Diverse Approaches for Machine Learning Based Recommendation System

[1]Shivang Agarwal, [2]Shivang Kanwar, [3]Simranjeet Singh Dua, [4] Piyush Bhardwaj
[1,2,3]Student B. Tech, [4]Assistant Professor
[1,2,3,4]Department of Computer Science, BPIT, GGSIPU

Abstract - Recommendation system assumes essential job in Internet world and is utilized in numerous applications. It has made the accumulation of numerous applications, created global village and development for various data. Today there is a major wide range of methodologies and algorithms of data filtering and recommendation. In this paper we portray the proposal framework related research and after that presents different strategies and methodologies utilized by the recommender system User based approach, Item-based approach, Hybrid suggestion approaches, Association rule learning, Clustering and related research in the recommender system. At last we will demonstrate the fundamental difficulties and issues recommender systems go over.

Keywords - Recommendation, Collaborative filtering, Model based, Memory based, Content based, Hybrid, Clustering, Apriori.

I. INTRODUCTION

Usually named as Recommender Systems, they are basic algorithms which plan to give the most significant and precise things to the client by separating valuable stuff from of a tremendous pool of data base. Recommendation engines finds information designs in the informational collection by learning purchasers' decisions and produces the results that co-identifies with their requirements and interests.

Recommendation system is a sharp system that gives thought regarding thing to clients that may intrigue them a few examples are amazon.com, motion pictures in movie lens, music by last.fm. In this paper distinctive approaches with their methods are referenced to look at the confinement of every procedure in legitimate way to give appropriate future recommendations.

II. BACKGROUND

Collaborative filtering is winding up extraordinarily well known as it contributes in diminishing data over-burden. Collaborative filtering-based recommender system predicts new things of enthusiasm for a client dependent on connections registered between that client and different clients. A no. of frameworks is based on the use of partitioning/clustering algorithm on ratings dataset which are then followed by collaborative filtering for developing a Movie Recommender System.

Approaches of Recommendation System

Recommendation system is usually classified on rating estimation

- Collaborative Filtering system
- Content based system
- Association rule learning based
- Hybrid system

In content-based approach, comparative things to the ones the client favoured in past will be prescribed to the client while in collaborative filtering, things that comparative gathering individuals with comparable tastes and inclinations like will be suggested. To defeat the restrictions of both methodology cross breed frameworks are recommended that consolidates the two methodologies in some way [9].

I. Collaborative filtering system

Collaborative filtering produces suggestions dependent on the information of clients' mentality to things, that is it utilizes the "wisdom of the group" to prescribe things. Collaborative filtering systems work by gathering user comment as evaluations for things in a given field and abusing similitudes in rating activities among a few clients in deciding how to prescribe a thing. Collaborative filtering systems prescribe a thing to a user dependent on opinions of different users. Like, in a film suggestion application, Collaborative filtering system attempts to discover other similarly invested users and after that prescribes the movies that are most preferred by them. Although there are many collaborative filtering techniques, they can be divided into two major categories [9]:

- Memory Based approaches
- Model Based approaches

Memory based Approach Memory-Based Collaborative Filtering methodologies can be isolated into two primary areas: user-item filtering and item-item filtering. A user-item filtering will take a specific user, discover users that are like that user dependent on similarity of ratings, and prescribe things that those comparative users preferred.

Interestingly, item-item filtering will take an item, discover users who preferred that thing, and find different things that those users or comparative users additionally loved. It takes things and yields different things as proposals.

- Item-Item Collaborative Filtering: "Users who liked this item also liked …"
- User-Item Collaborative Filtering: "Users who are similar to you also liked …"

After we have built the user-item matrix you calculate the similarity and create a similarity matrix.

The similarity values between items in Item-Item Collaborative Filtering are measured by observing all the users who have rated both items.

For User-Item Collaborative Filtering the similarity values between users are estimated by watching every one of the things that are appraised by the two users.

A distance metric normally utilized in recommender systems is cosine similarity, where the appraisals are viewed as vectors in n-dimensional space and the likeness is determined dependent on the point between these vectors. Cosine similarity for users a and m can be determined utilizing the formula beneath, where you take dot product result of the user vector u1 and the user vector u2 and isolate it by multiplication of the Euclidean lengths of the vectors.

Merits and Demerits of Memory Based Approach

User based procedures relate users by mining their (comparative) ratings and afterward suggest new things that were favoured by comparative users. Item based methods connect the items by mining (comparative) ratings and after that prescribe new, comparable items. The primary points of interest of the two systems are that they use data that is given base up by client ratings, that they do not belong to any category or domain and require no content analysis and that the nature of the suggestion improves after some time. CF procedures are restricted by various disservices. Above all else, the purported „cold start" issue is because of the way that CF strategies rely upon adequate user performance from the past. Even when such systems have been running for some time, this issue develops when new users or things are included. New users initially need to give an adequate number of ratings for items so as to get precise suggestions dependent on user-based CF (new user issue) [8]. New items must be evaluated by an adequate number of users on the off chance that they are to be recommended. Another drawback for CF systems is the sparsity of the past user activities in a system. Since these systems manage network driven data, they bolster very much-loved tastes more unequivocally than disliked tastes. The learners with a surprising taste may get less subjective proposals, and learners with basic taste are probably not going to get disliked things of top quality prescribed. Another basic issue is adaptability. RSs which manage a lot of information, as amazon.com, must almost certainly give suggestions progressively, with the quantity of both the users and things surpassing millions [8].

2) Model Based Approach

In model-based CF algorithms, a hypothetical model is proposed of user rating conduct. As opposed to utilize the raw rating information specifically in making predictions, rather the parameters of the model are evaluated from the accessible rating information and the model is utilized to make predictions. Many model-based CF algorithms have been considered in the course of the most recent years. For instance, examines two probabilistic models, in particular, clustering and Matrix factorization. In four portioning based clustering algorithms are utilized to make predictions, prompting better versatility and precision in contrast with random partitioning [4].

Techniques of Model Based Approach

K-MEANS CF: k-means clustering is applied to distinguish the sections. k-means is a clustering technique that has discovered wide application in data mining, insights and AI. The input to k-means is the pair-wise separation between the things to be bunched, where the separation implies the uniqueness of the things. The quantity of clusters, k is pair-wise an input parameter. It is an iterative algorithm and begins with an arbitrary parcelling of the things into k groups. Every cycle, the centroids of the clusters is processed, and everything is reassigned to the cluster whose centroid is nearest. The Algorithm is Described Below [4]:

Algorithm k-means clustering [4]

1. Input: R = r1 … rm
2. Function kmeans (R*;* k)
3. c*i* = r*pi; ∀*r*pi* ∈ R*; ∀*ci* ∈ C*; ∀*i* = 1*; : : ; k*;*
4. While (*k* C*0 ¡* C *k6*= 0)
5. C*0* = C;
6. *Ci = {j: sj; i ≥ sj;i* ; ∀i** = 1*; : : : ; k}; ∀i* = 1*; : : : ; k*;*
7. $ci = \frac{\sum rj}{|ci|}$ *; ∀j ∈ Ci; ∀i* = 1 *;: : : ; k*;*
8. End While 9. return C*0*.

CLUSTER MODEL: To discover customer who are like the user, group models partition the customer base into numerous portions and treat the errand as a classification problem. The algorithm will probably appoint the client to the fragment containing the most comparative clients. To discover customer who are like the user, cluster models isolate the customer base into numerous portions and treat the task as a classification problem [2].

The algorithm's will likely dole out the user to the portion containing the most comparative customers. It at that point utilizes the buys and evaluations of the customers in the segment to produce recommendations. The sections commonly are made utilizing a clustering or other unsupervised learning algorithm, albeit a few applications utilize manually decided segments. Utilizing a similarity metric, a clustering algorithm bunches the most comparable clients together to frame groups or sections. Since ideal clustering over vast informational sets is unreasonable, most applications utilize different types of greedy cluster generation. They then continuously match customers to the existing sections, usually with some arrangement for making new or consolidating existing segments. When the algorithm creates the sections, it figures the user's similarity to vectors that condense each segment, at that point picks the portion with the most comparability and characterizes the user as needs be. A few algorithms group users into numerous sections and depict the strength of every relationship. Cluster models have preferable online adaptability and execution over collaborative filtering in light of the fact that they contrast the user with a controlled number of segments instead of the whole customer base. The complex and expensive clustering computation is run offline. Nonetheless, recommendation quality is low. Cluster models assemble various clients together in a segment, coordinate a client to a segment, and afterward consider all customers in the segment comparable customers to make recommendations. Since the comparable users that the cluster models find are not the most comparable clients, the recommendations they produce are less significant [2].

MATRIX FACTORIZATION: Model-based collaborative filtering is based on matrix factorization (MF) that has received higher exposure, primarily as an unsupervised method of learning for latent variable decomposition and reduction of dimensionality. Matrix factorization is widely used in recommendation systems where scalability and sparsity can be better addressed than CF based on memory. MF's goal is to learn users' latent preferences and the latent attributes of items from known ratings (learn features that describe rating characteristics) and then predict unknown ratings through the dot product of users and items' latent features. You fit this matrix to approximate as closely as possible your original matrix by multiplying together the low-rank matrices, which fill in the missing entries in the original matrix. Singular value decomposition (SVD) is a well-known method of matrix factorization. Collaborative filtering can be formulated with the use of singular value decomposition by approximating a matrix X. The general equation can be expressed as follows:

$$X = USV^T \qquad 1$$

Given m x n matrix X:

- *U* is an *(m x r)* orthogonal matrix
- *S* is an *(r x r)* diagonal matrix with non-negative real numbers on the diagonal
- *V^T* is an *(r x n)* orthogonal matrix

Diagonal elements in S are known as X's unique values. Matrix X can be factorized to U, S and V. The U matrix represents the user-related feature vectors in the hidden feature space and the V matrix represents the feature vectors in the hidden feature space corresponding to the items.

III. CONTENT BASED APPROACH

Any system implementing a content-based recommendation approach analyses a set of documents and/or descriptions of items previously rated by a user and builds a user interest model or profile based on object features rated by that user. The recommendation process essentially involves matching the user profile attributes against a content object's attributes. The result is a judgment of relevance which represents the level of interest of the user in that object. If a profile correctly reflects user preferences, the efficacy of an information access process has a tremendous advantage [13].

Methods for Content Based Feature Selection [7][9]

1. **Wrapper methods** evaluate various subsets of features by training a model for each subset and then evaluate the contribution of each subset to a validation dataset. Since the number of possible subsets is factorial in the number of features, various heuristics are used to select "promising" subsets (forward-selection, backward-elimination, tree-induction, etc.). Wrapper methods are separate from the algorithm of prediction [7].
2. **Filter methods** are based on heuristic measures, such as Mutual Information or Pearson Correlation, to score features based on the prediction task's information content. Filter methods are also independent of the algorithm in use, similar to wrapper methods. They do not require many models to be trained, however, and thus scale well for large datasets. However, filter methods cannot naturally be extended to recommend systems where the prediction target varies and depends on the history of the user as well as the item being considered. This work suggests a framework and algorithms to address the above-mentioned challenges [7].
3. **Embedded methods** are an algorithm family in which the selection of features takes place during the training phase. They are not based on cross-validation, unlike wrapper methods, and therefore scale with the data size. Since the selection of features is an inherent property of the algorithm, however, an embedded method is closely linked to the specific model: if the recommendation algorithm is replaced, the selection of features must be revised [7].

Techniques of Content Based Approach

NAÏVE BAYES: Naïve Bayes is an inductive learning probabilistic approach that belongs to the general Bayesian class. These approaches generate a probabilistic model based on previously observed data. The model estimates that a posteriori probability, P(c), is document d belonging to class c. This estimation is based on the a priori probability, $P(c)$, the probability of observing a document in class c, $P(d|c)$, the probability of observing the document d given c, and $P(d)$, the probability of observing the instance d. Using these probabilities, the Bayes theorem is applied to calculate $P(c|d)$[3][20]:

$$P(c|d) = \frac{P(c)P(d|c)}{\mathrm{P(d)}} \qquad 2$$

Merits and Demerits of Content Based approach

The approval of the content-based recommendation paradigm has several advantages:

1. USER INDEPENDENCE-content-based recommendations only take advantage of the ratings provided by the active user to create their own profile. Rather, collaborative filtering methods require ratings from other users to find the active user's "nearest neighbours"[5].
2. TRANSPARENCY — Explanations of how the recommendation system works can be provided by explicitly listing the content features or descriptions that caused an item to appear in the recommendation list. These features are indicators to be consulted to determine if a recommendation is to be trusted [5].
3. NEW ITEM — Recommenders based on content can recommend items that have not yet been rated by any user. As a result, they do not suffer from the first-rate issue, which affects collaborative recommendations that rely solely on the preferences of users to make recommendations. Therefore, the system would not be able to recommend it until the new item is rated by a substantial number of users [5].

Content-based systems have several shortcomings:

1. LIMITED CONTENT ANALYSIS — Content-based techniques have a natural limit on the number and type of features associated with the objects they recommend, either automatically or manually.
2. OVER-SPECIALIZATION - Recommenders based on content have no method inherent in finding something unexpected. The system suggests items with high scores when matched against the user profile; therefore, items similar to those already rated will be recommended to the user. This drawback is also referred to as the problem of serendipity to highlight the tendency of content-based systems to produce recommendations with limited novelty.
3. NEW USER - Before a content-based recommendation system can really understand user preferences and provide accurate recommendations, sufficient ratings must be collected. Therefore, the system will not be able to provide reliable recommendations when few ratings are available as for a new user [5].

IV. ASSOCIATION RULE LEARNING

Association rule learning is a rule-based machine learning method to discover interesting relationships in large databases between variables. It aims to identify strong rules found in databases using some interesting measures.[1] This rule-based approach also creates new rules as it analyses more data. The ultimate goal, assuming a sufficiently large dataset, is to help a machine mimic the functional extraction and abstract association capabilities of the human brain from new uncategorized data.

Apriori Algorithm: Apriori is a transactional database learning algorithm for frequent itemset mining and association rule. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger sets of items as long as those sets of items appear in the database often enough.

We can build a movie recommendation engine using user-based filtering hybrid with Apriori. For example, suppose there is a dataset having movie ratings which contain userid, movieid, ratings where userid is unique to each user, movieid is id(unique to each movie) of the movie being watched and rated by the user. Steps that can be taken to build a good recommendation engine involving hybrid approach of user-based filtering and Apriori are: -

1. First, take a simple logical step which is from our intuition that users rating a particular movie with same ratings must have similar taste in watching movies. This step is basically where User-based collaborative filtering is used.
2. Follow the above intuitive approach by first combining the users into a particular set according to their movieid and rating. All the users that rated a particular movieid with same rating will go into one set.this can be done by just making a dictionary of key, value pair where key can be tuple of (particular movie, particular rating) and value will be a set containing all users rating a particular movie with particular ratings. and then iterating through entire movie rating dataset analyzing each row of data and mapping each user to dictionary according to (movieid, movie rating) tuple.
3. After second step dictionary will be having key values where key value can be any combination of movieid, rating that is there will be M x R keys where M is count of movie_id ie. movies and R is count of possible ratings that is value 3(1,2,3,4,3) where 1 is the most disliked rating and 3 is the most liked rating of movie.
4. Then all the values of a dictionary should be considered as new dataset where each row will represent the value of the particular key of a dictionary.There after dictionary values are being as a new dataset, each row in the dataset will correspond to set of users who rated a particular movie with a particular rating in each row where total no of rows possible will be M x R.
5. Then treat each row as a transaction where user_ids will be treated as items. Then apply apriori algorithms to find out associations rules between users that will tell us how many times a particular set of users occurred together.Then use that association results as users behaving similarly and having similar taste and then recommend movieids of other users to a particular user of each association rule and do same for every other association rules we find out after running apriori algorithm.

Issue with Apriori Approach

Sometimes, it may need to find a large number of candidate rules which can be computationally expensive. Calculating support is also expensive because it has to go through the entire database.

V. HYBRID APPROACH

Conventional recommender system approaches like collaborative filtering (CF), content-based, and knowledge-based filtering have both advantages and limitations. For example, cold start and sparsity are the limitations of collaborative filtering whereas content-based approaches suffer from narrowness and require descriptions. Using hybrid approach to make prediction we get more robust recommendation system. [1][6][9]

Types of Hybrid

Weighted Hybrid In this approach, a score for every counselled item is just the weighted total of recommendation scores for every item. Weights for every context source are user-configurable through interactive sliders. Mechanically optimizing the set of weights for every context supply is fascinating, however not trivial. Empirical bootstrapping may be used to calculate the best weight scheme, but historical knowledge is required for this approach [6].

Mixed Hybrid. In this approach, results for every source are graded, then the top-n items are picked from every supply, one recommendation at a time by alternating the sources. This approach solely considers relative position in a very stratified list and doesn't embrace individual recommendation scores. In cases wherever a recommendation is created by multiple context sources the rule merely selects future recommendation from the ranked list for that source [6].

Issue with Hybrid Approach

Reliable Integration: The primary drawback is to reflect the collaborative and content-based information once creating recommendations. a simple answer is to use collaborative and content-based strategies in parallel or in cascade. However, such an approach has drawbacks. Although meta recommender systems are planned to pick out a recommender system among standard ones on the idea of certain quality measures the disadvantages of the chosen system are inherited. Moreover, the heuristics-based integration controlled in alternative studies lacks a principled justification [3].

Efficient Calculation: Reliable Integration: the primary drawback is to reflect the collaborative and content-based information once creating recommendations. a simple answer is to use collaborative and content-based strategies in parallel or in cascade. However, such an approach has drawbacks. although Meta recommender systems are planned to pick out a recommender system among standard ones on the idea of certain quality measures the disadvantages of the chosen system are inherited. Moreover, the heuristics-based integration controlled in alternative studies lacks a principled justification [3][9].

VI. CONCLUSION

Several recommendation systems support collaborative filtering, content primarily based filtering and hybrid recommendation strategies and so far most of them are ready to resolve the issues whereas providing improved recommendations. However, because of data explosion, it's needed to figure on this analysis space to explore supply new strategies that may provide recommendation in an exceedingly big selection of applications while considering the standard and privacy aspects. Thus, the current recommendation system wants enhancement for present and future necessities of better recommendation qualities.

REFERENCES

[1] Alexandrin Popescu and Lyle H. Ungar, David M. Pennock and Steve Lawrence," Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments", POPESCUL ET Al,2001

[2] Greg Linden, Brent Smith, and Jeremy York,"Amazon.com Recommendations Item-to-Item Collaborative Filtering", IEEE Computer Society 2003.

[3] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno," An Efficient Hybrid Music Recommender System Using an Incrementally TrainableProbabilistic Generative Model",IEEE 2008

[4] Zunping Cheng, Neil Hurley," Effective Diverse and Obfuscated Attacks on Model-based Recommender Systems" 2009 ACM.

[5] Pasquale Lops, Marco de Gemmis and Giovanni Semeraro," Content-based Recommender Systems: State of the Art and Trends", Springer 203

[6] Svetlin Bostandjiev, John O"Donovan, Tobias Höllerer," TasteWeights:A Visual Interactive Hybrid Recommender System" 2012 ACM

[7] Royi Ronen, Noam Koenigstein, Elad Ziklik and Nir Nice," Selecting Content-Based Features for Collaborative Filtering Recommenders", ACM 206

[8] Hendrik Drachsler, Hans G.K. Hummeland Rob Koper, "Personal recommender systems for learners in lifelong learning networks: the requirements,techniques and model".

[9] Bhumika Bhatt, Prof. Premal J Patel, Prof. Hetal Gaudani A Review Paper on Machine Learning Based Recommendation System