01/06

Research Methodology ⇒ Collecting literature for literature review

Interest :- Content Recommendation

Movie Recommendation ⇒ Start collecting literature

(Need to check what models are used nowadays for Movie Recommendation).

* I will also review about Netflix, as it is currently famous for movie database and recommendation. (Need one section for that in literature review)

* Check for database available :- which one will be feasible!

* Thinking to recommend movies based on
  - ↳ genre
  - ↳ cast
  - ↳ year
  - ??

Plan this later...

02/06

⇒ Finalized Movie Recommendation system.

Various literatures :— (Need to select best few) for review

1. Movie GEN: A Movie Recommender System

✓ Uses hybrid recommendation approach
↳ with the help of machine learning and cluster analysis

One of the approach used to hybrid is collaborative approach. user preference (i/p)

↓

Machine learning model ⇒ SVM

↓

Prediction (o/p) of movie title

further i/p

↓

K-Means Clustering

↓

Cluster of similar movies.

2. Movie mender : Movie Recommendation System

↳ uses hybrid approach ⇒ collaborative ~~feat filtering~~ + Content - based

✡ Takes user-ratings matrix as database using web crawler.

↳ Fill it using Content-based predictor

⇒ Recommendations are provided by collaborative filtering

3. An Improved approach for Movie Recommendation System :- Hybrid Approach (Again!!!)

Content-based + collaborative filtering

✡ SVM as classifier & genetic algorithm for optimization

↳ Classified Movies from SVM are clustered using k-Means for Recommendation purpose.

05/06

My Project idea :- (as of now)

* Movie Recommendation based on some criterion (genre, title, year)
* Hybrid approach (for sure) !!
* Need to check which classifier/method can be used for approach.

* If time persists ?!! ... will try for UI.

Review Research continued :-

4. Netflix Recommender System :- → Definitely good paper for review

✓ ↳ Good paper about different Recommendation algorithms used by Netflix.

↳ Can inspire for 2-3 ideas (for sure)

↳ Gives different perspective of how any recommendation can be built i.e., what users want as good recommendations?

5. Personalized Movie Recommendation based on Deep Learning.

↳ Collaborative - filtering based approach

↳ uses deep learning models

↳ Seq2Seq model based on LSTM RNN

Complex Research paper ; but good one as it uses deep neural net approach

6. Movie Recommendation System using Sentiment Analysis from Microblogging Data

↳ hybrid approach

Tweets (User reviews) ⇒ Database } unique set

⇒ Sentiment analysis by VADER employed to provide recommendations.

7. Movie Rec ⇒ Hybrid approach

* using cumulative wt. of different attributes
* uses k-means algorithm.

10/06

Literature ~~Rev~~ Review → Selected Papers

1. Movie GEN: → hybrid approach (use traditional model)
2. Netflix Recommender System → gives idea about ~~diffent~~ different recommendation techniques
3. Movie Recommendation using microblogging data → different kind of dataset (tweets)
   ↳ leads to unique approach
4. Personalized Movie Recommendation based on deep learning → Neural Net. (Deep Learning approach).

Review Paper Structure :- 1. Abstract : (Summary)
2. Introduction : About Recommendation System :- * general context
   * types of approach
   ↳ Further, a crisp intro. about movie ~~reco~~ recommendations.
3. Literature Review :- Structure in order of each paper.

but start with 'Netflix Recommender' paper
↳ It gives a good base...
Then all 3 papers
4. Conclusion : (whole paper in 1 para).

↓

Final Review Paper Structure.

11/06 ⇒ Change of plans for review paper ⇒ Not including deep learning paper (too complex and taking space)
* Cannot remove anyone of other 2 papers as they are uniquely setting the tone of review.

⇒ Recommendation techniques :- Recommender Systems Handbook, 2nd ed.
~~Content - Based filtering~~
Content - Based :- ~~filtering~~ :
User - Ratings matrix :-
matrix of m users and n items

Each row ⇒ users ; Each column ⇒ items

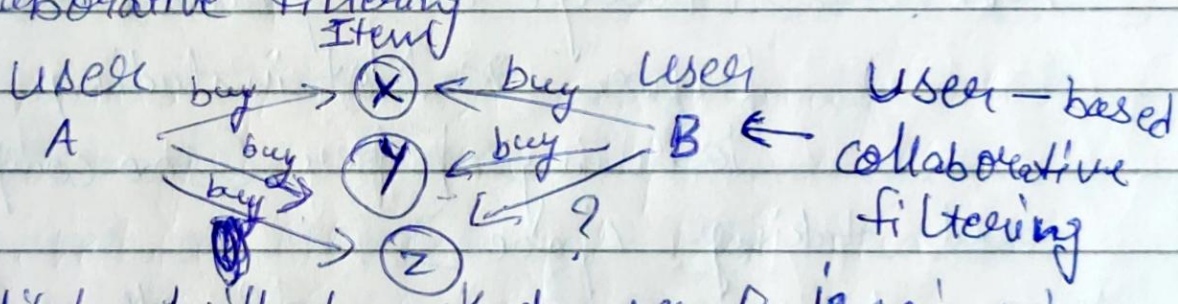
(i,j) cell ⇒ rating given by user i to item j.

★ This matrix is usually Sparse as users have not rates most movies.

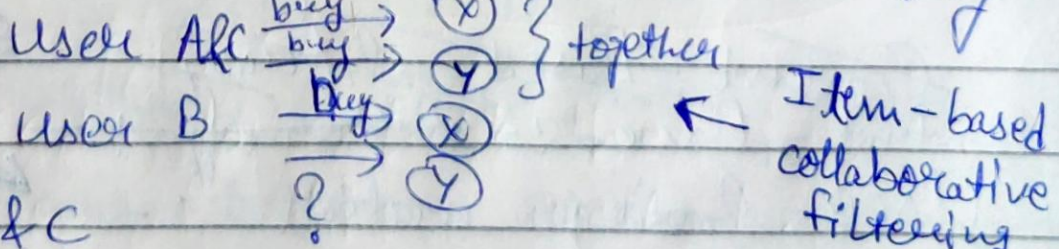
⇒ To predict those Empty values ⇒ To recommend movie

↳ type of content-based

⇒ Collaborative filtering



How likely will be that user B is going to buy item Z, given Z is bought by A too? Since both users have bought similar items X, Y in past, it is highly likely that user B will buy item Z too. ⇒ User-based filtering



If user A ~~&~~ B buy item X & Y together, then there are high chance that both items will be

bought by B ! ⇒ item-based filtering

15/06

⇒ Idea checked before :- create a movie recommendation system based on genre, ~~title~~, year, cast.

↓

☆ This demands similarity between two movies in order to get us good recommendation

↳ Metadata-based recommender.

☆ Need to check techniques which uses similarity measure :- vector representation

Count Vectorizer

represents count of each word in vocabulary according to its frequency, in form of one-hot vector.

↓↓

If word present in doc, then +1, else 0.

TF-IDF vectorizer

assigns weight along with ~~frequency~~ to each ~~for~~ word.

TF ⇒ term frequency in a document

IDF ⇒ No. of documents that contain (take) term t

Weight ⇒ <0, 1>

~~Most of the related~~ work used this

⇒ TF-IDF vectorizer uses cosine similarity for checking similarity b/w 2 texts/values.

$$\text{cosine} = \frac{a \cdot b}{|a||b|}$$

↓

Higher the cosine similarity, more similar two ~~~~ text will be.

↳ This will make pretty good model!

☆ Advantage :— ☆ I understand this concept well.

☆ Can be helpful to get basic idea of content-based (metadata) movie recommendation.

☆ Fulfill my idea for recommending movies based on genre, ~~title~~, year, cast provided by user.

Need to check the implementation of this method.

* Will help in NLP too (currently on obj feature)

20/05 Various Collaborative - filtering techniques :-

* Similarity measures
* Supervised learning
* Clustering

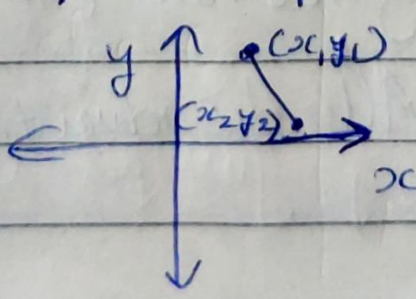
* Remember user - ratings matrix.
  ↳ Need to fill the empty cells of matrix.

* Empty cells i.e. rating can be predicted by finding users who has similar taste ~~in~~
  ↳ we can ~~calculate~~ check which user has similar taste by measuring distance
  ↳ Euclidean distance &
  Pearson correlation.

2 measures popularly used

⇒ Euclidean distance :- dist. b/w 2 datapts. plotted in n-dimensional space.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



⇒ Plotting of n-dimensional space

⇒ Pearson correlation :- Ranges b/w −1 to 1

−1 ⇒ negative correlation (totally opposite)

0 ⇒ no relation

+1 ⇒ positive correlation (very similar)

★ Better than euclidean as it checks for both similarity & dissimilarity.

30/06 ⇒ Research Proposal ⇒ not perfect plan selected.

★ Need to give any one above learned method in detail.

★ Will be discussed with Dr. T if any further changes required.

★ Structure of proposal ⇒ Abstract (summary) ⇒ Intro. & Background ⇒ Detailed intro. about movie recommendation system & various approaches.

↳ Research Question (most imp. part) ⇒ All disadvantages of basic model.

⇒ Research Methodology ⇒ Methods used; approach used.

⇒ Schedule (whole schedule of research with Gantt chart)

⇒ Summary (conclusion of Research Proposal)

Research Proposal ⇒ Hybrid Approach

Content-based

Collaborative-filtering

Disadvantage :- cold-start problem

↳ ~~recom~~ Recommendations cannot be made for new user.

Research methodology

methods proposed ↳ Cosine similarity for checking similarity b/w users & movies

↳ TF-IDF vectorizer to give weightage to different metadata of movie

★ My main approach still stands with the idea that user will be able to get movie recommendations based on genre/year/cast (any detail i/p user gives).

10/07 Searching for Dataset

↳ checking other literature

↳ Kaggle

↳ github

⇒ Checked various literatures related to movie recommendation systems :-

★ Most of them are using 2 datasets ~~(open-sourced)~~ Not sure!!

1. MovieLens dataset (grouplens.org/datasets/movielens)

2. TMDB dataset (themoviedb.org)

⇒ I will be using MovieLens dataset as it has multiple variants available in various sizes. Also, imdb dataset need to have api.

★ Movielens dataset can be directly used in colab using just wget & unzip.

11/07 MovieLens dataset (Different variants)

★ I will be using only 4 variants, ~~either~~ because ~~either~~ it is used in many literatures and results can be compared. ~~or it is~~

Dataset can be changed!

1. ml-latest-small.zip
   - ↳ smallest dataset
   - ↳ does not have other metadata such as tags.
2. ml-latest.zip
   - ↳ largest dataset
   - ↳ let's see if it fits first!
3. ml-20m.zip
   - ↳ not too large; not too small
   - ↳ 20M movie ratings
   - ↳ popularly used in most research works.
   - ↳ stable dataset
4. ml-25m.zip
   - ↳ too large dataset,
   - ↳ let's see if it fits first!
   - ↳ stable dataset

15/07 Going through datasets for ~~checking~~ learning the data!

⇒ Exploratory analysis ⇒ will help for data pre-processing!

Important files

Movies-25M
- → ratings.csv
- → movies.csv
- → tags.csv

Movies-latest-small
- → ratings.csv
- → movies.csv

✡ User ~~details~~ ratings :- Only unique userId provided ratings for each movie

✡ Movies.csv ⇒ movieId, genre, title

I guess these files will be enough for ~~R~~research project implementation.

⇒ Selected both dataset as while ~~testing~~ checking similarity between movies/users, large dataset will give us hard time!!

16/07 Final Report ⇒ Introduction

I will write a brief introduction about Recommendation System.

↳ different types of recommendation systems

⇒ General overview of Movies recommendation system ⇒ what input/output is expected?

Related Work: Thinking to write regarding Netflix ⇒ That will be helpful as I have thought of research project structure.

It has resemblance to Netflix Recommendation algorithms: (resemblance as in algorithms used for recommendations).

+abstract

This^ will constitute starting of Final Report!!

20/07 Started coding part for my research project !

General Idea :- To provide any new user a set of movie recommendations

★ user need to provide simple inputs
  ↳ number of recommendations, n
  ↳ year_range; from_year, to_year
  ↳ genre

★ These inputs can be easily given by any user who wants to check out movies. Simple inputs !!

Core Idea ⇒ ratings.csv has many user rating different movies

Popularity ↓ movie which is rated by many users

& Ratings ↓ movie which is highly rated

If both Popularity & Ratings are weighted accordingly, then ...

a good movie recommendations can be provided.

☆ Since this will utilize release year and genre of movies, this is ~~our content~~ = content-based recommendation system.

☆ Data pre-processing :-

↳ ratings.csv and movies.csv has been used.

↳ timestamp from ratings.csv has been removed

↳ Since release year is provided with movie title, it has been extracted seperately and placed in new column → year

↳ Typecasting ~~year~~ year, genre } datatypes ⇒ { int, str

☆ Since there is year-range, few if-else has been placed so user does not give wrong input ⇒ ☆ genre must be from given list

☆ from_year < to_year

21/07 Top - N Popular Movies

⇒ Data preprocessing completed !

⇒ input bounds provided so user does not feed wrong input

⇒ Core idea implementation :-

✯ Count of users who rated each movie has been extracted by checking ratings provided to each movie grouped by movieId

✯ Movies are filtered ⇒ movies which has less ratings than avg rating and less count of ~~people~~ user who rated it ⇒ removed

✯ Further, selected list is arranged by most number of people who rated movie.

↳ Top - N Popular movies Recommender system completed !!

22/07 Top - N Rated Movies

⇒ Data preprocessing completed (same)

⇒ input bounds provided so user does not feed wrong input (same)

⇒ Core idea implementation :-

✱ Count of user who rated each movie will be calculated by grouping using movieId.

✱ Average rating of movie will be calculated once the total count of users are calculated.

⇒ Movies will be filtered ⇒ same as Top - N Popular; movies having less than avg. rating and avg. count will be filtered out.

⇒ Finally, selected movies will be arranged as per highest rating provided by users.

↳ Top - N Rated Movies Recommender System completed!!

25/07 ⇒ Final Report ⇒ Methodology

↳ Hybrid approach

↳ Since, using both content-based and collaborative filtering approaches

⇒ I will mention both approaches in perspective of movie recommender systems.

Content-Based

↓↓

uses metadata of movie (year, genre) for recommendations

user need to have basic idea about movies.

⇒ Simple I/ps ⇒ year, genre

Collaborative filtering

↓↓

user ratings of input movie to check similarity with other movies;

(other users who rated same input movie - what they watched) as recommendations.

⇒ Also mention both similarity measures
↳ • using Pearson correlation coefficient
↳ using cosine similarity.

This constitutes methodology section of Final Report!

Note:- Dataset description need to be included in Implementation section

---

30/07 Collaborative - filtering approach
↳ user need to provide movie
↓
movie ratings will be extracted which will give us users
↓
other movies rated by these users will help us to get selected movies for recommendations

Calculated using similarity score

31/07

Top-N Similar Movies by Cosine Similarity

⇒ Same data preprocessing as content based movie recommender.

★ User-ratings matrix is calculated which gives details of each movie title rated by each user

★ Cosine similarity is measured between each movie ratings provided by each user.

★ Total count of ~~rec~~ users who rated movie is also considered, where only top 0.01% is selected

Top-N Similar Movies by Pearson Correlation

⇒ Same data preprocessing as content based movie recommender.

★ Here also user-ratings matrix will be created.

★ corrwith function used with method="pearson" to check similarity between movie ratings.

★ Here also, total count of users who rated movie (only top 0.01%) is selected

01/08 Implementation Section of Final Report

↳ Dataset description :- Details about dataset used; little bit description about main/important .csv files.

↳ Data Preprocessing :- Preprocessing of data before implementing functions for content-based and collaborative filtering recommendations.

↳ Implementation of content-based movie recommender systems
  ↳ Top-N Popular Movie Recomm. System
  ↳ Top-N Rated Movie Recomm. System

↳ Implementation of collaborative filtering ~~base~~ movie recommender systems
  ↳ Top-N Similar (by Pearson Correlation) Movie Recomm. System
  ↳ Top-N Similar (by Cosine Similarity) Movie Recomm. System

05/08 Evaluation for Recommender systems

⇒ considering user-surveys as movies recommended will be evaluated effectively by humans.

☆ 10 random users (family & friends)

⇒ Testing of Content-Based Movie Recommender System ⇒ number of recommendations are kept ~~same~~ fixed ⇒ 10

Users	from_year	to_year	genre	Popular	Rated
1	2011	2015	Action	10/10	9/10
2	1991	2000	Romance	7/10	7/10
3	2005	2015	Mystery	10/10	8/10
4	2001	2015	Fantasy	9/10	6/10
5	2011	2019	Sci-Fi	10/10	10/10
6	2005	2015	Animation	10/10	5/10
7	1994	2019	War	8/10	8/10
8	2011	2019	Adventure	10/10	9/10
9	2011	2019	Horror	8/10	6/10
10	2005	2010	Thriller	9/10	8/10

* ~~All~~ year-range & genre is fixed by each user to evaluate top-N popular & rated movies

* Observations :- * Top-N popular Movies Recommender was effective! 7 users out of 10 ~~that~~ liked atleast 9 movies out of 10 recommendations.

* Top-N Rated was not effective compared to Top-N Popular ⇒ Only 3 users out of 10 liked atleast 9 movies out of 10 recommendations.

06/08

⇒ Testing of Collaborative filtering recommender

Users	from	to	movie	Cosine	Pearson
1	2005	2015	Inception	9/10	10/10
2	2000	2018	Ratatouille	9/10	7/10
3	1994	2004	Ocean's 11	10/10	10/10
4	2000	2018	Gladiator	9/10	8/10
5	2011	2018	Conjuring	9/10	10/10
6	2012	2018	Deadpool	9/10	9/10
7	2008	2018	Avengers	9/10	8/10
8	1994	2000	Fight Club	10/10	9/10
9	1991	2000	Pulp fiction	9/10	9/10
10	2001	2018	Lalaland	9/10	8/10

★ Both similar movies recommender did good job.

★ However, movies recommended using cosine similarity ~~of will be~~ was very effective.

★ Also, the user survey helped to avoid cold-start problem, since users found atleast 2 new movies from recommendations.

★ Few users actually watched them and responded that they liked the ~~new~~ unseen movies recommended by recommender system.

⇒ Results & Discussion in final Report

↳ Need to plot ~~abo~~ previous mentioned results.

★ ~~Or~~ Barplot will help to visualize the effectiveness of recommender system.

★ Result will be in ~~form of~~ how many movies did user liked from the recommendations!

10/08 ⇒ Few final sections and touch-up of Report left ⇒ Abstract, Conclusion,

⇒ Github repository is created and code notebooks will be uploaded.

⇒ Presentation Structure :-

1 min. Introduction :- About Movies Recommender System

2 mins. Methods :- All 4 notebooks explained

1 min. Results :- Results of user-survey presented

1 min. Discussion :- Comparison & Discussion from above results.

~5 mins

4 Deliverables need to be submitted by 15th August ⇒ Report, Repository, Presentation and Research Journal !!

15/08

D - DAY

Conclusions :- Good research work

Basic concepts of Movie Recommender system is studied and researched well.

Next ?: Thinking to increase complexity by bringing deep-learning models.

→ Learn about 'Surprise' library

→ Will try playing with GUI !

★ Thanks to Dr. T for effectively distributing Research work. Able to do research from scratch!

Final Deliverables

⇒ Presentation → Completed

⇒ Final Report → Completed

⇒ Journal → Completed

⇒ Github Repository → Completed

X