Using Topic Modeling to Enhance Recommender Systems

Rynah Rodrigues
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14623
rcr8111@rit.edu

Abstract—There are several products in social media that are rated, reviewed and reviews on those products are marked as useful or not. A majority of such information is in an unstructured form, mainly text. All this information could be used to develop a recommendation system that would improve user experiences and financial revenue for organizations. Recommendation systems usually use ratings and usefulness of reviews to recommend products to users. However, they provide very generalized recommendations, that specific users may not be inclined to. These recommenders do not explore reviews containing text to a great extent, in providing recommendations. The idea is to develop a customized user recommendation system through user profiling. This user profiling is carried out in the form of Topic Modeling where user’s affinity towards topics are considered while making recommendations.

I. INTRODUCTION

The widespread growth in social media and the internet in general has provided a platform for the public to put forth their views, opinions and interests. Many organizations have used this aspect to their benefit in analyzing how the public responds to their products and services. Many customers provide valuable feedback in form of ratings, reviews and even providing a helpfulness score to indicate how useful a review provided by another customer was. Organizations have been accumulating a bulk of this information over the years. They use all this information to improve their products and services and to help better customer experience.

One of the most commonly used strategies companies deploy to improve the sale of their merchandise is recommender systems. This aims at showing customers what else the organization has to offer in addition to what they are looking for. What merchandise the organization decides to recommend is crucial to improving sales. If there is no thought put into this process all the recommendations could be futile and would play little to no role in improving the revenue for the organization. The recommendation systems thus aim at capturing the interest of the customers through the products they recommend in a manner that is quick, automated, can work with and manage large databases and cater to several customers. They do this by analyzing customers previous purchases and buying patterns to recommend product based on either product similarity or user similarity. Ratings are the simplest way to identify similarities and help in the prediction process. The prediction process aims at predicting a rating for a customer and a product that the customer has not bought using ratings from all users for products. If the predicted rating is high that product is recommended to the user. A drawback to this approach is that it provides a very generalized recommendations. If a user is recommended a rock music CD and the user has no interest in rock music and prefers country music instead, then even though the predicted rating is high, the recommendation is irrelevant to the user. Thus, there needs to be a way to make this approach more customized to favor user preferences. Such information is readily available in the form to reviews a customer provides. Reviews provide a lot of information as to the key aspects that interest or disinterest a user about products. Their views and opinions, likes and dislikes are all expressed in the form of reviews. This data can be used to extract valuable information about the user which ratings alone are just not able to provide. For instance, a user gives a rating of ‘2’ for a product. This would indicate that the user is not satisfied with the product. However, the review the user provides indicates that the user has no issues with the product but was very disappointed with the delivery services provided. Thus, depending only on ratings may be misleading. Thus, by incorporating review information, a qualitative aspect is added to the system which will help provide personalized customer experience, thereby improving sales for the organization. Working with reviews is however complex as it is unstructured and in its raw form. Thus, not a lot of recommender systems use this information to a great extent.

Topic Modeling is developed for text data, and helps to extract relevant topics, the documents represent. This project aims at extracting the useful information present in the reviews with the help of Topic Modeling. Users are associated with topics they have a strong affinity towards, helping recommender systems to recommend products that intersect with the user’s most preferred topics. The rest of the report provides a brief overview of the related work done in the area, a detailed documentation the methodology adopted, discussion of the results, some useful exploratory analysis performed and finally a conclusion and scope of future work that can be carried out.
II. BACKGROUND

Traditional Recommender systems make use of user ratings to make predictions. In [6], several approaches to using ratings for recommendations have been discussed. One of the approaches uses content based filtering with makes a prediction rating by mapping features of the item to user’s preferences using a decision tree. Decision trees however, are unstable and their cost logarithmic. The collaborative filtering approach used a non-linear Latent factor model. User-based KNN and Item-based KNN. Latent factor model seemed to perform better than the KNN models.

Since the ratings alone generate fairly generalized results, in the recent focused has shifted to incorporating user’s preference aspect into the models using Reviews. In [7], a very naive approach was used where while a user browses for a product, reviews of similar users, about the product were displayed to give users a sense of what other people thought about the products they were browsing. Once a user bought something, they were asked to fill out a questionnaire as to what their views were on the products they purchased. In [9], they have focused on e-commerce product that are very expensive and thus do not have users with much experience. They build a network for similar users according to their preferences for related features and edges representing similar preferences in a pairwise manner. Using a Latent class regression model they identify sub-communities where all members have provided like-minded reviews. Thus when a new user enters, they are asked about their preferences and are mapped to the correct community and recommendations are made according to the past behaviour of that community.

In [8], a very interesting context-aware recommender system has been proposed. It uses labeled LDA to get an idea of user’s preferences towards features already stated and uses a user’s past rating behaviour to make predictions about the ratings, that user would provide. However, the approach required that the data be supervised, which make it difficult to obtain. In [10], the movieLens dataset is used to make recommendations using user reviews. User’s preference features are extracted using LDA, while collecting all the reviews for a particular movie and considering as one unseen document for a global model, movie features are extracted. This is then used to make predictions based on how much the user’s preference list for features they consider important maps to that of the movie features in general.

III. METHODOLOGY

A. Dataset Selection

In order to perform effective analysis and predictions, the dataset needs to be large with sufficient amount of users, products and reviews. Amazon has a site which contains Amazon products along with reviews from several users. The data is subdivided into smaller datasets representing products belonging to a particular category. Each of the datasets contain information regarding the product, user, rating, timestamp, product review, its usefulness and summary of the review. Each of the datasets approximately contains five to six lakh records. All of this information is quite useful and will help in the recommendation process as well as in the analysis of the data.

Each of the datasets was experimented with, in order to find a dataset which had a perfect balance of user to product information, while also taking into account usefulness of the reviews provided. Each of the datasets went through a three part filtering process and the dataset with sufficient amount of records after the filtering process was selected. The process is as follows:

![Fig. 1: Filtering Process for dataset.]

Helpful reviews is the ratio of number of user who found the review useful to the total number of users who rated the review. A threshold value of 0.5 was used to make the selection. Using this filtered data, users who rated at least 3 products were selected and finally products that were rated at least 5 times formed the final dataset. The “Home and Kitchen” dataset was selected with a total of 142457 records, having 31987 users and 11145 products.

B. Data Cleaning and Processing

This section is subdivided into two parts. The first includes data cleaning and processing of features for basic recommender system models and analysis. The second includes data cleaning and processing for Topic modeling phase.

1) Data cleaning and processing for basic Recommender Systems and Analysis:

- The user and product unique identification number was alphanumeric. This kind of data makes it difficult to index records and retrieve them easily during the process of coding. Thus, both these values were converted to integer values starting from zero in Excel using the “lookup” formula.
- The dataset included two features “numerator” and “denominator” which represented the number of users who found the review useful and the total number of user who rated the review respectively. During the filtering process in dataset selection, only the useful reviews were retained, derived from these two features. Thus these features are redundant since all reviews in the dataset are useful, and were thus eliminated.
- The user’s name is another feature which was eliminated as it does not provide any impactful information since users can be uniquely identified by their user ID.
• The date feature required formatting as it was not recognized as a date attribute in Excel as well as Python. Formatting was performed in Excel itself.
• For building basic recommender system models, a separate dataset was created from the existing cleaned dataset which included User and Product identification number and their corresponding rating provided by a particular user for a specific product.

2) Data cleaning and processing for Topic modeling:
• The cleaned dataset in the previous step was used to perform additional cleaning and processing. This section focused on two of the features in the dataset which were review and summary, as they both include text data. The processing of this data was performed by loading the data in Python.
• Performed text cleaning on the data by removing stop words and punctuations as it would affect the topics that will be generated in the topic modeling phase. This was done using “NLTK” package for stop word removal and “string” package for punctuation removal.
• Word lemmatization was used to eliminate inflectional word endings and retain only the dictionary form of words referred to as lemma. This was also implemented using the “NLTK” package in Python library.
• Additionally, code was implemented in Python to extract the top most frequent words used in the reviews and summary after performing the above mentioned text processing techniques. The list generated was used to manually eliminate useful words that is words that could have an impact on the topics generated. Thus the list contained redundant words such as love, would, could, great etc which were used for further cleaning of the text data.

C. Basic Recommender System Models

The purpose of this project is to analyse how well text data present in the form of reviews helps in improving recommendation models. In order to do this, it is important to first see how this dataset responds to basic Recommender systems in general. This helps provide a baseline for the purpose of comparison and analysis. The idea here is to build a recommendation system model using only the ratings provided by a user to a specific product. Then the model is tested to see how well it performs in predicting user’s ratings for other products currently not rated by the specific user. If the predicted rating for the user is high for a product, then that product should be recommended to the user.

1) Model Selection: There are two very common approaches to building recommender systems, collaborative filtering and content-based filtering. The latter uses metadata to build profiles of the users or products and makes associations based on the common threads between them for predictions. This approach however requires additional information about users and products to build their profiles which the dataset being used does not support.

Collaborative filtering on the other hand takes into account similarity between between users or products based on past experiences to create a new user-product association. There are two types of collaborative filtering algorithms, they are as follows:

• User-based collaborative filtering: This model first identifies users that are similar to each other based on the ratings they have provided for products bought in the past. Then for a specific product highly rated by a user ‘A’ and user ‘B’ being similar to user ‘A’ as identified by the model and not having rated this product, this product will be recommended to users ‘B’.
• Item-based collaborative filtering: This model first identifies products that are similar to each other based on the ratings given to them by users. Then for a product not yet rated by the user, identify all similar products and predict the rating based on what the user has rated those similar products. The products with a high predicted rating count will be recommended to the user.

Both of the above mentioned Collaborative filtering algorithms work very well with minimal in amount of data and is efficient. These algorithms however suffer from the issue of cold start when predicting for a new product using user-based and for a new user using item-based collaborative filtering. In both cases there isn’t enough rating information to help in the prediction making process.

Besides the two most commonly used approaches to building Recommender systems, Matrix Factorization is also an effective approach. It uses latent features to interpret interactions between users and products, and a high interaction between the two is used in making a prediction. Given the users and their ratings for products a matrix can be created where the rows represent users and the columns products. The matrix is filled with either the rating a user has given a product or nothing in case the user has not rated the product. The idea is to predict the missing values within the matrix [1]. The dot product of the vectors representing features associated with the item and users provides the overall interest of the user with a particular product [2].

2) Testing the Models: Once the models have been built it is imperative to evaluate how the model performs on the test data that is the data that the model has not previously seen. This provides an intuition of how the model will behave in the real world. It also helps in making a comparison between different approaches and make a decision as to which approach is best suited for the application. This evaluation is performed by measuring the error present in the predicted rating compared to the
actual rating a user provided for the product present in the test data. The error measures used to evaluate the model are as follows:

- **Root mean squared error (RMSE)**: This measures the deviation of the predicted result from the actual value for each of the predicted instances. It then averages all the deviations which is the RMSE for the model [3]. The formula is presented below [3].

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left( \frac{x_i - x_{true}}{x_{true}} \right)^2}
\]  

- **Precision**: The precision in this context means the number of recommended items that are good items. The measure of goodness is determined by the rating an item receives. In this case if an item has a rating of four or five, it is considered good enough to recommend. Thus for precision, all items that have a predicted rating of four or five are termed as ‘recommended items (RI)’. Then from the test set, collect all those items having a rating of four or five which are the true relevant items and term them as ‘good items (GI)’. Then count how many recommend items are present in the set of the good items divide by recommend items to obtain precision percent. Precision is given by:

\[
Precision = \frac{RI \text{ in } GI}{RI} \times 100
\]

- **Recall**: Recall on the other hand means the percentage of good items that were recommended. In this case we check how many of the good items are present in the recommended set and divide it by the total number of good items (true relevant items).

\[
Recall = \frac{RI \text{ in } GI}{GI} \times 100
\]

**D. Topic Modeling**

The dataset has two features summary and customer review both present in text format which were used for Topic modeling. Topic modeling on reviews helps uncover common aspects that concern reviewers and which they discuss frequently. The topics generated help organize the reviews and associate them with a topic. All the reviews from a particular topic, indicate that the users within that group share similar concerns and thoughts about the products they are reviewing. Topic modeling also helps grouping similar products together, based on the key features and specifications which highlight these products. This categorizes products and associates them with topics. This is particularly useful when there is not much information about products other than the product identification number. This usually occurs when the data is given to a third party for the purpose of analysis or for surveys, and user and product information needs to be kept confidential.

The are several approaches to generating topic models. The 2 approaches used for this project are Latent Semantic Indexing (LSI) and Latent Dirichlet Allocation (LDA). The most popular is LDA. Below is a brief description of the working of the two approaches.. In the case of this project, a document represents an individual review :

- **Latent Semantic Indexing (LSI)** [4]: In this approach words present in a document are considered in association with other words present in the document. The idea behind this is that for a topic there are certain set of words that are most frequently used to represent the topic. This model aims at finding those relationships among the words for a document. There are three types of relations among words same or similar word meanings (car, automobile), type of something (automobile, vehicle) and part of a larger concept (gears, car). An interesting fact is that Google has a patent over these relationship types. However this technique leaves room for ambiguity. Sometimes a same word can have very different meanings depending on the context in which it is used. For instance, ‘being fair’ can have two different interpretations as to either being light skinned or being just and reasonable.

- **This is an extension of the above technique of topic modeling. It uses a statistical approach which includes complex probability. Every word in the document is like an attribute and are grouped together to form a topic. A document can encompass several topics. The model then considers pairs of the topics to compare and to see how much of the topic is represented in the particular document and does this with all topic combinations generated for the document. Thus being able to retrieve all relevant documents associated with a topic. This helps eliminate the limitation of ambiguity found in LSI.

Once the topics have been generated for all the summary data and reviews, an association of the topic with products and users need to be created. There are two approaches as to how this can be handled:

- **Approach 1**: This is sort of a naive approach. Here, a direct association of each review is made with a topic that best represents that topic. Thus, since a user may provide several reviews, the user gets associated with all of those topics. Products too, have several reviews about them which get associated with many topics. This however fails to provide the best representation of a user or a product with a topic. For instance, if a user ‘X’ provided a review which get associated with topic ‘A’. This is the only review of user ‘X’ which got associated with topic ‘A’. Thus the user will get associated with this topic, even though there was only one review that the user provided which got associated with this topic. This makes it a very weak association and does not strongly represent the user’s affinity towards this topic. Similarly, a product if associated with topics better highlighting its features makes a stronger association.

- **Approach 2**: This approach attempts at overcoming the drawbacks mentioned in the approach above. Thus first the topics generated were refined topics and were anal-
used to see what their high frequency words included. Topics related to packaging, delivery etc were eliminated as they do not help categorize products well. For instance two users who raised concerns regarding packaging could be reviewing two completely different product and thus associating these products together would not make much sense. Thus all such topics were eliminated. Some of the topics were merged together as well to increase the number of instances in those topics whenever possible. Below is table of the first 6 topics, with the top 4 most probable words are as follows:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Probable Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>['blender', 'cube', 'smoothie', 'blend']</td>
</tr>
<tr>
<td>1</td>
<td>['juice', 'juicer', 'pulp', 'fruit']</td>
</tr>
<tr>
<td>2</td>
<td>['cake', 'baking', 'pie', 'cookie']</td>
</tr>
<tr>
<td>3</td>
<td>['blade', 'peeler', 'knife', 'sharp']</td>
</tr>
<tr>
<td>4</td>
<td>['bread', 'loaf', 'pizza', 'yeast']</td>
</tr>
<tr>
<td>5</td>
<td>['coffee', 'maker', 'ground', 'brewing']</td>
</tr>
</tbody>
</table>

TABLE I: Topics and high frequency words.

From the above topics, topic 0 and 1 were merged and labeled as juicers/blenders. Topic 3 was labeled as cutters, topic 5 as coffee, while topics 2 and 4 were merged and labeled as baking/breads. Similar process was carried out for the rest of the topics as and when required.

Associating a review with a topic directly, only represents what that review conveys and has very little information within a single text to better highlight all of the concerned feature of a product. Thus by grouping all the reviews provided by users for a particular product, a lot of information about the product can be generated. Then by performing topic modeling on this group of reviews, topics with high frequency words are generated. This is then mapped to the original set of topics to find the most suitable match, thereafter creating an association of that product with a topic. For this project only the best matching topic was selected. An extension could include top two or three topics as well. Then for each user, their affinity towards a topic was determined by the number of times the user reviewed products associated with that topic. The user’s rating for the product was not considered in this case, as rating suggests the user’s satisfaction with the product and not the user’s affinity towards a topic. If the user has made an attempt to review several product belonging to a topic, it indicates that the user has an interest with the topic, weather or not the user is satisfied with the products belonging to the topic is a completely different aspect.

E. Recommender System Models with Topic Modeling

The basic recommender system models generated above made use of only the ratings provided by customers to make predictions. Though the accuracy of those models were reasonable in general, it did not take into account the user’s profile or affinity towards products being sold. For instance the basic model predicts that the user would rate a particular coffee product with a rating of ‘5’. This makes the product a good recommendation for the user. However, if the user is not a coffee person and prefers tea instead, this recommendation would be completely irrelevant to this user. The drawback of these models is that it fails to explore the raw text content which is provided in the form of user reviews and summary information. From these features a lot of information can be extracted about the users and what their preferences are. Now an association between users and the products they like can be generated but this would amount to a lot of efforts and storage space as there are approximately 31K users and 11K products to be considered. This is where topic modeling can be put to use. A few topics can encompass a wide variety of products and users only need to be associated with a topic rather than being associated with every individual product. To incorporate topic modeling information into our models, the K Nearest Neighbors (KNN) algorithm is used. This needs to be built from scratch as it needs to incorporate not only the nearest neighbours but also the topic intersections as well. These topic intersections are sets of topics that the user is interested in based on some threshold values. If a user has rated more products belonging to a topic than the threshold value (between 4-7), then the user has an affinity towards that topic. Created three different subsets involving one, three and five topics respectively. For subsets containing one and three topics in it, a threshold value was used to ensure users had rated products within the topic at least equal to the threshold value. However, for the subset containing five topics no threshold value was used. So there were topics which were rated only once or twice. For most of the models the three topic preference set were used as they truly presented the users affinity towards the topics. Only in cases where the models do not perform too well a five topic preference set is used. Five topic may not be a true representation, as users reviewing a topic only once does not represent their affinity towards that topic.

The K Nearest Neighbors algorithm which is built from scratch is first tried out without using topic modeling. This involves a similar process as the one using topic modeling as described below. However, the K nearest neighbors are selected purely on the basis of its distance from the target user. There is no incorporation of any topic intersections. This gives us a model for comparison as well as gives an intuition as to how the model that is being built from scratch performs on this dataset.

There are three different types of KNN models that were considered. They are as follows:

- User Based KNN: In this model, a user-item matrix is built where the users are represented along the rows and the items are represented along the column. Then if the user has rated an item that information will be filled in the cell, corresponding to that user row and item column. The matrix is a sparse one. Now to make a prediction of the user’s rating, for an item, all the rating pertaining to that item are isolated and considered as the target data. All the remaining ratings for other items are used as vectors, that is every row is a vector. Then using a
distance metric, the distance of all the other users from the target user is measured. Then the K nearest users whose topic preference sets intersect with the target user topic preference sets are selected. This provides a set of user who are similar to each other according to our model. Then the average rating of the corresponding similar users is obtained from the target data, which was initially isolated. In simple terms, it is the average of the ratings given by similar users for the same item the prediction is being made for the target user. For instance a prediction needs to be made for user (U6) for item (P3), the matrix is as follows:

![User-based KNN from scratch](image.png)

**Fig. 2: User-based KNN from scratch**

- **Item Based KNN**: This has a similar approach to user based KNN. The matrix however is reversed where the rows represent items while the columns represent users. To make predictions about a user’s rating for an item the users, all the user’s ratings for items are considered as the target data. Then a distance metric is used to measure distance to all items from the target item. The K nearest items to the target item that belong to the same topic or belong to topics that intersect with target user’s topic preference set is considered. This model gathers together all items that are similar to the target item and gets an average of what the target user has rated similar items from the target data set.

![Item-based KNN from scratch](image.png)

**Fig. 3: Item-based KNN from scratch**

- **User Topic Based KNN**: This is hybrid model which takes both similar users and similar products into account. The two models above used the entire dataset to make predictions. In this case the data is segregated topicwise. This ensures all similar product are grouped together. Then user based KNN is carried out on this dataset to get similar users to the target user, making this a hybrid model.

![User-topic based KNN from scratch](image.png)

**Fig. 4: User-topic based KNN from scratch**

The distance metric used to measure the distance between target vector and the other vectors was cosine similarity and euclidean distance. The figure below depicts how these distances are calculated [5].

![Distance Metric](image.png)

**Fig. 5: Distance Metric**

Cosine similarity (θ), measures the angle between the two vectors while Euclidean distance (d) is the straight line distance between the two vectors [5].

The error measures used to evaluate the model are Mean squared error (MSE) and mean absolute error (MAE). The formulae are as follows [6]:

\[
MSE = \frac{\sum (V'_{u,i} - V_{u,i})^2}{n} \tag{4}
\]

and

\[
MSE = \frac{\sum (V'_{u,i} - V_{u,i})}{n} \tag{5}
\]
Where $V_{u,i}$ is the actual rating for user $u$ and item $i$ and $V'_{u,i}$ is the predicted rating for user $u$ and item $i$.

IV. RESULTS AND DISCUSSION

A. Results of Basic Recommender System model

<table>
<thead>
<tr>
<th>Model</th>
<th>RMSE</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Based</td>
<td>1.459</td>
<td>0.606</td>
<td>0.643</td>
</tr>
<tr>
<td>Item Based</td>
<td>3.240</td>
<td>0.568</td>
<td>0.539</td>
</tr>
<tr>
<td>Matrix Factorization</td>
<td>1.460</td>
<td>0.595</td>
<td>0.610</td>
</tr>
</tbody>
</table>

TABLE II: Basic Recommender System results

The User Based Collaborative filtering model performs the best, followed by Matrix Factorization.

B. Results of Topic Modeling

1) The topics generated from the summary text using LSI model were very vague and were thus discarded.
2) LSI and LDA were both tried on the review text. Topics generated by LDA were far more distinct and those topics were used for further processing.
3) Results from Approach 1 (refer section 3.D) are as follows:

![Fig. 6: Instances per Topic](image)

From the results above, it can be seen that all the reviews favor the last topic and is not balanced.

4) Results from Approach 2 (refer section 3.4) are as follows:

![Fig. 7: Products per Topic](image)

Even though there were a lot of products for topics ‘measuring’ and ‘coffee’, many of the users have an affinity towards topics ‘cleaners’ and ‘baking/breads’

C. Results of Recommender systems with topic modeling

1) Result 1:

<table>
<thead>
<tr>
<th>Error Measure</th>
<th>Without TM</th>
<th>With TM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>61.32%</td>
<td>66.87%</td>
</tr>
<tr>
<td>MSE</td>
<td>2.6225</td>
<td>2.2845</td>
</tr>
<tr>
<td>MAE</td>
<td>1.1235</td>
<td>0.9475</td>
</tr>
</tbody>
</table>

TABLE III: User-based KNN (Cosine Similarity, three topic preference-set, $K = 320$)

2) Result 2:

<table>
<thead>
<tr>
<th>Error Measure</th>
<th>Without TM</th>
<th>With TM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>62.10%</td>
<td>64.00%</td>
</tr>
<tr>
<td>MSE</td>
<td>2.6325</td>
<td>2.4345</td>
</tr>
<tr>
<td>MAE</td>
<td>1.1235</td>
<td>1.0185</td>
</tr>
</tbody>
</table>

TABLE IV: User-based KNN (Euclidean Distance, three topic preference-set, $K = 310$)

3) Result 3: Item-based KNN (Cosine Similarity, three topic preference-set, $K = 100$)

<table>
<thead>
<tr>
<th>Error Measure</th>
<th>With TM (3-topic)</th>
<th>With TM (5-topic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>56.55%</td>
<td>57.18%</td>
</tr>
<tr>
<td>MSE</td>
<td>2.965</td>
<td>2.761</td>
</tr>
<tr>
<td>MAE</td>
<td>1.223</td>
<td>1.179</td>
</tr>
</tbody>
</table>

TABLE V: Item-based KNN (Cosine Similarity, $K = 130$)

4) Result 4:

From the results above, the balance of products to topics is far better, than those that are seen in Approach 1 where reviews are associated with topics. Topics ‘Measuring’ and ‘Coffee’ have a lot of products. Below is the distribution of users per topic.

![Fig. 8: Users per Topic](image)
Error Measure | With TM (3-topic) | with TM (5-topic) |
---|---|---|
Accuracy | 58.65% | 62.02% |
MSE | 2.8845 | 2.841 |
MAE | 1.1875 | 1.13 |

TABLE VI: Item-based KNN (Euclidean Distance, $K = 100$)

Overall Result 4 has performed better than that of Result 3. However, for 5-topic, MSE for Result 3 is better.

5) Result 5:

| Error Measure | Cosine Similarity | Euclidean Distance |
---|---|---|
Accuracy | 64.88% | 63.42% |
MSE | 2.3605 | 2.499 |
MAE | 0.9885 | 1.042 |

TABLE VII: Hybrid Model (three topic preference-set)

Cosine Similarity for Hybrid model performed much better than that of Euclidean Distance. Overall, Result 1 has the best results for accuracy MSE and MAE, followed by Cosine Similarity results for Hybrid model.

V. EXPLORATORY ANALYSIS

- Useful v/s useless reviews

![Fig. 9: Useful vs Useless reviews](image)

- This shows the growth in ratings and how the ratings of products have improved over the past few years from 2000 to 2014

![Fig. 10: Growth in Review with Ratings](image)

- From the plot we see users have reviewed coffee products the most, followed by baking/breads even though measuring had most of the products associated with it.

![Fig. 11: Reviews per Topic](image)

- For each topic, the plot above displays how content users are with each topic. Green is contented while red is discontented.

![Fig. 12: User Topic Content](image)

- This plot displays how content users are with coffee related products over the years. There the a drastic drop between 2013 to 2014.

![Fig. 13: Users content with coffee products](image)
• This is a comparative analysis of how the topics compare to each other over the years. From this we can see that there are more discontented users with cleaning products as compared to that of baking/breads.

![Baking/bread vs Cleaning user content](image1.png)

**Fig. 14:** Baking/Breads and Cleaning Products

• The plot below describes the timeline of two users. For user1 there seems to be a decline in the number of reviews they have been providing over the years. This shows that User 1 is more likely to stop buying products as compared to User 2

![Comparison of User1 and User 2 over the years](image2.png)

**Fig. 15:** User1 and User 2’s reviewing pattern over the years

• If there is a gradual decline of more than 3 reviews, across the years those users are more likely to discontinue the buying of products. The plot below shows the users who may most likely continue to buy products and those that will discontinue.

**VI. FUTURE WORK AND CONCLUSION**

The initial experiments with basic recommender models showed that the model perform reasonably on the dataset. With the incorporation of topic modeling the the best improvement achieved was of about five percent with User-based KNN. This may not seem as a considerable improvement but considering how sparse the dataset it, this improvement is decent. Thus topic modeling has helped in improving the results and has helped profile customers preferences and group similar product together. Exploratory analysis on the dataset uncovered some interesting facts. Such analysis really help organizations understand the area which need to be focused on, which products need improvement and what efforts should be taken to ensure customers continue to invest in the company.

This topic information has further scope. There are several different recommender models which could include this topic information. It will be interesting to see which of these models performs well and can improve the model even further. The cold-start issue has not been considered here, but is of importance. There are several methods to address the cold start issue which can be considered for future experiment. There are also several exploratory analysis which can be very useful and will helpful unearth useful and productive information.

**REFERENCES**