and an item v_j, and value 0 represents no interaction. Let $\mathcal{B} \in \{0, 1\}^{mxg}$ be the movie-genre matrix where B_{ij} is 1 if movie i has genre j. And, Let $\mathcal{C} \in [0, 1]^{nxg}$ be the user-genre matrix where the value of C_{ij} represents how much user i likes that genre j.

4 PROBLEM FORMALIZATION

Before diving into the proposed approach, it is crucial to formally define the problem of incorporating serendipity into recommender systems using user-user clustering and collaborative filtering.

The objective of user-user clustering to provide a set of top-N serendipitous recommendations S(u) for each user u, such that S(u) $\subseteq \mathcal{I}$. We aim to set a trade-off between the obvious recommendations R(u) and the serendipitous recommendations S(u). These recommendations should be a combination of relevant and serendipitous items to the user, leading to improved user satisfaction and engagement.

Relevance refers to the extent to which recommended items match the user's preferences or needs, while serendipity pertains to the unexpectedness or novelty of the recommendations. We define a recommendation to be relevant if its predicted rating or interaction likelihood surpasses a predefined threshold. Serendipity, on the other hand, is achieved when the recommended items are not only relevant but also unexpected, leading to a pleasant surprise for the user.

5 THE PROPOSED MODEL

5.1 User-based Collaborative Filtering

I will be using KNN method for collaborative filtering.

(1) Finding k nearest neighbours using cosine similarity between users. It can be done using the Cosine similarity formula:

$$w_{u,v} = \frac{\Sigma_{i \in \mathcal{I}} r_{u,i} r_{v,i}}{\sqrt{\Sigma_{i \in \mathcal{I}} r_{u,i}^2} \sqrt{\Sigma_{i \in \mathcal{I}} r_{v,i}^2}}$$

This will be computed and stored in a matrix. And based on similarity we will choose the top-k+1 most similar user (the most similar will be the user itself) and store them in list K.
(2) Finding candidate items from the k Neighbours and removing the items the current user has already rated in list KC.

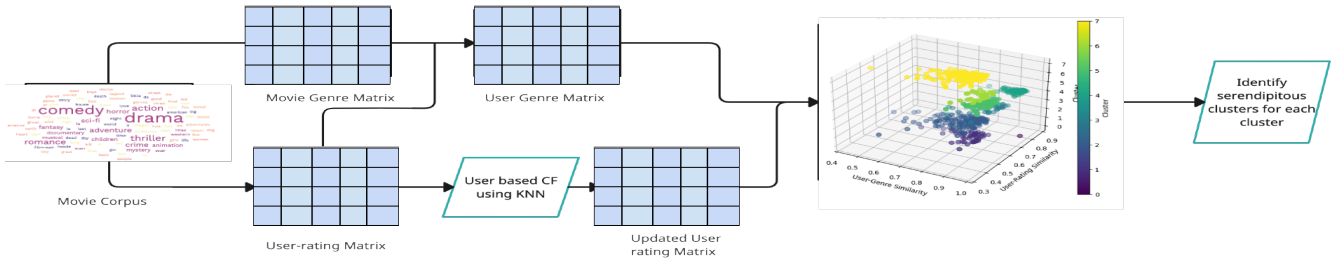

Figure 1: Workflow

(3) Computing the rating for the candidate items.

$$\hat{r_u i} = \overline{r_u} + \frac{\Sigma_{v \in K} w_{u,v}(r_{v,i} - \overline{r_v})}{\Sigma_{v \in K} w_{u,v}}$$

We will compute the rating for each element in the candidate sets, and update in our matrix for rating prediction.

(4) For top-N recommendation, we will sort all the candidate items based on their ratings and choose the top-N.

5.2 User-genre matrix

For the clustering process we will be using a concat of the updated rating matrix and the user-genre matrix. To get the user genre matrix the formula used is :

$$C_u = \Sigma_{i \in S} B_i * A_{u,i}$$

where S is the set of movies user u has rated. B_i is the row of movie-genre matrix representing the genres of the corresponding movies.

Then each row of C is normalized to avoid bias.

5.3 Clustering Users

For clustering users we will be using k-means. Consider a matrix $\mathcal{D}$ which is a concat of the updated rating matrix and the user-genre matrix along the users.

(1) For the matrix D, perform the k-means algorithm for a range of values and find the best k value using elbow method or silhouette score.

(2) Obtain k cluster and for each cluster we identify the favourite movies of the users, and based on the favourite movies, we determine the jaccard similarity between the clusters.

$$J_{c_1,c_2} = \frac{|LikedMovies_{c_1} \wedge LikedMovies_{c_2}|}{|LikedMovies_{c_1} \vee LikedMovies_{c_2}|}$$

(3) For each user cluster c we find serendipitous clusters with similarity less than threshold α

(4) The liked movies of the serendipitous clusters are combined and added to the serendipitous movie list for cluster c.

(5) Then we compute the top N serendipity movies for each user, using unexpectedness of a movie compared to the users's choice

$$GenreDiff_{u,i} = \Sigma_{g \in \mathcal{G}} |C_u g - B_i g|$$

$$Unexp_{u,i} = \frac{GenreDiff_{u,i}}{|G|}$$

(6) Sort the above list based on serendipity score and get the top N Serendipity Movies.

5.4 Re-Ranking

Considering the top-N Recommended movies from CF and the top-N serendipity movies. We need a re-ranking algorithm that keeps a trade-off between recommending the movies which are obviously liked by user but also including serendipitous movies in the list, while considering the user preference.

(1) List = top-N CF Recommended movies + top-N serendipity movies

(2) Relevance Score using cosine similarity between user-genre and movie genre vectors:

$$Rel_{u,i} = \Sigma_{g \in \mathcal{G}} C_{ug} B_{ug}$$

(3) Unexpectedness Score:

$$Unexp_{u,i} = \frac{GenreDiff_{u,i}}{|G|}$$

(4) Trade off using s (serendipity weight):

$$hybrid_score = (1 - s) * Rel_{u,i} + s * Unexp_{u,i}$$

We can now sort the list based on hybrid_score and provide the top-N movies.

6 EXPERIMENTS

6.1 Dataset Description

In this study, we use the MovieLens dataset, a widely adopted benchmark for evaluating recommender systems, particularly in the domain of movie recommendations. MovieLens is provided by GroupLens Research and contains user-generated movie ratings alongside movie metadata. We will be using the ratings and movies file for this project.

6.2 Evaluation

We evaluate our CF model on the following metrics :

MAE	0.6828
RMSE	0.90325
Precision	
Recall	

	userId	movieId	rating	title	genres
0	200	79132	4.721432	Inception (2010)	Action|Crime|Drama|Mystery|Sci-Fi|Thriller|IMAX
1	200	858	4.693372	Godfather, The (1972)	Crime|Drama
2	200	50	4.677888	Usual Suspects, The (1995)	Crime|Mystery|Thriller
3	200	5618	4.653791	Spirited Away (Sen to Chihiro no kamikakushi) ...	Adventure|Animation|Fantasy
4	200	3275	4.619820	Boondock Saints, The (2000)	Action|Crime|Drama|Thriller
5	200	293	4.612219	Léon: The Professional (a.k.a. The Professiona...	Action|Crime|Drama|Thriller
6	200	48516	4.549239	Departed, The (2006)	Crime|Drama|Thriller
7	200	527	4.532772	Schindler's List (1993)	Drama|War
8	200	2329	4.510015	American History X (1998)	Crime|Drama
9	200	1617	4.484232	L.A. Confidential (1997)	Crime|Film-Noir|Mystery|Thriller

	movieID	title	genres
0	117646	Dragonheart 2: A New Beginning (2000)	Action|Adventure|Comedy|Drama|Fantasy|Thriller
1	1112	Palookaville (1996)	Action|Comedy|Drama
2	33672	Lords of Dogtown (2005)	Action|Comedy|Drama
3	496	What Happened Was... (1994)	Comedy|Drama|Romance|Thriller
4	5666	Rules of Attraction, The (2002)	Comedy|Drama|Romance|Thriller
5	3893	Nurse Betty (2000)	Comedy|Crime|Drama|Romance|Thriller
6	106594	Red Flag (2012)	Comedy|Drama|Romance
7	73881	3 Idiots (2009)	Comedy|Drama|Romance
8	232	Eat Drink Man Woman (Yin shi nan nu) (1994)	Comedy|Drama|Romance
9	351	Corrina, Corrina (1994)	Comedy|Drama|Romance

Figure 2: Top 10 CF recommendation and top 10 serendipity recommendation

7 CONCLUSIONS AND FUTURE WORK

Conclude the work and make discussions about future directions.

ACKNOWLEDGEMENT

Acknowledges for the help you received such as useful discussions with colleagues.

REFERENCES

[1] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.