Explainable Recommendation System

Thesis Part-II (CS67102) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Master of Technology

in

Computer Science and Engineering

by

Onkar Telang

(18CS60R70)

Under the supervision of

Prof. Sudeshna Sarkar

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2019-20

June 5, 2020

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: June 5, 2020 (Onkar Telang)
Place: Kharagpur (18CS60R70)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Explainable Recommendation System**" submitted by **Onkar Telang** (Roll No. 18CS60R70) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Master of Technology in Computer Science and Engineering is a record of bona fide work carried out by him under my supervision and guidance during Spring Semester, 2019-20.

Date: June 5, 2020

Place: Kharagpur

Prof. Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Abstract

Name of the student: **Onkar Telang** Roll No: **18CS60R70**

Degree for which submitted: **Master of Technology**

Department: **Department of Computer Science and Engineering**

Thesis title: **Explainable Recommendation System**

Thesis supervisor: **Prof. Sudeshna Sarkar**

Month and year of thesis submission: **June 5, 2020**

Explainable Recommendation system states the reason as to why an product is recommended to user. The explanation is based on decision rule of tree construction and latent factor is learnt as a function of these rule. The thesis work address the problem of learning effective latent factor of group of similar user that share the similar characteristics and preferences. This involves improving the performance of model by adding global preference information of relative importance among features of product and predicate selection strategy for regression tree construction. The recommendations are explained using performance of those features and multiple features based template sentences is used for explanation.

Acknowledgements

I express my sincere gratitude to Dr. Sudeshna Sarkar, Professor, Computer Science & Engineering Department, Indian Institute of Technology, Kharagpur for his supervision and guidance during the project. Her valuable advices, encouragement, suggestions and friendly attitude towards me, during this work helped a long way in bringing this project report to this stage. I am also thankful to her for extending all kind of help and for providing the necessary facilities required during this work.

I am very much thankful to Prof. Dipanwita Roy Chowdhury, Head, Department of Computer Science & Engineering Department, IIT Kharagpur for providing necessary facilities during the project work.

I am thankful to Faculty Advisor Asst. Prof. Pralay Mitra and all the faculty members of the department for their valuable suggestions, which helped me improve on this work.

I would like to thank my family for all their love and encouragement. I would like to thank my parents, whose love and guidance are with me in whatever I pursue.

Onkar Telang

Contents

List of Figures

List of Tables

Abbreviations

RS	**R**ecommendation **S**ystem
ERS	**E**xplainable **R**ecommendation **S**ystem

Chapter 1

Introduction

With the surge in the E-Commerce, Search Engine, Online Social media and Video streaming Services that millions of people use on the daily basis, recommender system see through massive contents and make decision related to personalized recommendation. The impact of recommender system is profoundly increasing day-by-day. There has been growing need to understand user's interest and build the system that user trust so that right content would be deliver in the timely manner. Explanations about why the items are recommended, bridges the trust gap between users and recommender system. High quality explanation increases user's reliability and confidence in the system which saves the time, help them make better decision and persuade users to try items.Thus, the goal of recommender system is to get user's response and quickly find useful item for them.

There are lots of online services that employs recommender systems and provides recommendation on most relevant items. With such recommendations, it gets inconvenient for user to understand the reason behind the recommended item. The user need to spend additional resources in form of time or money for identifying whether the recommended items match to user's preferences. There have been case that the user give up the recommended item due to lack of clarity guidance.

Hence explaining the reason has been regarded as important task in recommendation system to thrive in the businesses. It is assumed that explanation guide user's buying pattern and affects there preferences.

FIGURE 1.1: Explainable RS Example: Set of distinguished features associated with item that corresponds to User Preferences. source:[2]

Some of the earlier work with recommendation systems focused on content-based recommendation, collaborative filtering or hybrid approach. The Content based recommendation system suggest items based on user's preferred item in the past while Collaborative approach maps interest of similar person. The techniques used for content based RS are Information retrieval, Clustering, Bayesian Classifier etc. and Probabilistic model for Collaborative filtering. There are studies that suggest different parameters for RS required to deliver relevant recommendations

- **Coverage:** Range of Items that recommender system able to recommend.
- **Relevance:** Deliver the highly relevant recommendation in the timely manner.
- **Scalability:** Work on large scale data items, required to scale up with growing dataset and resource consumption.
- **Reliability:** Essential building block for successful recommender system.

while these parameters required to deliver 'Relevant' Recommendations.Yet, there are few parameters of 'Effectiveness', 'Explainability','User Satisfaction' and 'Transparency' lacked in previous studies.

An explanation serves as an information and reason behind recommendation displayed to users. There are different explanation style such as rule based, radar chart, relevant user/item and images etc. Each style used for different purpose whose effectiveness vary with the type of service where recommendation system employed. The source of information required to recommend an item to user too vary with the nature of service. However, they are generally categorized into implicit and explicit data sources. The implicit data source includes frequency of item views, click rate, bookmarks etc. These sources are easier to collect and available in large quantities.The other type of source requires feed-backs from users. Usually, they are side information provided by the user such as rating, images and reviews etc.

1.1 Cold Start

Cold start problem occurs when there are insufficient data available to recommend an item. Thus, system's predictive ability become limited. The system fails build user's personalized profile as there preferences are yet to be understood. The cold start problem arises in two different cases i.e When new user or item enters the system. Hence, Cold start become one of major problems in recommendation system domain.

Chapter 2

Literature Survey

Over the span of last two decade, there has been abundant research articles published with regard to recommender system. The research work employed various techniques pertaining to Content based filtering, Collaborative or Hybrid based recommendation. One of the earlier technique used to model the user preferences based on TF-IDF weighting scheme. Previous studies of a decade long shown that successful recommendation systems depends on the Collaborative Filtering technique, that utilizes asked fundamental question as "how a system should explain those recommendations to its users". and these question has gained lot of attention during recent times. Whenever there lacks the transparency, user may be put around in dilemma of assessing the recommendation quality by taking action over the recommendation, e.g., best items in the market; Yet in order for users to accept the personalized results, It is required to first build confidence in the system by providing explanation as a piece of additional information behind recommendation. Studies have shown that it improves trust, persuasiveness and help them make quick decision[3].

2.1 Display Styles

It is possible to explain the same recommendation with many different explanation. These explanations can be displayed in different styles with different information sources that could drive the users utilization and behaviour towards item. There are studies that suggest the goal of explanation(Tintarev 2007)[9]

- **Satisfaction:** Ease of Usability
- **Trust:** Improve users confidence in RS
- **Scrutability:** Allow users to give feedback to system
- **Transparency:** Explains how system work
- **Efficiency:** Allow to make quick decision
- **Effectiveness:** Ease of decision making
- **Persuasiveness:** Convince to buy/try an item

(Tintarev 2015)[10] shown different display styles as images, natural language text, relevant item/user, template based features etc.

(Herlocker et al.2004)[4] proposed 21 different explanation ways for a CF based recommendation system. (Bilgic and Mooney 2005) studied keyword-style, neighbor-style and influence-style in a content-based book recommendation system. (Sharma and Cosley 2013)[7] conducted users studies to investigate the effect of social explanations, e.g. "X, Y and 2 other friends like this.". However, these studies focus mostly on content-based CF algorithms, whose recommendation mechanism are easier to interpret yet recommendation accuracy is known to be inferior as compared to modern latent factor models. The study from (Herlock et al.2004) concluded that a histogram style displaying the ratings from similar users was the most persuasive among other explanation interface.

2.2 Latent Factors Models

With the introduction of Matrix Factorization as an effective method during 2009's Netflix prize competition, The system can learn item and user representation in the form of Latent Dimension Using SVD Decomposition(Koren et al,2009)[5], NMF. This method gained wide popularity during late 2000s for addressing the recommendation relevance and accuracy.

Matrix Factorization and Tensor Factorization are known to be Latent Factor Model. They have traditionally achieved great success in large scale industry based recommender systems. These type of algorithms map users and recommendation candidates to a lower dimensional latent space. They encodes user and item affinities among different entities. Although they produce high quality results , the latent and nonlinear characteristics of this family of solutions makes it extremely difficult to explain the generated recommendations. There effectiveness and lack of interpretability among these algorithms has attracted increasing attention to explore the possibilities.

Abdollahi and Nasraoui[1] introduced explainability as a constraint in factorization: the learnt latent factors for a user should be close to those learnt factors for the items positively rated by the user. However, such type of algorithms only explain ratings, either the overall rating or feature-level ratings, while ignore the details in a user's comment. They are restricted to extremely generic explanations, such as "You might be interested in [feature], on which this product performs well".

2.3 Deep Learning

Attention mechanism has been shown eective in various machine learning tasks such as image/video captioning and machine translation. The key idea of soft attention is to learn to assign attentive weights (normalized by sum to 1) fora set of features:

higher (lower) weights indicate that the corresponding features are informative (less informative) for the end task. In the

eld of recommendation systems, He et al. introduce an attention mechanism in CF which consists of both component-level and item level attention module for multi-media recommendation.(Chong Chen et al., 2018) proposed a Neural Attentional Regression model with Review-level Explanations (NARRE) that learnt the usefulness of each review by utilizing the attention mechanism that automatically assigns weights to reviews when modeling users and items.

Chapter 3

Dataset and other details

3.1 Data Description

Publicly available product information of 'Amazon Cell Phone'[1] and 'Yelp Restaurant Business'[2] category has been used for the thesis work.

'Amazon Cell Phone' product information

- ASIN (Amazon Product Unique Id)
- Brand of product
- Title of product
- URL of product
- URL of product Image
- Price of product in Dollar
- Unique Amazon User Id
- Name of profile
- URL of reviews
- Number of up-votes for reviews

[1] Amazon data : `http://jmcauley.ucsd.edu/data/amazon/`
[2] Yelp data : `https://www.yelp.com/dataset`

Data Set	No of Users	No of Reviews	No of Items
Amazon	10561	2,242,624	13428
Yelp	10719	7,325,236	10239

Table 3.1: User and Reviews Counts

- Rating of product given by user
- Unix timeStamp
- Summary of reviews
- Text of review

'Yelp Restaurant' Business information

- Unique Business Id
- Name of Business
- Full address of restaurant business
- City of restaurant
- State of business
- Unique Id of review
- Unique User Id
- Username of Business
- Review upvotes
- Text of Reviews

3.2 Data Set Preparation

3.2.0.1 Data PreProcessing and Feature Extraction

For each review corresponding to every user, We process and clean the reviews using NLTK library. Many Core Natural Language processing task can be performed using NLTK library. Each review express degree of satisfaction with item purchased/subscribed by the user. We use the information contain in the review for understanding user preferences. One of key success of good Recommendation system attributed to modeling user preferences. User talks about certain feature liked/disliked by them. Hence, extracting sentiment and feature gives the system more insight into user's overall interest.

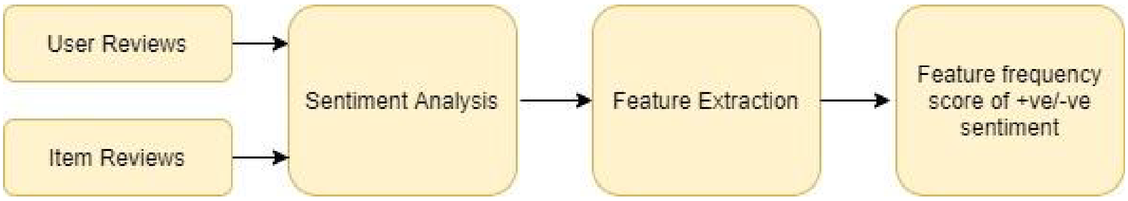

FIGURE 3.1: Steps of Preprocessing

NLTK tools splits the a review text into sentences. The tokenizer is being used to further split sentence into words. Next, all those sentences are tagged using Stanford CoreNLP library. Later, Relation detection mechanism is being searched to extract the potential features from the sentences.

3.2.0.2 Feature Indexing and Opinion Matrix

We index all the extracted features from the dataset. Using all the sentiment score given by a User for all Feature and all the sentiment score received by all the User to a Item, We create an User Opinion U_o and Item Opinion I_o matrix of dim: $U * T_f$ and dim: $I * T_f$ respectively. Where T_f is the total number of extracted features. We put sentiment score for Item opinion matrix and frequency score for User opinion matrix. F_{ij} is frequency of user i mentioning feature. f_l feature in all user i reviews. ${p_{ij}}^u$ is positive sentiment polarity of all user all generated reviews about item j. similarly n_{ij} is the negative sentiment polarity. This capture relative emphasis given to features by user i.

$$
\begin{aligned}
F_{ij}^{u} &= \Phi, \qquad\qquad if \quad p_{ij}^{u} = n_{ij}^{u} = 0 \\
&= p_{ij}^{u} + n_{ij}^{u}, \qquad\qquad Otherwise
\end{aligned}
\tag{3.1}
$$

The Data-Sets of Rating matrix and Opinion Matrices are splited into 90% of training and 10% testing data.
Number of feature extracted for Amazon Data = 101
Number of feature extracted for Yelp Data = 104

Chapter 4

Collaborative Filtering using Regression Tree

The work in this thesis address the problem of learning effective latent factor of group of similar user that share the similar characteristics and preferences. This involves improving the performance of model by adding global preference information of relative importance among features of product and predicate selection strategy for regression tree construction. The recommendations are explained using performance of those features in an item over complete dataset and multiple features based template sentences.

4.1 Introduction

Latent factor models provides one of the state-of-art accurate recommendation and empirical performance. These models are extremely difficult to explain there results. There are different models proposed that incorporates and approximate there structure of latent factor for explanation such as similar user/item in latent space [1] and phrase level sentiment analysis based models [11, 13]. However, due to joint learning framework of some of these models, explanations fidelity corresponds to those factors is hard to measure.

The matrix factorization technique can be expressed as

$$L(U, V, O) = \min_{U,V} \sum_{(i,j)\epsilon O} (r_{ij} - u_i^T v_j)^2 \tag{4.1}$$

Where L is an loss function and u_i and vj are user and item latent factors respectively.

Accurate explanation improves acceptance and confidence of recommendation. It enables them to make informed decision. Most of the existing explanation methods are based on nature language, neighbourhood and feature-level explanation. A study on well known movie recommendation shown that 86% of user desired convincing feature-level explanation [3]. In our thesis work, feature-level explanation has been used for explanation of recommendation.

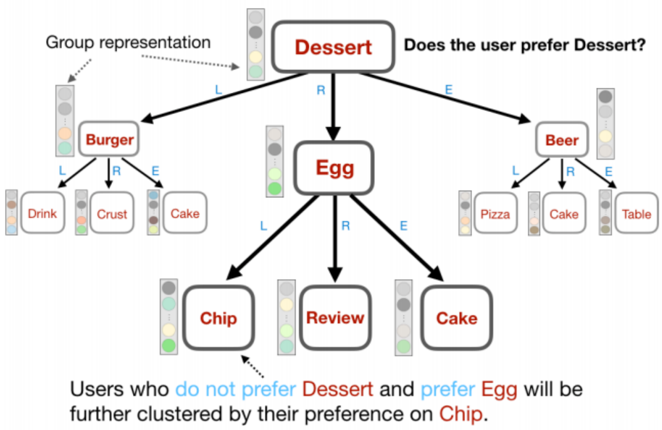

FIGURE 4.1: Explainable RS using user regression tree. latent factors of a user can found at the leaf node. source: FacT [8]

Tree based models are one of the most naturally interpretable technique used in modern applications. The rule based explanation are easily perceivable and require minimal cognitive effort to make decisions. It add transparency to the process of recommendation which fulfil one of explainable recommendation goal as described in section 2.1. We can model tree structure for both users and items in the hierarchy of features based upon some loss function over complete dataset. These explainable rule maps the user to there latent factors. Each Node represents a latent factors and group of users that share the similar preference and taste towards items. Due to group level learning of latent factors, data sparsity problem can be mitigated to some extent using third branch of tree as 'Empty'. The 'Left' and 'Right' branch of tree represents 'Prefer' and 'Decline to Prefer' respectively.

4.2 Baseline Model

4.2.1 Predicate Selection and Loss Function

User tree and Item tree are constructed using some decision rule/function as Predicate over User opinion and Item opinion matrix respectively (section 3.2.0.2).

Feature Vector: It is the column vector in either of the context of user and item opinion matrix. It represents the Frequency score corresponds to each user in User opinion matrix and Sentiment Score in Item opinion matrix.

Each feature is represented by there corresponding Feature vector in User and Item matrix respectively.

If a Predicate is initialized with the threshold t_l, then based on the value of threshold the Feature vector along with corresponding entire matrix is split into three partition of Matrix in Left, Right and Empty Branch of tree.

$$
\begin{aligned}
L(f_l, t_l) &= \{i | F_{il} >= t_l\} \\
R(f_l, t_l) &= \{i | F_{il} < t_l\} \\
E(f_l, t_l) &= \{i | F_{il} = \Phi\}
\end{aligned}
\tag{4.2}
$$

Based on the nature of partition, the model updates three corresponding partition of User Set U_o into u_L, u_R and u_E using Stochastic Gradient Descent.
The gradient is calculated using loss function of User Tree equation(4.4).

$$\begin{aligned} \Delta u_L &= (-2 \times u_L[i,j] - dot(u_L.V[j])) \times V[j]) + 2 \times \lambda_v * u_L \\ u_L &= u_L + lr \times \Delta u_L \end{aligned} \tag{4.3}$$

Where λ_v is the Item regularized term and Δu_L is updated using random element of u_L in the batch iteration.
Initially, All the users assigned with the same random vector. Similarly for all the items too assigned with the same random vector

The updated u_L from equation (4.3) is used for the optimization function in equation(4.4)

$$(\bar{f_l}, \bar{t}_l^u) = \arg \min_{f_l \epsilon F, t_l^u \epsilon T_l^u} \min_{u_L, u_R, u_E} L(u_L, V, O_L) - \lambda_b \sum_{i \epsilon L(f_l, t_l^u))} B(u_L, V, D_i^o)$$

$$+ \mathrm{L}(\mathrm{u}_R, V, O_R) - \lambda_b \textstyle\sum_{i \epsilon R(f_l, t_l^u))} B(u_R, V, D_i^o) \quad + \quad L(u_E, V, O_E) - \lambda_b \textstyle\sum_{i \epsilon E(f_l, t_l^u))} B(u_E, V, D_i^o) \quad + \lambda_u(\|u_L\|_2 + \|u_R\|_2 + \|u_E\|_2) \tag{4.4}$$

Where O_L, O_R, O_E are actual observed rating and $B(u_L,V,D_i)$ is Bayesian Personalized Ranking [6] in order to introduce pairwise ranking loss for

$D_i = \{(j,l) \; \| r_{ij} > r_{il}\}$

Similarly for the item tree, the loss function is given by

$$(\bar{f_l}, \bar{t}_l^v) = \arg \min_{f_l \epsilon F, t_l^v \epsilon T_l^v} \min_{v_L, v_R, v_E} L(U, v_L, O_L) - \lambda_b \sum_{i \epsilon L(f_l, t_l^v))} B(U, v_L, D_i^o)$$

$$+ \mathrm{L}(\mathrm{U}, \mathrm{v}_R, O_R) - \lambda_b \textstyle\sum_{i \epsilon R(f_l, t_l^v))} B(U, v_R, D_i^o) \quad + \quad L(U, v_E, O_E) - \lambda_b \textstyle\sum_{i \epsilon E(f_l, t_l^v))} B(U, v_E, D_i^o) \quad + \lambda_v(\|v_L\|_2 + \|v_R\|_2 + \|v_E\|_2) \tag{4.5}$$

The equation (4.3) and (4.4) are the objective function seeking to find the value of predicate leading to minimal split loss value. Hence the loss value is the function of nature of partition. The minimal loss occurs when a threshold is able to make partition of most similar User and Item based on frequency and sentiment score respectively.
An optimal predicate need to create the partition of input User/Item based on the value of threshold t_l leading to minimal loss. Hence, the problem of finding the optimal threshold in discrete value feature vector to split further User/Item turns into exhaustive search of threshold in all the feature vector.

4.2.2 Tree Construction

The Latent factors of User tree and Item tree are iteratively updated in an alternative fashion after constructing at intermediary maximal possible depth. From equation (4.4) and (4.5), All the item vector V is used for user side optimization and similarly vector U in item side.

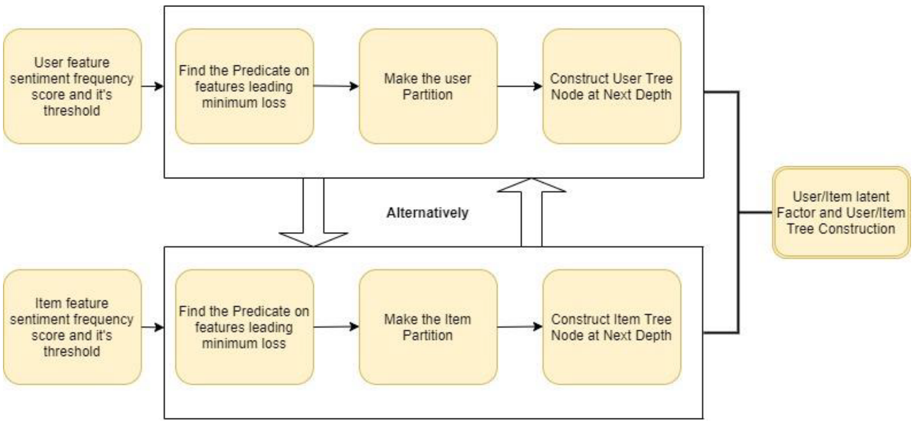

Figure 4.2: Alternative Optimization

The alternative optimization stops when we reach the maximum depth or when the consecutive updation of latent factors become extremely small. Thus, we cannot split further the group of users as they highly share there preference due to corresponding frequency scores in the User opinion matrix.

4.3 Experimentation

4.3.1 Predicate Selection and Nature of Split

The baseline model employs simple strategy of linearly finding the optimal predicate at each node. This strategy has two disadvantage
Over-fitting : Although regularized term balance out the over-fitting problem of latent factors. Selection of predicate that leads to minimum split loss value at each node end up over fitting the tree. The comparison experimentation of exhaustive linear searching versus predicate using variable length binning strategy leads to the conclusion.
Training Time : The baseline models states that due to sparsity nature of the real world dataset, the training time seems feasible. However, calculating the split loss with every potential predicate required lot of training time as split loss is calculated after gradient update.

The Plot 4.3 shows that sentiment score of value 1 occured more than 4000 times in Item opinion matrix of Amazon Data and the Plot 4.4 shows that sentiment score of value 1 occured more than 12000 times in Item opinion matrix of Yelp Data. Both the plot is approximately binomial distributed with sentiment score on both side scale from 0 value. When the baseline model run on the dataset. There are two observation on both the Amazon and Yelp data.
1) The frequency of Selected Predicate in Item tree construction approximately close to 58% of times for predicate 0, 24% of times for predicate 1, 6% for predicate 2 and 10% of times for others value.
2) The length of feature vector for item tree decreases with increase in depth as the

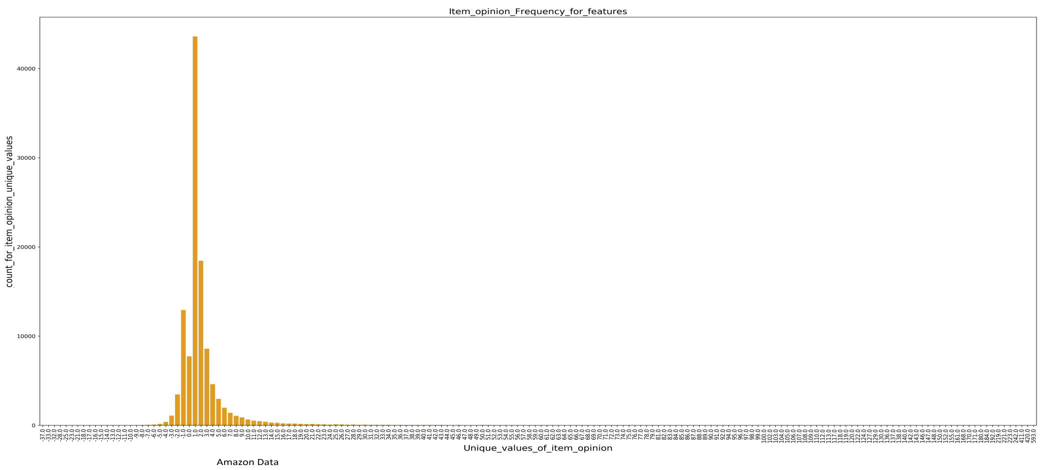

FIGURE 4.3: Count of Sentiment score vs Sentiment value of Item on Amazon Data

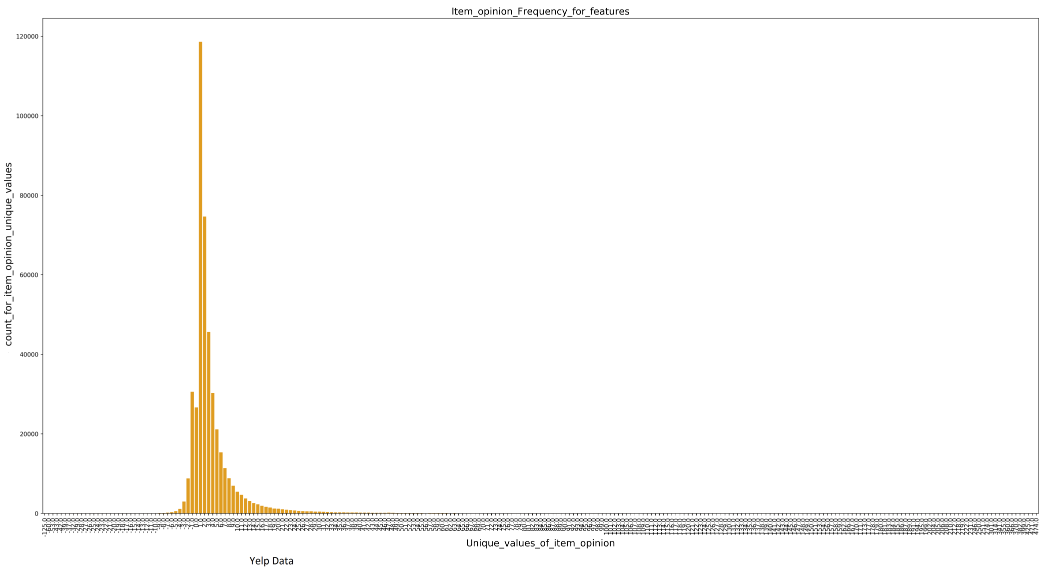

FIGURE 4.4: Count of Sentiment score vs Sentiment value of Item on Yelp Data

split in opinion matrix takes place recursively down the depth. for higher length of feature vector, the split of feature vector has always been close to equal length with selection of predicate closer to median value.

4.3.2 Experimental Setup using Non-linear Function

Based on the observations in section 4.3.1, There is a behaviour of predicate selection around neutral sentiment value. It also infers the decision area with positive and negative sentiment score for the item. The proposed method finds the optimal predicate to improve the training time of item tree construction.

We define tanh function as

$$\tanh = \frac{\exp^{x} - \exp^{-x}}{\exp^{x} + \exp^{-x}} \tag{4.6}$$

The range of tanh function belongs to [-1,1]

The strategy for selection of predicate for item tree construction are as follow

1) Apply tanh function to every element.

2) Find the mean of the vector

3) Subtract every element with the mean value

4) Find the index i of element with minimum mod(x) modulus value

5) The element at index i in feature vector is the optimal predicate.

The algorithm is able to deal with extreme sentiment score and takes O(n) time complexity. The baseline model needs to calculate the split loss with every potential element for searching the predicate. However, this method look for central value of sentiment score with given feature vector value range to make the decision. It is possible that the methods chooses sub-optimal minimal predicate which also leads to generalize the tree and latent vectors. The proposed algorithm can be applied when the length of array is greater than or equal 12.

4.3.3 Experimental Setup using PageRank Algorithm

PageRank is the one of the successful ranking algorithm on nodes of Graph. The importance score is computed for each node in the graph. It assumes that if a node connected to other important node then that node is important.

In the proposed method, PageRank Algorithm is being applied to learn the relative importance of features. Many user talks about features associated with product only fewer number of times in there reviews. fig 6.6. Large number of feature contains small frequency value in User opinion matrix. Therefore, most of the user would share the similar preference. This has shown in the analysis of baseline model as 3 is the maximum depth of user tree constructed for Amazon and Yelp data. The tree constructed using baseline model cannot be further splited because many user describe most feature in fewer number of times.

Most of the recommendation algorithm consider all the feature of equal weightage. However, some of the feature receiving high frequency score is the testament that there is general tendency of preference towards certain feature. Thus, It is crucial to consider the general importance of feature for ranking of recommendation [12]. The relative importance of feature act as global preference information.

The User opinion matrix has Dimension: I * T_f. T_f is total number of feature extracted from Data. The matrix is decomposed using BPR optimization [6, 12]. The User Feature score can be described as

$$(\bar{p}_u, \bar{q}_f) = -\sum_{u \epsilon U} \sum_{i \epsilon I^+, j \epsilon I^-} \log_e \sigma(s_{ui} - s_{uj}) + \lambda(\sum_{u \epsilon U} \|p_u\|_2 + \sum_{i \epsilon I} \|q_i\|_2 + \sum_{j \epsilon I} \|q_j\|_2) \tag{4.7}$$

The page rank algorithm need to find Co-Relation matrix between latent factor of features. The co-relation matrix (CM) is defined as

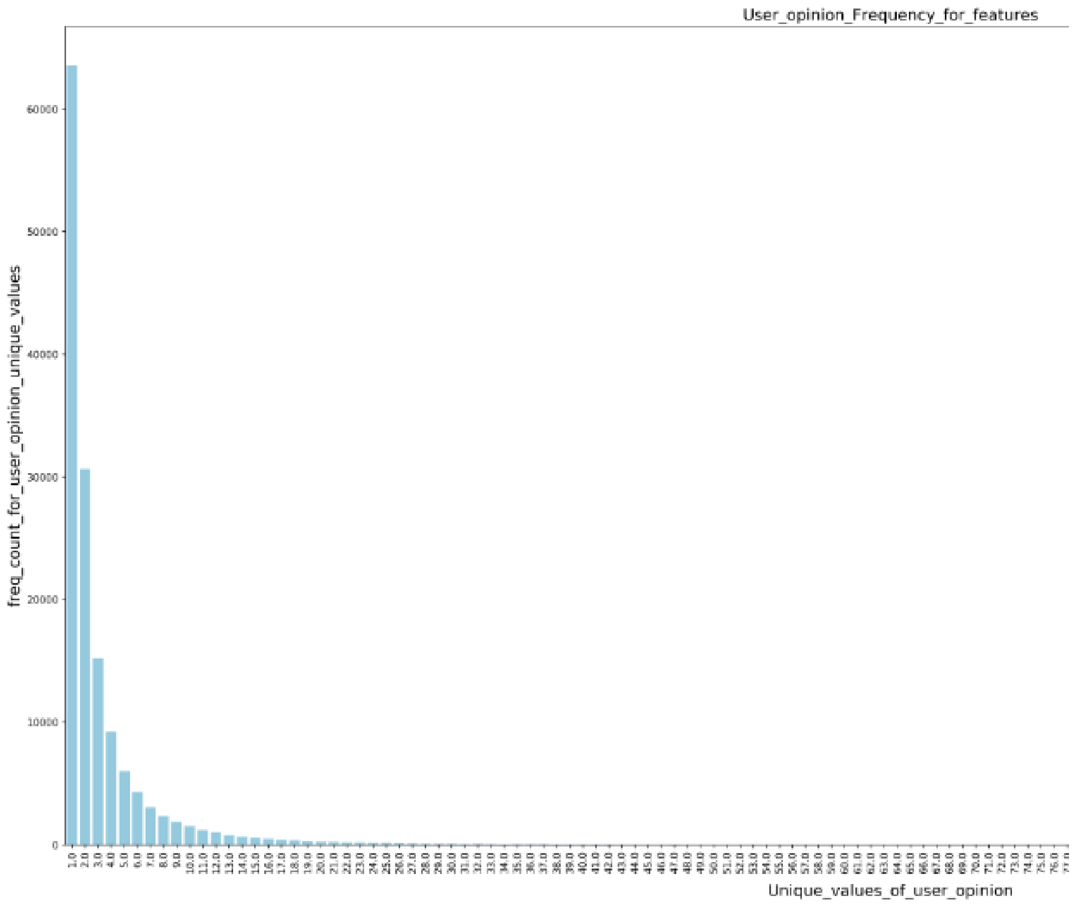

FIGURE 4.5: count of Feature frequency score mentioned by all users of Amazon Data-set. The Stretched Scale of X axis for Yelp suggest small count of many unique frequency value.

$$CM = [c_{ij}] \tag{4.8}$$

Where c_{ij} is the cosine similarity between feature latent factor i and feature latent factor j. The importance score w(i) using page rank is calculated [12]

$$w(i) = \frac{PR(i) - min(Pr)}{max(PR) - min(Pr)} \tag{4.9}$$

Networkx library is used for implementation of PageRank algorithm. Complete graph is created with nodes equals no of features. Each edge weight is the value correspond to CM matrix i^th row and j^th column.

The output of PageRank has produced probability distribution values correspond to each feature. The probability values are scaled by 100 to match the range with dataset and these values multiplied with each feature vector.

Chapter 5

Evaluation and Explainability

The performance measurement of all proposed methods for rating prediction is done using Root mean square error.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(\bar{r}_i - r_i)^2} \tag{5.1}$$

Feature Coverage for explanation = Fs/Ft

Fs = no of shared features between user i and item j from there root to leaf

Ft = Total feature from root down to user leaf node = depth of tree

In Order to explain the recommendation of an item, Locate user and item in there corresponding leaf node

To Find the Shared features, Intersect the each of the leaf's Node path's features to their corresponding root node

We can plug these multiple shared features into templates explanation

The Templates are

"We recommend this item to you because its [good/excellent][feature 1] matches with your [emphasize/taste] on [feature1], and [feature 2]..."

Chapter 6

Result

Dataset	Baseline		Non-linear modified approach		Page Rank approach	
—	Training	Testing	Training	Testing	Training	Testing
Amazon	18.659595	18.670656	19.314742	19.371001	18.534011	18.581306
Yelp	13.200128	13.259111	14.117969	14.159606	13.09592	13.217306

TABLE 6.1: Comparative Table of RMSE Values

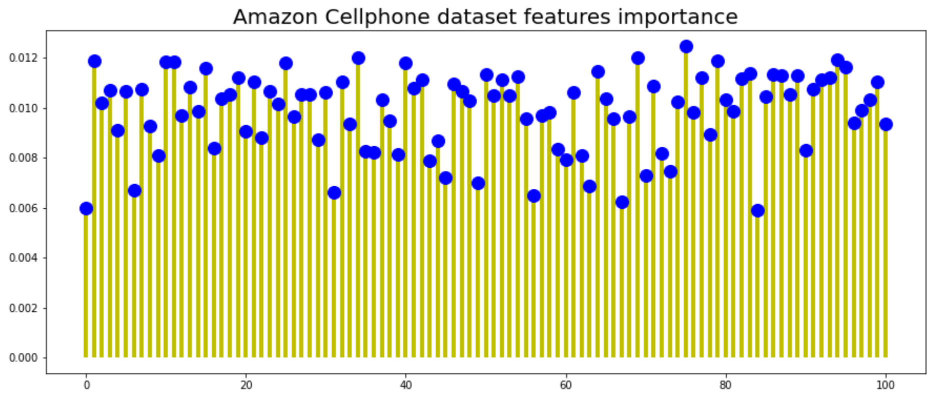

FIGURE 6.1: Probability Distribution Values produced by pageRank correspond to feature importance of features in Amazon DataSet

Explanation : In Page Rank Trees : To Find feature based explanation

Suppose We want to recommend Item j to User i

1) At 63 leaf node : User i resides of User Tree

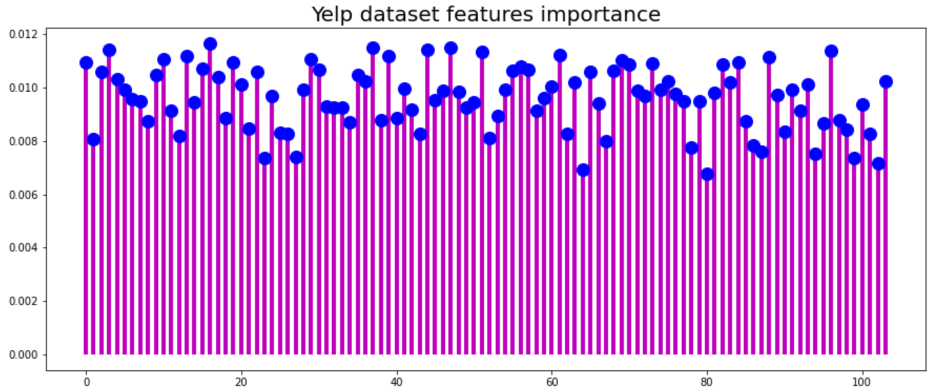

FIGURE 6.2: Probability Distribution Values produced by pageRank correspond to feature importance of features in Yelp DataSet

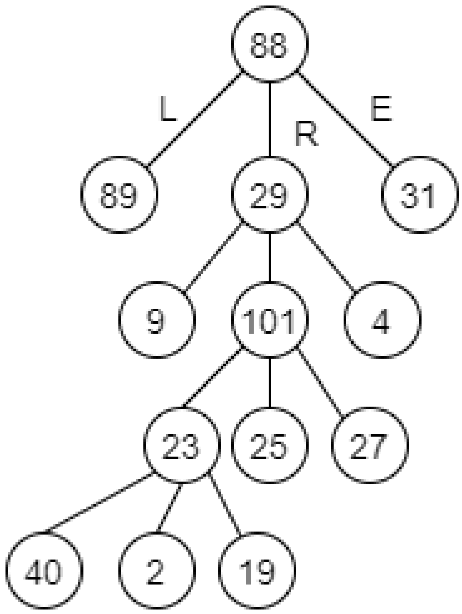

FIGURE 6.3: Using page rank: Item tree

Sr. No	Node	Split loss
1	89	695912
2	101	423990
3	9	219728
4	17	194731
5	62	10479

TABLE 6.2: Using Non-linear Approach : Item tree loss table

2) At 27 Leaf Node : Item j resides of Item Tree

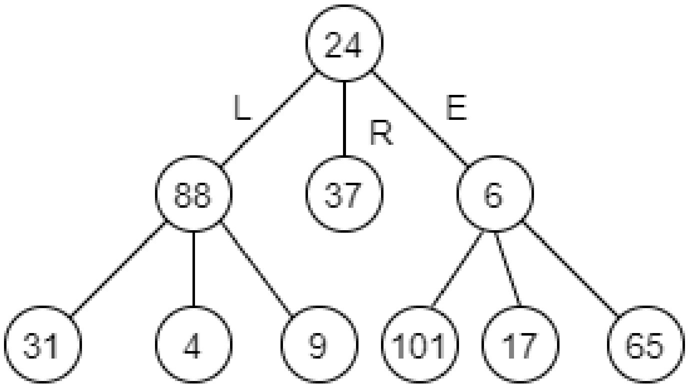

Figure 6.4: Using non-linear: User tree

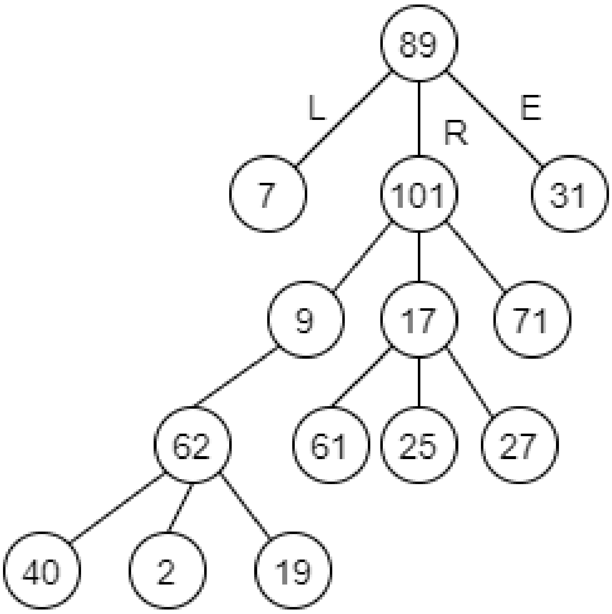

Figure 6.5: Using non-linear: Item tree

Sr. No	Node	Split loss
1	24	501431
2	88	372651
3	6	39549
4	31	17752
5	4	190571
6	17	229167

Table 6.3: Using Page Rank: User tree Loss

Path from Leaf to Root at User Tree : 63 - 31 - 88 - 24

Path from Leaf to Root at Item Tree : 27 - 101 - 29 - 88

The intersection of feature : 88 , 24, 31, 63

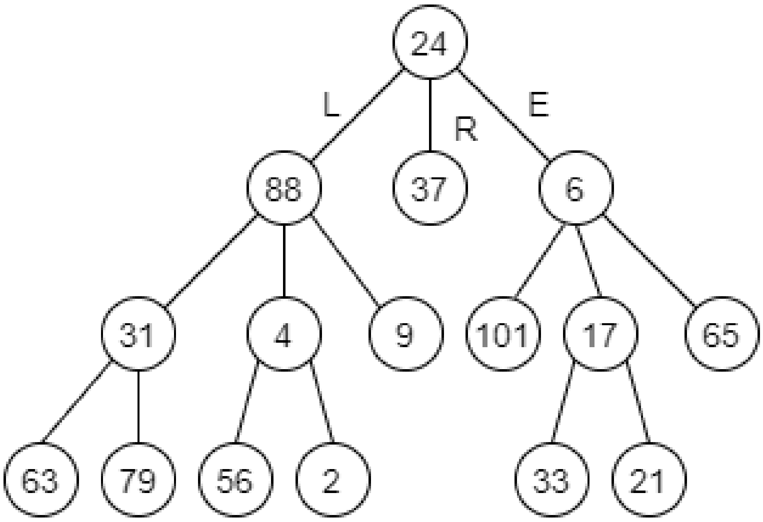

Figure 6.6: Using page rank: User tree

Sr. No	Node	Split loss
1	88	668474
2	29	363915
3	101	327937
4	23	25014

Table 6.4: Using Page Rank Loss : Item tree table

Sr. No	Node	Split loss
1	24	649264
2	88	491733
3	6	425275

Table 6.5: Using Non-linear Approach : User tree table

Name of Feature at 88 : ‘Amoled Display Screen’, ‘Samsung Phone’, ‘Android’ , ‘Good Battery’

Explanation :

“We recommend this item to you because its excellent feature ‘Amoled Display Screen’ matches with your emphasize on Samsung Phone , Android and Good Battery.

Feature Coverage = 1/4 = 0.25

Chapter 7

Conclusion

As feature vector at user side gets more diverse unique value due to feature importance score.It add more expressive power for user tree construction where large no of feature share same frequency count. This add information to further partition and make the group of user and learn more refined latent factors. Thus slighly improves the recommendation system.

Chapter 8

Future Work

By incorporating the feature importance, We can add global preference information to the latent factor model. The tree construction process is performed using predicate rule. There can be more complex predicate function that can be introduced in the tree construction process such current frequency sentiment at Item latent factor for user tree construction at each node.

Bibliography

[1] Behnoush Abdollahi and Olfa Nasraoui. Using explainability for constrained matrix factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 79–83, New York, NY, USA, 2017. Association for Computing Machinery.

[2] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 1661–1670, New York, NY, USA, 2015. Association for Computing Machinery.

[3] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, CSCW '00, page 241–250, New York, NY, USA, 2000. Association for Computing Machinery.

[4] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[5] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[6] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. *CoRR*, abs/1205.2618, 2012.

[7] Amit Sharma and Dan Cosley. Do social explanations work? studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 1133–1144, New York, NY, USA, 2013. Association for Computing Machinery.

[8] Yiyi Tao, Yiling Jia, Nan Wang, and Hongning Wang. The fact: Taming latent factor models for explainability with factorization trees. *CoRR*, abs/1906.02037, 2019.

[9] N. Tintarev and J. Masthoff. A survey of explanations in recommender systems. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 801–810, 2007.

[10] Nava Tintarev and Judith Masthoff. *Explaining Recommendations: Design and Evaluation*, pages 353–382. Springer US, Boston, MA, 2015.

[11] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. Explainable recommendation via multi-task learning in opinionated text data. *CoRR*, abs/1806.03568, 2018.

[12] H. Zhang, I. Ganchev, N. S. Nikolov, and M. O'Droma. Weighted item ranking for pairwise matrix factorization. In *2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–5, 2017.

[13] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research Development in Information Retrieval*, SIGIR '14, page 83–92, New York, NY, USA, 2014. Association for Computing Machinery.