Product Recommender System

Nitisha Pandharpurkar
Courant Institute of Mathematical Sciences, NYU
New York, NY
nvp263@nyu.edu

Nicholas Hyland
Courant Institute of Mathematical Sciences, NYU
New York, NY
nsh263@nyu.edu

***Abstract*—**

The aim of this project is to build an information filtering system that seeks to predict the rating or preference a user would give to a particular item.

Keywords—recommendations, analytics, content-based filtering, collaborative filtering, LDA

I. Introduction

For any given online retail or ecommerce service, in order to promote purchases from users, recommendations are used to predict items or products that a particular user is more likely to purchase. The goal of this study is to produce a ranked list of suggestions for any given user, which can be narrowed down to sub-categories of that particular product. Multiple algorithms can be used to derive these insights, mainly those who introduce bias due to groupings of similar users, and those based on any particular user's history. Weights can also be adapted based on feedback as a next stage in the project, to emulate an actual crowd-sourced system.

II. Motivation

The recommended products for any given user can be displayed and offered as a suggestion to the user, increasing the likelihood that a certain user will click or purchase that product, and in effect boost the profitability for those online services. If a user decides that the suggestion is incorrect, a re-evaluation of the suggestions can be made.

III. Related Work

Matrix Factorization Techniques for Recommender Systems

– Yehuda Koren, Robert Bell, Chris Volinsky

The two primary areas of collaborative filtering are neighboring methods and latent factor models. Neighboring methods being centered on computing the relationships between items or users, and the most successful latent factor models are based on matrix factorization, which characterizes both items and users by vectors of factors inferred from rating patterns. Matrix factorization employs different types of input, with one dimension representing users and the other representing the items of interest. Either explicit feedback (user input regarding their interests), or implicit feedback (user behavior, purchase history etc.) can be used. Matrix factorization models map both user and items to a joint latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space. This method has been shown to deliver accuracy that is superior to the classical nearest-neighbor techniques, and also offer a compact memory-efficient model that systems can learn easily.

RedTweet: Recommendation Engine for Reddit

– Hoang Nguyen, Rachel Richards, Chien-Chung Chan

In this paper, the authors have compared results for four approaches: linear SVM, random forests, ensemble of Naive Bayes classifiers (classical, POS naive bayes, threshold naive bayes), and WordNet distance. The tf-idf vector is used as a basis for the algorithms. The task in this system is to find genres from texts taken from Twitter, Reddit, internet posts and the Brown dataset. They found that the results are comparable for all approaches, but WordNet requires a large amount of data to be accurate.

I think that our project has a similar need to find the genre from a given text. The Amazon dataset has a review field, and we would suggest books based on identified genre. WordNet distance is an interesting tool to use, because in spite of the size of WordNet, it is difficult to use WordNet information directly because there is no representation for each word and its sense. The WordNet algorithm is computationally efficient as compared to machine learning approaches. Since we are building a system for a fairly large application usage of WordNet distance in clustering is one of the less-conventional methods which we will experiment with.

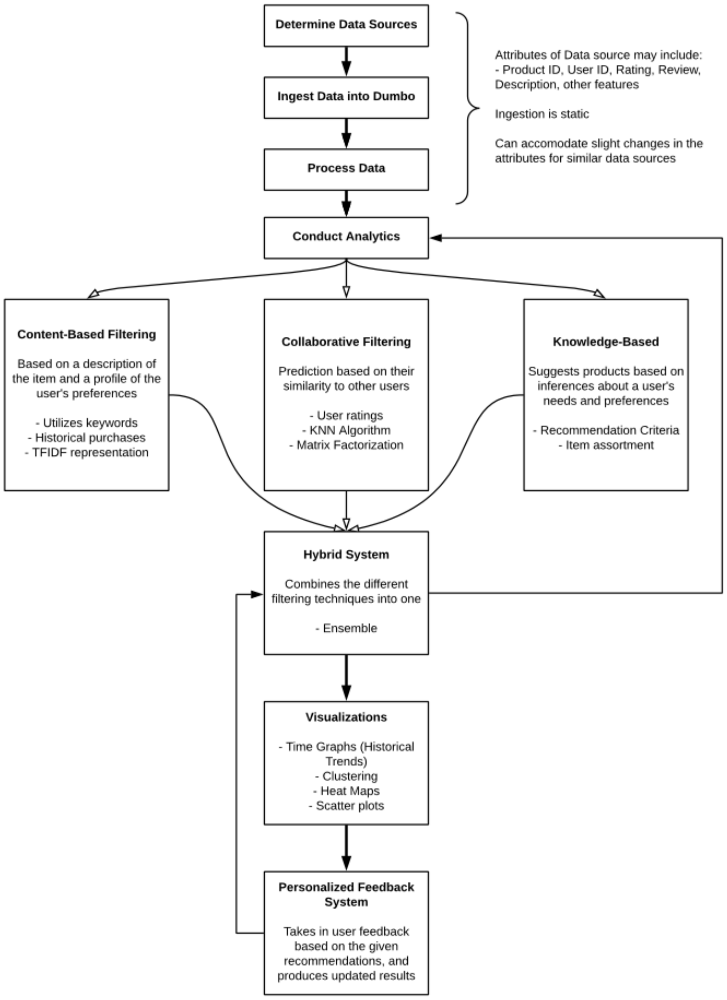
Latent Dirichlet Allocation

– David M. Blei, Andrew Y. Ng, Michael I. Jordan

This paper has shown collaborative filtering and document classification to be an application of the Latent Dirichlet Allocation algorithm. It is an unsupervised learning algorithm which assigns a certain structure to a set of documents, finds a finite set of latent topics present in the documents and assigns each topic a probability. LDA is a three-level hierarchical Bayesian model, where each topic is modeled as a Dirichlet distribution.

IV. Application Design

1) Content-Based Filtering


Determine Data Sources
Ingest Data into Dumbo
Process Data
Attributes of Data source may include:
- Product ID, User ID, Rating, Review, Description, other features
Ingestion is static
Can accomodate slight changes in the attributes for similar data sources
Conduct Analytics
Content-Based Filtering
Based on a description of the item and a profile of the user's preferences
- Utilizes keywords
- Historical purchases
- TFIDF representation
Collaborative Filtering
Prediction based on their similarity to other users
- User ratings
- KNN Algorithm
- Matrix Factorization
Knowledge-Based
Suggests products based on inferences about a user's needs and preferences
- Recommendation Criteria
- Item assortment
Hybrid System
Combines the different filtering techniques into one
- Ensemble
Visualizations
- Time Graphs (Historical Trends)
- Clustering
- Heat Maps
- Scatter plots
Personalized Feedback System
Takes in user feedback based on the given recommendations, and produces updated results

The aim of this filtering method is to classify the content into different categories or genres. In the Yelp dataset, the category attribute is present for each business, however, in the Amazon dataset, we must generate a category for each book review ourselves as each book is not classified. Content-based filtering intends to provide a recommendation for a particular user based on other similar products that they have bought before. The methods we used to classify the different categories or topics are:

a) *Latent Dirichlet Allocation:* This is an unsupervised method for finding the most relevant topics in a set of documents. In this case, the set of documents are the set of review texts of each review. The only parameter we vary is the number of categories to gather. We can derive insights on the clusters of categories by varying the number of categories.

2) Collaborative Filtering:

This method of filtering is used to make predictions on the interests of a particular user based on the preferences and history of similar users.

User-Similarity Insights and Visualization:

We used k-means clustering to cluster similar users together and scatter plots to visualize patterns and relationships in user data. We used the Spark MLLib library to classify the data, and k = 20 with cosine similarity. The vector format is [3.5 4.2 … 0.2] if a user has rated 3.5 for product 1, 4.2 for product 2, for all products frequently bought by that subset of users. We first split the users to reduce the size of the data based on geographical location, eg. All users in New York City.

3) Hash Partitions

For tables frequently used, we converted the String key to a Long key, and used hashed partitions on modulo 100 of the key. This sped up the performance significantly, taking one second per lookup as opposed to 4 minutes.

V. DATASETS

1)*Amazon Books Dataset*

This dataset contains 8.9 million complete product reviews data and metadata on books available through Amazon, spanning May 1996 – July 2014. The dataset was obtained through Julian McAuley, an Assistant Professor of the Computer Science Department at UCSD.

Size:	9.46GB
Format:	JSON
Collection:	Static (one-time collection)
Schema:	

bookID [String], helpful [array(long)], rating [double], reviewText [String], reviewTime [String], reviewerID [String], summary [String]

2)*Yelp Dataset*

This dataset contains 5.2 million reviews and 1.1 million tips by 1.3 million users for over 174 thousand businesses across the Yelp network.

Size:	6.50 GB
Format:	JSON
Collection:	Static (one-time collection)
Schema:	

Business

name [String], businessID [Integer], city [String], state [String], starRating [Integer], attributeTags [array(String)], categories [array(String)]

User

name [String], id [Integer], yelpingSince [String], reviewCount [Integer]

Review

userID [Integer], businessID [Integer], starRating [Integer], reviewText [String]

VI. Remediation

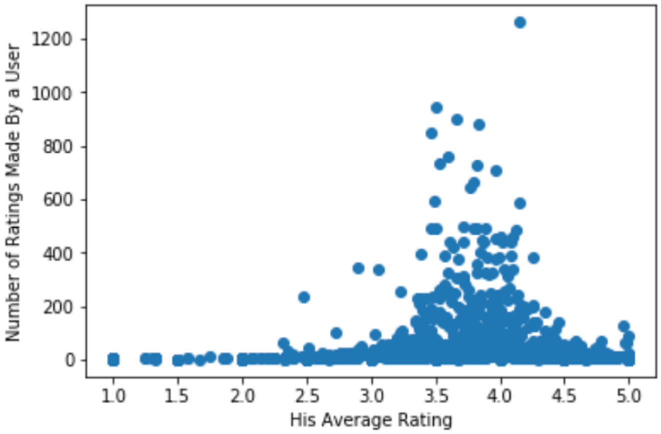
If a user decides that a particular suggestion or recommendation is incorrect, the idea is to perform a re-evaluation of the suggestions. This feedback of the system will also us to determine the effectiveness of the recommendations.

VII. Experiments

1) LDA Topic Modeling

Category	Weight
funny	0.03387741601511503
horrible	0.02695250582961795
easy	0.02667473084377978
scary	0.02641808756512150
classic	0.02245668886894862
enjoyable	0.02135043025947848
fiction	0.02064945257523609
info	0.02002435617892339
helpful	0.01926654462411966
gift	0.01876700367213290

2) User Analytics



We classified users as critics if they have more than 100 reviews.

3) K-Means Clustering

We haven’t visualized the K-means clusters because our data contains a large number of dimensions.

4) Content Based Suggestions

Example 1:

Carle Clinic Association,Urbana,IL,2.5

Suggested:

Ho Dang D MD,Chicago,IL,2.5
CU Eyecare,Chicago,IL, 3.5
5 Star Nutrition Chicago,Chicago,IL 3.5

Example 2:

Crepe Cafe,Champaign,IL,3.0

Suggested:

Zorba's Restaurant,Champaign,IL,,4.0
K-Bowl,Champaign,IL, 3.5

5) Collaborative Filtering Suggestions:

Danny with 8 reviews, yelping since -2014, average rating 3.25 was suggested similar users:
a) Hers And,4,-2016,3.25
b) Ken,16,-2016,3.19

VIII. Conclusion

This project implemented a basic framework for a recommender system on a huge dataset and included approaches for both content-based and collaborative filtering.

Acknowledgment

The authors of this paper would like to thank NYU HPC for their support by allowing us to run their application on their cluster.

References

1. McAuley, Julian. “Amazon Product Data.” Amazon Review Data N.p., n.d, Web 5 Apr. 2017.
2. Koren, Yehuda, et al. “Matrix Factorization Techniques for Recommender Systems – IEEe Journals & Magazine.” 9 Aug. 2009.
3. Nguyen, Hoang, et al. “RedTweet: Recommendation Engine for Reddit.” *SpringerLink*, Springer US, 10 May 2016.
4. M., David, et al. “Latent Dirichlet Allocation.” *Journal of Machine Learning Research*, 1 Jan. 1970.