Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Using Clustered Database for Food Recommendation System

Ravinarayana A
Information Science Dept.
National Institute of Engineering,
Mysuru, India

Pooja M C
Information Science Dept.
National Institute of Engineering,
Mysuru, India

K Raghuveer
Professor and Head
Information Science Dept.
NIE, Mysuru, India

Abstract— Food recommender system is used to find food of one's choice and interest. The goal of the paper is to develop food recommendation system. Once the user inputs the food items, the recommender system looks into its ingredients and recommends food items having similar ingredients. The location of the user is taken into consideration in order to suggest near-by restaurants. K-means clustering algorithm and similarities measures are used in order to cluster the food items. To find the matching keywords, we use a hybrid similarity measure. Document similarity is found out using a document similarity technique.

Keywords—Food recommendation system; content based technique; hybrid similarity measure;recommender systems

I. INTRODUCTION

Machine learning, Artificial Intelligence and Data science are the most happening branch of Computer science research field. Recommendation systems comes under the field of machine learning. Recommendation systems are most widely used nowadays, they have changed the way people find product and information. Nowadays, we can find many recommendation systems in our everyday online life. For example, in sites like Amazon, based on our shopping history items will be recommended and in sites like IMDB, we can find movie recommendations. Why not implement the same in suggesting food? People are likely to try new food every time in different places. Instead of taking a wild guess of what he/she may like it is better to have recommendation system.

This paper focuses on building a recommendation system that will suggest food items based on user inputs and also provide user a list of nearby restaurants. Content based and collaborative filtering are the two approaches for recommender system. In collaborative filtering model is build based on user's past behavior as well as similar decisions made by others. Content based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties.

In this project, we make use of content based filtering to recommend food. It works in following fashion, first user is asked to input his/her favorite food items and location in which he wants to find restaurants. A list of top ingredients of user's favorite food items is fetched. Clusters of food items is maintained along with the list of important keywords that belong to respective cluster. Clustering is done by using k-means method.

A. Types of Recommender Systems

1. Item-Item Content Recommenders

Content recommenders are those which do not need ratings or user preferences. In item-item recommender it takes bunch of items, manually assigns a list of features and calculate how close or far away each item is from other item based upon the feature. For example, Lets consider the list of movies the user has been already watched and positively rated. We will have one more list with all movies that has been rated. Now we will recommend the user with the movies which he/she may like based upon the ratings of user watched movies.

2. Item-Item Collaborative Recommenders

Collaborative recommenders are those which uses the individual user behavior or attitude. In item-item recommender it looks at which items are tend to be purchased with other items and recommend these frequent ride-alongs. For example, there are three users who have rated all movies with almost same marks. If one of the user say A has not rated some movie say "hulk" which means the user has not watched this movie, then based upon the other users positive ranking this user will get this item recommended.

3. User-Item Collaborative Recommenders

In this recommender it takes the ratings of group of people who have rated things and distance is calculated between the pair of people based on rating and suggests most similar people to the user, in order to recommend that people who are similar to the user.

4. Hybrid Recommenders

This approach combines both content based and collaborative-based filtering. Here it predicts the content-based and collaborative-based predictions separately and then combines them or by unifying the approaches into one model or by adding collaborative-based capabilities into content-based approach (and vice versa). For example, user has to create an account on website and evaluate all movies he/she has watched. Here the recommender will make use of collaborative filtering first and with that outcome will apply content-based filtering. By combining these two approaches we get movie recommendations.

II. EXISTING SYSTEM

In Recipe Recommendation [18], they dealt with recipe interest problem that are related to website like AllRecipes.com. One example is, the site want to build a system to recommend new recipes to users. Here recommendation was given based on user's favorite recipes. Recommendation system gives list of recommendations in one of the two ways that is collaborative and content based filtering. In this they have experimented with content based, collaborative filtering and hybrid recommendation techniques to the recipe interest problem. After experimenting with all the approaches they found that collaborative filtering was the successful one, but the other two also have their own unique strengths.

Existing online food store asks user to input the ingredients that they want to have delivered. According to a study by Richmond we can see that users want to access the social aspects of physical store to have multiuser experience. A recommender system was designed for recipe based online food shopping. In this author used to label user's clusters like vegetarians and non vegetarians. These clusters were made available to user so that they can navigate among other user groups. This system works as follows, first user logs on to system and he/she is given recommended recipe. User can add recipe to shopping basket which in turn adds the ingredients. User can later ask for next recipe that matches category of user and recipe is recommended based on user group or category [13].

Food recommender system on Amazon [11] aims to build recommender system for grocery and gourmet food. Recommender system is built based on people's review on amazon.com. Amazon is one of the popular shopping website and has large number of user reviews on variety of products including different kinds of food. Several models are used in predict task like Bias SVD, Basic latent factor, linear regression and SVD++. These models are trained on different dataset based on characteristics of data. Their performance is evaluated in terms of mean squared error. As a result latent factor and SVD++ is suitable for experienced users and inexperienced users are best suited to linear regression model.

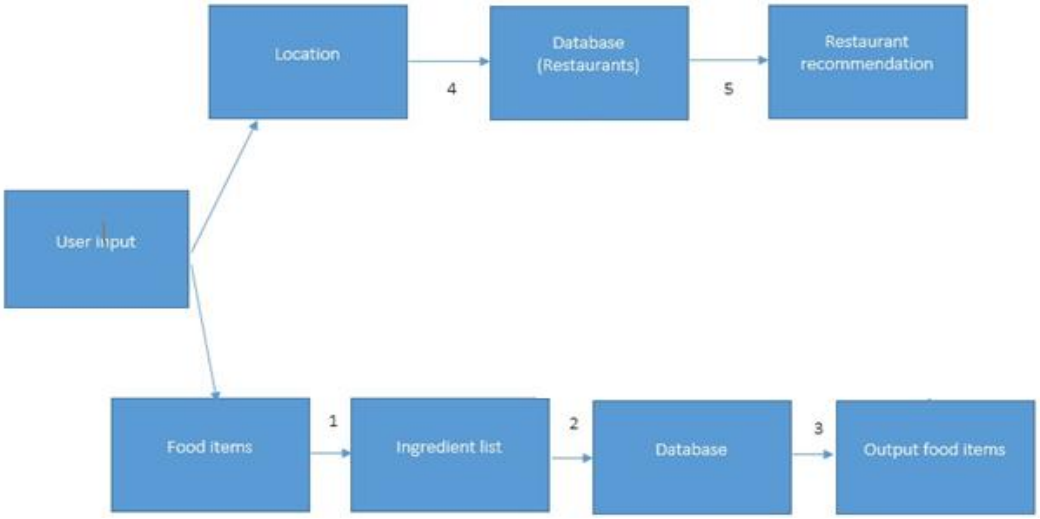
Personalized food recommender system suggests food to user in daily diet based on nutrition guidelines. Recommendations are made based on both individual diet needs and preference. Knowledge based framework is used to design the system. The basis for constructing knowledge base are ontology and some rule based knowledge [16].

III. PROPOSED SYSTEM

Recommender system is one of the most interesting topic in web for giants like Google, Amazon, Flipkart etc. as well as in research field. In order to overcome challenges that are faced in this field, there has been a lot of research. In this project, we are trying to increase the speed and performance of the recommender system by using an advanced approach for maintaining the database. Hence a lot of time can be saved by reducing the number of comparisons for finding the similarity in the database. By using this idea, we are storing the data in terms of clusters in different multiple levels. Hence once after getting the input, we list the ingredients of those items by vectoring them. In our project, we use tf-idf vectorizer for this purpose. Then we compare top terms of each cluster present in the outer level.

Most similar cluster is found and then if number of food items present in that cluster is greater than the required number of items, then we go inside the cluster and finding similarity is continued for the clusters present in next level. As we can imagine finding similarity for each and every food item takes $O(n^3)$ time complexity. But we are reducing "n" value significantly here.

We are using a Django web app as our user interface. This app has been made fully responsive using Bootstrap. By using a form we take input from this interface and then using the recommendation program we find the most similar food items and it is displayed on the screen of the web app. We have used Django app, so that we will have entire freedom to use our python program and all other web designs can be done very efficiently and easily.



1. Ingredients are listed using a tf-idf vectorizer.
2. Similarity is found between the ingredients list and the clustered database.
3. Most similar cluster is found.
4. By using Geocoding, we find the coordinates of the location.
5. Using Places API restaurant near to the location is displayed.

IV. IMPLEMENTATION AND PERFOMANCE ANALYSIS

Database is created from the dataset which are stored in text files. Dataset text files are food.txt and labels.txt. food.txt file contains all the food items with their ingredients and '*' is used at the end of every food item in order to differentiate the documents. labels.txt file contains the food item names. Database is stored using file system. If any cluster contains more than 10 items, then that is clustered again to reduce the number of items in a cluster. We have total 197 food items in our database.

We have used a hybrid similarity [6] measure for finding the similar keywords in the food that is present in the database and the top keywords obtained from the user input. This hybrid similarity measure is a hybrid of Jaccard's, N-grams and hlength.

$$JNLA = \frac{jaccard + Ngram + hlengt}{3}$$

$$hlengt = exp\left[\frac{-abs(len(A) - len(B))}{len(A \cup B)}\right]$$

And we have used document similarity measure for finding the similarity of documents [5].

$$Similarity_{Keyword}(p,q) = \frac{KN(p,q)}{Max\left(kn(p), kn(q)\right)}$$

kn(p) and kn(q) are the number of words present in the documents. KN(p,q) is the number of common words present two documents.

Table 1 gives the comparision of the output that is obtained for different keyword similarity measures. As we can see number of matched keywords vary a lot as smilarity measure changes. And during these tests, threshold value for 2 words to be matched was 0.6. And we have taken top 10 keywords or ingredients from the input and each cluster had 10 top keywords in their list.

Table 1. Comaprison of different similarity measures on count of matching keywords and corresponding output.

Input	**Keyword similaity measure**	**Ouput (Recommended Food items)**	
		Clustered Database	**Non-clustered database**
Mango pickle, Tomato Soup, Pudina Paratha	JNLA	Hot and Sour Tomato Pickle, Aratikaaya Masala (Spiced Raw Banana Curry)	sweet kachori, potato in curd gravy, spicy roti, vaangi bath, spicy sev,spicy khaja
	Jaccards	Gobi manchurian, baked bread rolls, potato cake, cabbage kofta curry, Kofta Lajawab	Seedai, Baigan fry, sweet kachori, spicy roti, spicy sev, spicy khaja
	Bigrams	Gobi manchurian, baked bread rolls, potato cake, cabbage kofta curry, Kofta Lajawab	Mango pickle, spicy roti, spicy sev, lime pickle, spicy khaja, Majjige Huli
	Cosine similarity	Gobi manchurian, baked bread rolls, potato cake, cabbage kofta curry, Kofta Lajawab	Mango pickle, spicy roti, spicy sev, lime pickle, spicy khaja, Majjige Huli

We have given three spicy items as input. In clustered database approach, JNLA gives the most similar items compared to other similarity measures. In Non-clustered database also we can conclude that JNLA [6] recommends the most similar items beacause JNLA combines the best of all the similarity measures resulting in good recommendation.

As we have explained earlier, we have used clustered databse for recommendation program. Now, we will see how this boosts the perfomance by reducing number of comparisions. We can see this by looking at the graphs Fig.1 and Fig.2. By comparing these graphs, we can see that we have reduced the time by 94%.

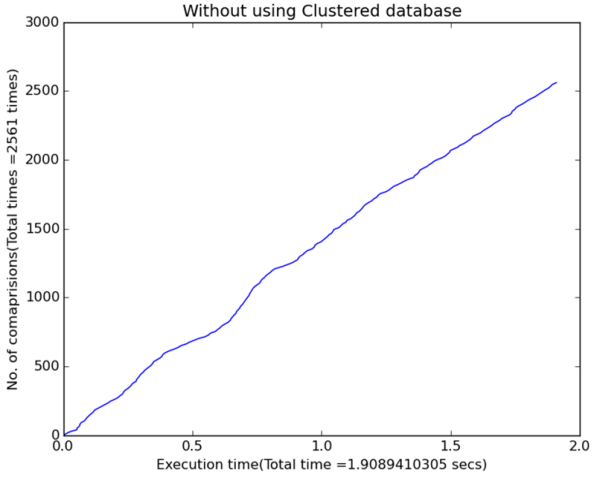


Fig 1

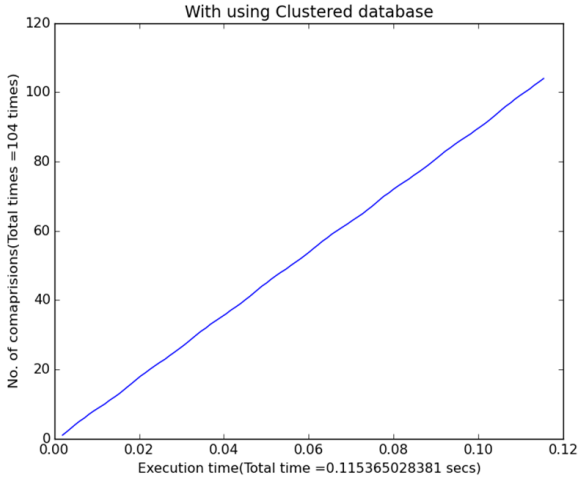


Fig 2

Now we will compare the output that has been produced when we use clustered database and non-clustered database. Table 2 shows the comparison.

Table 2. Comaprison of output produced in two different types of database

Input	**Output**	
	Clustered Database	**Non-clustered database**
Masala dosa, palak paneer, idli	Idli, Thaval Adai, Tamarind Uppma, Eggplant Gotsu, Plantain Podimas, Plantain Curry	Thattai, Baigan fry, masala dosa, Rasavangi, mixed dal dosa, bread besan bajji
Dal payasam, Jalebi, 7 cup sweet	Sweet pongal, Poli, Somaasi, Baadhusha, Kozhakattai, tandoori roti	paruppu thogaiyal, paruppu podi, sweet kachori, Motichoor Ladoo, 7 cup sweet, soan papdi
Mango pickle, Tomato Soup, Pudina Paratha	Hot and Sour Tomato Pickle, Aratikaaya Masala (Spiced Raw Banana Curry)	sweet kachori, potato in curd gravy, spicy roti, vaangi bath, spicy sev, spicy khaja

V. FUTURE ENHANCEMENTS

In our project, the inputs given by user were rendered in database for fetching the ingredients present on those food items. By extending this idea, we can scrape the ingredients present in the input in web, so that any food item can be given as input. We have used basic k-means algorithm for clustering. Instead of k-means other clustering methods like hybrid algorithms can be used for better performance and accuracy. User specific recommendations can be given using reinforcement learning approach. Only those restaurants can be recommended which has the recommended food items in their menu.

VI. CONCLUSION

By using clustered database in food recommendation system, we can speed up the execution in a large extent. But in our program we have used simple k-means for clustering the food items. By using advanced methods we can cluster the food items more accurately. Hence we can conclude that, by using clustered database we can address the problem of dimensionality. We have taken less amount of items here. But if the number is large, we can see the performance boost in a better manner.

REFERENCES

[1] E-Seok Jo, Yong-Sung Kim ,"A Good Food Restaurant Recommendation System Using Termite", International Journal of Advancements in Computing Technology(IJACT), Volume 5, Number 12, August 2013
[2] Ipek TATLI , "Food Recommendation System", June,2009
[3] Sumedh Sawant, Gina Pai, "Yelp Food Recommendation System"
[4] Marian Runo, "FooDroid: A Food Recommendation App for University Canteens", June 21, 2011
[5] Seung-Shik Kang, "Keyword-based Document Clustering"
[6] Jatsada Singthongchai and Suphakit NiwattanakulK. Elissa, "A Method for Measuring Keywords Similarity by Applying Jaccard's, N-Gram and Vector Space", Lecture Notes on Information Theory Vol. 1, No. 4, December 2013
[7] Wael H. Gomaa, Aly A. Fahmy, "A Survey of Text Similarity Approaches", International Journal of Computer Applications (0975 – 8887) Volume 68- No.13, April 2013
[8] Mihalcea., Corley, & Strapparava, "Corpus based and knowledge-based measures of text semantic similarity", Proceedings of the American Association for Artificial Intelligence 2006

[9] Miller, Beckwith, Fellbaum, Gross & Miller, "WordNet: An online lexical database"., Int. J. Lexicograph. 3, 4, pp. 235–244, 1990
[10] RVVSV Prasad1 and V Valli Kumari2, "A Categorical Review Of Recommender Systems", International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.5, September 2012
[11] Tao Huang, Huan Zhou, Kai Zhou, "Food recommender system on Amazon"
[12] Gediminas Adomavicius, Alexander Tuzhilin, "Context-Aware Recommender Systems"
[13] Martin Svensson, Jarmo Laaksolahti, Annika Waern, Kristina Höök, "A Recipe Based On-Line Food Store"
[14] Guy Shani, Asela Gunawardana, "Evaluating Recommendation Systems", Microsoft Research
[15] Michael D. Ekstrand, John T. Ried, Joseph A. Konstan, "Collaborative Filtering Recommender Systems", Foundations and TrendsR in Human–Computer Interaction, Vol. 4, No. 2 (2010) 81–173
[16] Napat Suksom, Marut Buranarach, Ye Myat Thein, Thepchai Supnithi, Ponrudee Netisopakul, "A Knowledge-based Framework for Development of Personalized Food Recommender System"
[17] Michael J. Pazzani, Daniel Billsus, "Content-based Recommendation Systems", FXPAL-PR-06-383
[18] Jeremy Cohen, Robert Sami, Aaron Schild, Spencer Tank, "Recipe Recommendation", May 13, 2014

