Machine Learning Based Food Recipe Recommendation System

Manju N.
JSS Science and Technology University, Mysuru

Machine Learning Based Food Recipe Recommendation System

M.B. Vivek, N. Manju and M.B. Vijay

Abstract Recommender systems make use of user profiles and filtering technologies to help users to find appropriate information over large volume of data. Users profile is important for successful recommendations. In this paper, we present two approaches to recommend recipes based on preferences of the user given in the form of ratings and compare them to identify which approach suits the dataset better. We use two approaches namely, item based approach and user based approach to recommend recipes. For item based approach Tanimoto Coefficient Similarity and Log Likelihood Similarity would be used to compute similarities between different recipes. For user based approach Euclidean Distance and Pearson Correlation are used. We use similarity techniques of user based approach and introduce fixed size neighborhood and threshold-based neighborhood to the same. The performance of the user based approach is found to be better than item based approach. The performance for the Allrecipe data set is found to be better than the simulated dataset since there are more number of interactions between users and items.

Keywords Collaborative filtering · Item based · User based · Fixed size neighborhood · Threshold-based neighborhood

M.B. Vivek (✉) · N. Manju
Information Science and Engineering, Sri Jayachamarajendra College of Engineering,
Mysore, India
e-mail: vivekmb9090@gmail.com

N. Manju
e-mail: manjun007@gmail.com

M.B. Vijay
Computer Science and Engineering, Sri Jayachamarajendra College of Engineering,
Mysore, India
e-mail: vijay85.m.b@gmail.com

D.S. Guru et al. (eds.), *Proceedings of International Conference on Cognition and Recognition*, Lecture Notes in Networks and Systems 14,
DOI 10.1007/978-981-10-5146-3_2

1 Introduction

Recommender systems are most commonly used in e-commerce websites to help user select items based on their interests or preferences. Recommendation systems present the user with a list of items on which a user might be interested, based on his current choice of an item. A recommender system makes use of information of the user's profile and compares the information to present a list of relevant recommendations.

The goal of a Recommendation system would be to give relevant recommendation to the users for items that might interest them. Designing a recommendation engine mainly would depend on the domain as well as the characteristics of the data available. A data source records the quality of interactions between users and items. The two approaches to build a recommendation system are—collaborative filtering approach and content based approach [1].

Collaborative filtering approach work by collecting user interactions in the form of ratings or preferences for each of the items and identify similarities amongst other users to determine how to recommend an item. Whereas in content based, recommendations are given by comparing representations of content describing an item to representations of content that interests the user. A hybrid recommendation system can be built by combining the collaborative filtering approach and content based approach.

A Recommender system goal is to predict a rating or preference that user's might give to an item ([1, 2]). Recommendation system is also used in financial services [3], twitter followers [4] and e-commerce [5]. Recommender systems produce recommendations either using collaborative based or content-based filtering approaches [6]. Collaborative filtering approach makes use of rating given to items previously purchased as well as similar ratings given by the users. In content based filtering approach a set of features of an item are utilized to make recommendation with similar properties [7]. These two approaches are combined to obtain a hybrid approach to make recommendations [8]. Herlocker provides an overview of evaluating recommendation systems [9] and Beel et al. provides the problems in offline evaluations [10]. Machine learning techniques are useful when there is vast information is available which has to be classified and analyzed, such as web information exploitation [11]. Talavera and Gaudioso [12] make use of classification techniques to analyze students' behavior. Their main goal is to reflect students' behavior, supporting tutoring activities on virtual learning communities. Zaiane's proposal [13] is one the first that used association rules. A review of many machine learning techniques is given by Adomanvicius and Tuzhilinin [1], where Decision trees, clustering, artificial neural networks and Bayesian classifiers are used. Alejandro Bellogin, Ivan Cantador provide the overview of personalized recommender systems using machine learning techniques [14]. Rajabi et al. [15] give an overview of recommender systems using profiles and machine learning methods. However, there only have been a few attempts to use machine learning techniques as we propose here. In our approach, Machine learning techniques are used to make recommendations, evaluate the system and make implicit improvement on its performance.

In this paper we make use of collaborative filtering approach i.e. item based collaborative filtering approach and user based collaborative filtering approach to give recommendations to users based on preferences given by each of them in the form of ratings to the recipes.

The remainder of this paper is organized as follows. In Sect. 1 we give a brief description of related work. In Sect. 2, we discuss methodology. In Sect. 3, experiments and results are discussed. Conclusion is discussed in Sect. 4.

2 Methodology

Our application makes use of Collaborative Filtering approach to make recommendations. Initial recipe recommendations are based on grocery items selected and therefore after the user rates recipes then recommendations will be made by computing similarities between different users based on their preference data. For each of the users, user profiles are created based on the history of recipes rated. To compute item based similarity for recipes based on preferences by the user, Tanimoto Coefficient similarity (see [16, 17]) and LogLikelihood similarity [18] are used. To compute user based similarity based on preferences for the recipes, Euclidean Distance [19] similarity and Pearson Correlation [20] similarity are used.

2.1 Item Based Recommendation

In item based collaborative filtering approach, recommendations are based on how similar recipes are to recipes. This type of recommendation just sees whether the user has rated a recipe or not. It does not take into account the values of the ratings. The similarity values are used to get a ranked list of recommended recipes. To calculate the similarity, we make use of two similarity measures namely, Tanimoto Coefficient similarity and LogLikelihood similarity.

Tanimoto Coefficient similarity is based on the Tanimoto Coefficient. This value is an extended Jaccard coefficient. It is the number of recipes that two users express some preference for, divided by the number of recipes that either user expresses some preference for. It is the ratio of the size of intersection to the size of union of the users' preferred recipes. The actual preference values do not matter, only their presence or absence does. When two recipes completely overlap, the result is 1. When they have nothing in common, it is 0. The value is never negative.

Tanimoto coefficient [21] is given by:

$$T(a,b) = \frac{N_c}{N_a + N_b - N_c}$$

where,

N_a Number of customers who rates item A
N_b Number of customers who rates item B
N_c Number of customers who rate both items A and B.

Log-likelihood-based similarity [22] is similar to the Tanimoto coefficient- based similarity. This also does not take into account the values of individual preferences. It is based on the number of recipes common between two users, but its value is more of how unlikely it is for two users to have so much overlap, given the number of recipes present and the number of recipes each user has a preference for.

To compute the score, let counts be the number of times the events occurred together (n_11), the number of times each has occurred without the other (n_12 and n_21) and number of times neither of these events took place (n_22). By having the above information Log-likelihood ratio score (also known as G^2) is computed as,

$$\mathbf{LLR} = \mathbf{2\,sum}(\mathbf{n})(\mathbf{S}(\mathbf{n}) - \mathbf{S}(\mathbf{rowSums}(\mathbf{n})) - \mathbf{S}(\mathbf{colSums}(\mathbf{n})))$$

where S is Shannon’s entropy, computed as the

$$\mathbf{sum}(\mathbf{n\_ij}/\mathbf{sum}(\mathbf{n}))\mathbf{log}(\mathbf{n\_ij}/\mathbf{sum}(\mathbf{n})).$$

2.2 User Based Recommendation

User based recommendations are based on the preferences given by the user and how similar the users are according to the preferences given by them. The similarity values are used to obtain a list of recommended recipes. To calculate the similarity between two users, we make use of two similarity measures namely, Pearson Correlation Coefficient similarity and Euclidean Distance similarity along with a fixed size neighborhood and threshold-based neighborhood.

An implementation of a similarity based on the Euclidean distance [23] between two users X and Y. Thinking of recipes as dimensions and preferences or ratings as points on those dimensions, a distance will be computed using all recipes (dimensions) where both users have expressed a preference for that recipe. This is simply the square root of the sum of the squares of differences in preferences or position along each dimension.

The similarity would be computed as:

$$1/\left(1 + distance/\sqrt{n}\right)$$

So the resulting values are in the range of (0, 1). This would weigh against pairs that overlap in more dimensions, which should indicate more similarity. More dimensions generally offer more opportunities to be farther apart. Actually, it is computed as

$$\sqrt{n}/(1 + distance)$$

where n is the number of dimensions. $\sqrt{\mathrm{n}}$ is chosen since randomly-chosen points have a distance that grows as $\sqrt{\mathrm{n}}$. This would cause a similarity value to exceed 1; such values are capped at 1. The distance isn't normalized in any way. Within one domain, normalizing wouldn't matter as much as it won't change the ordering. The implementation of the Pearson correlation [24] for two users X and Y is given as,

$\sum X^2$ **sum of the square of all X's preference values**.
$\sum Y^2$ **sum of the square of all Y's preference values**.
$\sum XY$ **sum of the product of X and Y's preference value for all items for which both X and Y indicate a preference**.

The correlation is then

$$\sum XY/\sqrt{\left(\sum X^2 * \sum Y^2\right)}$$

This correlation centers its data, shifts the user's rating values so that each of their means is 0. This is important, to achieve expected behavior on both the data sets. This correlation implementation is similar to the cosine similarity since the data it receives is centered-mean is 0. The correlation may also be interpreted as the cosine of the angle between the two vectors defined by the users' preference values.

3 Experiments and Results

For our work the recipe data is collected from Allrecipe website. There are about 46,336 recipes, 1,966,920 user reviews, and data from approximately 530,609 users to understand the fundamentals of cooking and user preferences. We scraped the data and obtained the data of about 940 users, 1.6 K recipes with 98 K user preferences. Along with the Allrecipe website data we also have our own data collected from the users using our application. There are 24 users, 124 recipes with 323 user preferences.

We implemented the item based approach making use of preferences given by the users' using two similarity techniques namely Tanimoto Coefficient Similarity and

LogLikelihood similarity. The recommendations are ranked according to the value of similarity measure. The performance of the approach is measured using classical Recall measure based on main ingredients. The recall measure is defined as

$$Recall = \frac{|\{\text{relevant recipes}\} \bigcap \{\text{retrieved recipes}\}|}{|\{\text{relevant recipes}\}|}$$

Table 1 results achieves an average recall of about 23 and 28% for Tanimoto Coefficient similarity and Log likelihood similarity respectively for Allrecipe dataset while it achieves an average recall of about 4 and 1% for the simulated dataset.

We implemented the user based approach making use of preferences given by the users' using similarity based on Euclidean Distance and Pearson Correlation with fixed size neighborhood and threshold-based neighborhood. The recommendations are ranked according to the value of similarity measure. The performance of this approach is measured over a fivefold evaluation using Average Absolute difference (AAD) and Root Mean Squared Error (RMSE) in estimated and actual preferences when evaluating a user based recommender for fixed size neighborhood and threshold based neighborhood for different percentages of training data and test data. Lower the values indicate more accurate recommendations for the respective datasets.

Table 2 shows the best results in estimated and actual preferences for Allrecipe dataset with user based recommender using two different similarity metrics with a nearest n user neighborhood.

Table 3 shows the best results in estimated and actual preferences for Allrecipe dataset using two different similarity metrics with a threshold-based user neighborhood.

Table 4 shows the best results in estimated and actual preferences for simulated dataset using two different similarity metrics with a nearest n user neighborhood.

Table 1 Average recall values for datasets

Implementation	Recall (Allrecipe dataset)	Recall (simulated dataset)
Tanimoto coefficient similarity	0.23	0.04
Log Likelihood similarity	0.28	0.016

Table 2 Results with 90% training data and 10% test data

Similarity	n = 100	n = 200	n = 300	n = 500	n = 1000
Pearson correlation (AAD)	0.83	0.7906	0.7782	0.8133	0.842
Euclidean distance (AAD)	0.7745	0.7484	0.7511	0.7667	0.820
Pearson correlation (RMSE)	1.04	0.9993	1.006	1.005	1.065
Euclidean distance (RMSE)	0.9783	0.9593	0.9649	0.9697	1.000

Table 3 Results with 80% training data and 20% test data

Similarity	t = 0.9	t = 0.8	t = 0.7	t = 0.6	t = 0.5
Pearson correlation (AAD)	0.9140	0.8771	0.8624	0.8339	0.8207
Euclidean distance (AAD)	0.8795	0.8919	0.8714	0.8022	0.7488
Pearson correlation (RMSE)	1.1177	1.1009	1.1005	1.0486	1.0424
Euclidean distance (RMSE)	1.1097	1.2740	1.1037	1.0090	0.9555

Table 4 Results with 90% training data and 10% test data

Similarity	n = 2	n = 4	n = 6	n = 8	n = 16
Pearson correlation (AAD)	1.663	1.69	1.69	1.576	1.639
Euclidean distance (AAD)	3.07	1.29	1.346	1.166	1.18
Pearson correlation (RMSE)	2.1357	1.839	1.678	2.078	1.8231
Euclidean distance (RMSE)	2.07	1.265	1.801	1.693	1.5893

Table 5 Results with 80% training data and 20% test data

Similarity	t = 0.9	t = 0.8	t = 0.7	t = 0.6	t = 0.5
Pearson correlation (AAD)	1.211	0.8977	1.6216	1.1686	0.9553
Euclidean distance (AAD)	1.666	0.8886	1.7776	1.95	1.2538
Pearson correlation (RMSE)	1.1956	2.2223	1.9247	1.8125	1.3195
Euclidean distance (RMSE)	2.84	2.0955	1.4513	2.2580	1.7751

Table 5 shows the best results in estimated and actual preferences for simulated dataset using two different similarity metrics with a threshold-based user neighborhood.

The performance or the estimated ratings for different users for Allrecipe dataset is found to be better than simulated dataset. This is because; the number of interactions between users and recipes in the form of ratings in the Allrecipe dataset is far greater than the simulated dataset. More the number of interactions mean the matrix constructed to compute similarity between users will be less sparse indicating more data available to the recommender system to identify similarities between different users.

4 Conclusion

In this paper we have implemented two approaches for recommending recipes based on user preferences in the form of ratings. Even though the running time of item based approach was better but more appropriate recommendations were given in the user based approach. The user based collaborative filtering along with the neighborhood of 200 users with Euclidean Distance similarity would provide more accurate recommendations for Allrecipe dataset. For simulated dataset a threshold of 0.8 with Euclidean distance similarity would provide us with good set of recommendations. User based approach was found to be more appropriate and performed better than item based collaborative filtering approach. Since the number of interactions between users and items are high in Allrecipe dataset, we find the user based approach for Allrecipe dataset to be better than simulated dataset. Results of the user based approach are found to be better than the item based approach.

This is a generalized approach for recommending recipes making use of user based approach. The same can be combined with content based approach to obtain a hybrid approach and can be applied in medical field to recommend medicines for patients suffering from various diseases and also in pharmaceuticals for recommending different drugs for different diseases.

References

1. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6), 734–749
2. Introduction to recommender systems handbook, Recommender systems handbook, Online ISBN 978-0-387-85820-3, Springer US (2011)
3. Facebook (2010) Pandora lead rise of recommendation engines—TIME
4. Felfernig A, Isak K, Szabo K, Zachar P (2007) The VITA financial services sales support environment. In: AAAI/IAAI 2007, pp 1692–1699. Vancouver, Canada
5. Gupta P, Goel A, Lin J, Sharma A, Wang D, Zadeh RB (2013) WTF: the who-to-follow system at Twitter. In: Proceedings of the 22nd International Conference on World Wide Web WWW 2013, Rio de Janeiro, Brazil. ACM 978-1-4503-2035
6. Almazro D, Shahatah G, Albdulkarin L (2010) A survey paper on recommender systems. arXiv:1006.5278v4[cs.IR]
7. Jafarkarimi H, Sim ATH, Saadatdoost R (2012) A Naïve recommendation model for large databases. Int J Inf Educ Technol (2012)
8. Mooney RJ, Roy L (1999) Content-based book recommendation using learning for text categorization. In: Workshop on recommender systems: algorithms and evaluation
9. Burke R (2002) Hybrid recommender systems: survey and experiments. User Model User-Adap Interact 12(4):331–370
10. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53. doi:10.1145/963770.963772
11. Beel J, Langer S, Genzmehr M, Gipp B (2013) A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation. In: Proceedings of the workshop on reproducibility and replication in recommender systems evaluation (RepSys) at the ACM recommender system conference (RecSys)
12. Srivastava J, Cooley R, Deshpande M, Tan P (2000) Web usage mining: discovery and applications of usage patterns from web data. SIGKDD Explor 1(2):12–23
13. Talavera L, Gaudioso E (2004) Mining student data to characterize similar behavior groups in unstructured collaboration spaces. In: Workshop on AI in CSCL, pp 17–23 (2004)
14. Zaïane OR (2006) Recommender system for e-learning: towards non-instructive web mining. In: Data mining in e-learning, pp 79–96

15. Bellogin A, Cantador I, Castells P, Ortigosa A (2008) Discovering relevant preferences in a personalized recommender system using machine learning techniques. Spanish Ministry of Science and Education (TIN2005–6885 and TIN2007-64718)
16. Rajabi S, Harounabadi A, Aghazarian V (2014) A recommender system for the web: using user profiles and machine learning methods. Int J Comput Appl (0975–8887) 96(11) (2014)
17. Seifoddini H, Djassemi M (2007) Merits of the production volume based similarity coefficient in machine cell formation
18. Abreu R, Zoeteweij P, Van Gemund A (2010) An evaluation of similarity coefficients for software fault localization
19. Evaluating and implementing recommender systems as web services using Apache Mahout. http://cslab1.bc.edu/~csacademics/pdf/14Casinelli.pdf
20. Madylova A, Oguducu SG (2009) A taxonomy based semantic similarity of documents using the cosine measure. In: Proceedings of international symposium on computer and information sciences, pp 129–134
21. Pearson correlation: definition and easy steps for use. http://www.statisticshowto.com/what-is-the-pearson-correlation-coefficient/
22. Discussion of similarity metrics. http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/sphilip/tani.html
23. Surprise and coincidence—musings from the longtail. http://tdunning.blogspot.in/2008/03/surprise-and-coincidence.html
24. Euclidean distance similarity. https://builds.apache.org/job/MahoutQuality/javadoc/org/apache/mahout/cf/taste/impl/similarity/EuclideanDistanceSimilarity.html
25. Pearson correlation similarity. https://builds.apache.org/job/MahoutQuality/javadoc/org/apache/mahout/cf/taste/impl/similarity/PearsonCorrelationSimilarity.html