Evaluation of Similarity Functions by using User based Collaborative Filtering approach in Recommendation Systems

Shaivya Kaushik[#1], Pradeep Tomar[#2]

[#1]*M-tech Scholar &Department of Computer Science, Gautam Buddha University*
[#2]*Department of Computer Science, Gautam Buddha University*
Gautam Buddha University,Greater Noida,India

Abstract-Recommendation Systems has been comprehensively analysed and are changing from novelties used by a few E-commerce sites in the past decades.Many of the popular and largest commerce websites are widely using recommendation systems.These are popular and important part of the e-commerce ecosystem that help users to find relevant and valuable information through large product spaces.The tremendous growth of visitors and the information poses few key challenges such as producing high quality recommendation systems,performing many recommendation systems per second for millions of users and items.The paper introduce user based collaborative filtering approach and the similarity function.The algorithm will identify relationships between different users and then compute recommendation for the users.This paperpresent a most commonly used similarity functions and their computation that aims to determine which similarity function result in producing most accurate recommendation.

Keywords-Recommendation Systems, Collaborative Filtering,Similarity Functions.

I. INRODUCTION

Recommendation System has been an important research topic in the last twenty years. Every day, we are inundated with choices and options. The sizes of these decision domains are frequently massive: Netflix has over 17,000 movies in its selection[1], and Amazon.com has over 410,000 titles in its Kindle store alone [2]. Supporting discovery in information spaces of this magnitude is a very significant challenge.Recommendation systems are asubclass of information filtering system that seek to predict 'rating' or 'preference' that a user would give to an item.The task of recommendation systems is to recommend items that fit a user's tastes,in order to help the user in purchasing items from an overwhelming set of options.These are providing personalized suggestion greatly increase the likelihood of a

customermaking a purchase compared to unpersonalized ones.

There are two basic strategies that can be applied when generating recommendations: content based and collaborative filtering.Content based approachesprofile users and items by identifying their characteristic features, such us demographicdata for user profiling, and product descriptions for item profiling. The profilesare used by algorithms to connect user interests and item descriptions when generating recommendations.

Collaborative filtering(CF), which attempt to predict what information will meet a user's needs based on data coming from similar users.Figure 1 represents basic structure of collaborative filtering algorithm. In the figure the collaborative recommendation system tries to found the peers of the target user that have similar tastes as the target user and then only the items that are most liked by the peers of target user would be recommended .In this paper the main focus on the collaborative filtering approach.The advantage of collaborative filteringapproach over content based is that CF approaches can be applied to recommender systems independently of the domain. They identify relationships between users and items, and make associations using this information to predict user preferences. However, content based approach is usually laborious to collect the necessary information about users, and similarly it is often difficult to motivate users to share their personal data to help create the database for the basis of profiling.

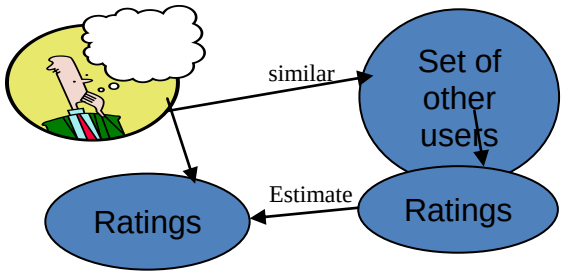

Fig1:Collaborative Filtering Structure

Collaborative filtering algorithms are usually categorized into two subgroups: memory-based and model-based.

Memory-based methods simply memorize the rating matrix and issue recommendations based on the relationship between the queried user and item and the rest of the rating matrix. The most popular memory-based CF methods are neighbourhood-basedor user-based methods, which predict ratings by referring to users whose ratings are similar to the queried user, or to items that are similar to the queried item. This is motivated by the assumption that if two users have similar ratings on some items they will have similar ratings on the remaining items. Or alternatively if two items have similar ratings by a portion of the users, the two items will have similar ratings by the remaining users.
Model-based methods, on the other hand, fit a parametric model to the training data that can later be used to predict unseen ratings and issue recommendations. Model-based methods include cluster-based CF [3, 4, 5, 6, 7], Bayesian classifiers [8, 9], and regression based methods [10]. The slope-one method [11] fits a linear model to the rating matrix, achieving fast computation and reasonable accuracy.In this paper user-based recommendation generation algorithm is analysed and implemented.The paper focus on different similarity functions for computing user to user similarities for generating recommendations for them.

A. General steps of recommendation system using Collaborative filtering approach

INPUT: A rating matrix consisting of users,items and their rating(user, item, top N users)

OUTPUT:Recommendation of items to user

Step1.Creating Rating Matrix:A recommender system needs information about a user more specifically a user's preferences in order to generate recommendations. These are stored in a ratings database, also known as a rating matrix.

Step 2.Calculating Similarity: Users tend to trust people in a given context with whom they share common views. This is the reason why similarity values are computed between users in a recommender system.

Step3.Finding top N Neighbours: Finding neighbours is typically done by sorting out the users with the highest similarity value.

Step4.GeneratingRecommendations:
Recommendations are generally made only for items which the active user has not rated. For every such item, a predicted rating is computed based on the ratings from every user in the neighbourhood on that particular item. This implies that only neighbours who have rated the item are considered.

II. Literature Survey

The literature survey is mainly categorize in two parts. One part is related to research related to collaborative filtering approach and other part is related to research related to similarity functions.

A. Research about Collaborative Filtering

The main aim of the collaborative filtering is to use people with similar preference to recommend the information needed.Through cooperation,the information needed is recorded and filtered to help recommend a more accurate result for the user. The information collected does not limit to those from people with similar preference. It is also important to collect those information that is from people with dissimilar preference. This is known as social filtering. This is an important feature for the e-commerce. The paper focus on thecustomer's past purchasing behaviour and compare it with other customers to find the customers with similar purchasing behaviour to recommend a list of items for this customer that theymight like. From the preferences of the group we are able to recommend products and services for a single person. In recent years, different algorithms with different mathematics formulas have been applied to improve the recommendation system by finding the strength of interest and these mathematics formulas establish a strong basis for collaborative filtering. Collaborative filtering does not provide a completely accurate solution, but the inclusion of mathematics formulas have indeed triggered many applications of collaborative filtering. Besides e-commerce, collaborative filtering is also applied for information retrieval, network personal video cabinet, and personal bookshelves.

The concept of collaborative filtering descends from the work in the area of information filtering.

The developers of one of the first recommender systems, Tapestry (other earlier recommendation systems include rule-based recommenders and user-customization), coined the phrase, collaborative filtering who first to publish in account of using collaborative filtering technique in the filtering of information. They built a system for filtering email called Tapestry which allowed users to annotate message. Annotations became accessible as virtual fields of the message, and uses could construct filtering queries which accessed those fields. Users could then create queries such as .show me all office memos that Bill thought were important.. The collaborative filtering provided by Tapestry was not automated and required users to

construct complex queries in a special query language designed for the task. The term collaborative filtering has been widely adopted in the field of recommender systems regardless of the facts that recommenders may not explicitly collaborate with recipients and recommendations may suggest particularly interesting items, in addition to indicating those that should be filtered out .

The fundamental assumption of CF is that if users X and Y rate n items similarly, or have similar behaviours (e.g., buying, watching, listening), and hence will rate or act onother items similarly . The collaborative filtering technique applied to recommender systems matches people with similar interests and then makes recommendationsbased on this basis. Recommendations are commonly extracted from the statistical analysis of patterns and analogies of data extracted explicitly from evaluations of items (ratings) given by different users or implicitly by monitoring the behaviour of the different users in the system.

In collaborative filtering a user's profile consists simply of the data the user has specified. This data is compared to those of other users to find overlaps in interests among users.These are then used to recommend new items. Typically, each user has a set of .nearest neighbours. defined by using the correlation between past evaluations. Predicted scores for un-evaluated items of a target user are predicted by recommender system using a combination of the actual rating scores from the nearest neighbours of the target user .

The problem of lack of transparency in the collaborative filtering systems was introduced in [12]. Collaborative systems today are black boxes, computerized oracles which give advice but cannot be questioned. A user is given no indicators to consult in order to decide when to trust a recommendation and when to doubt one. These problems have prevented acceptance of collaborative systems in all but low-risk content domains since they are untrustworthy for high-risk content domains.
Early generation collaborative filtering systems, such as GroupLens, use the user rating data to calculate the similarity or weight between users or items and make predictions or recommendations according to those calculated similarity values. The so-called memory-based collaborative filtering methods are notably deployed into commercial systems because they are easy-to-implement and highly effective .Customization of CF systems for each user decreases the search effort for users. It also promises a greater customer loyalty, higher sales, more advertising revenues, and the benefit of targeted promotions.

User-based CF methods [13] identify users that are similar to the queried user, and estimate the desired rating to be the average ratings of these similar users. Similarly, item-based CF [14] identify items that are similar to the queried item and estimate the desired rating to be the average of the ratings of these similar items. Neighbourhood methods vary considerably in how they compute the weighted average of ratings. Specific examples of similarity measures that influence the averaging weights are include Pearson correlation, Vector cosine, and Mean-Squared-Difference (MSD). Neighbourhood based methods can be extended with default votes, inverse user frequency, and case amplification [13]. A recent neighbourhood -based method [15] constructs a kernel density estimator for incomplete partial rankings and predicts the ratings that minimize the posterior loss.

B. Research about Similarity Functions

From the scientific and mathematical point of view, similarity/distance is defined as a quantitative degree that enumerates the logical separation of two objects represented by a set of measurable characteristics[16][17]. Measuring similarity or distance between two data points is a core requirement for several data mining and knowledge discovery tasks that involve distancecomputation. Examples include clustering (k-means), distance-based outlier detection,classification (KNN, SVM), and several other data mining tasks. These algorithms typically treat the similarity computation as an orthogonal step and can make use of any measure. For continuous data sets, the Minkowski Distance is a general method used to compute distance between two multivariate points. In particular, the MinkowskiDistance of order 1 (Manhattan) and order 2 (Euclidean) are the two most widely used distance measures for continuous data. The key observation about the above measures is that they are independent of the underlying data set to which the two points belong. Several data driven measures have also been explored for continuous data. The notion of similarity or distance for categorical data is not as straightforward as for continuous data. The key characteristic of categorical data is that the different values that a categorical attribute takes are not inherently ordered[18]. Thus, it is not possible to directly compare two different categorical values. The simplest way to address similarity between two categorical attributes is to assign a similarity of 1 if the values are identical and a similarity of 0 if the values are not identical. For two multivariate categorical data points, the similarity between them will be directly proportional to the number of attributes in which they match. Various similarity measure functions are enumerated in the literature

[19][20] such as Euclidean distance, cosine similarity whose applications are widespread in retrieving information or data from databases.

III. Similarity Computation

Similarity computation between items or users is a critical step in memory-based collaborative filtering algorithms. For a user-based *CF* algorithm, we first calculate the similarity, between the users*u* and *v* who have both rated the same items.There are many ways to determine the similarity between two things. In order to represent this similarity in a machine, there is a need to define a similarity score. If quantify different attributes of data objects, then employ different similarity algorithms across those attributes that will yield similarity scores between the different data objects. For example, represent people as data objects whose attributes are tastes in movies. Use a similarity metric to help us find which people are similar based on how similar their tastes are.There are many different methods to compute similarity or weight between users or items.

A. Euclidean Distance

A simple yet powerful way to determine similarity is to calculate the Euclidean Distance between two data objects. To do this, we need the data objects to have numerical attributes. We also may need to normalize the attributes. For example, if we were comparing people's rankings of movies, we need to make sure that the ranking scale is the same across all people; it would be problematic to compare someone's rank of 5 on a 1-5 scale and another person's 5 on a 1-10 scale.

$$Sim(u,v)=\sqrt{\sum_{i=0}^{n} (u_i - v_i)^2} \quad (1)$$

B. City-block distance

The city-block distance, alternatively known as the Manhattan distance, is related to the Euclidean distance. Whereas the Euclidean distance corresponds to the length of the shortest path between two points, the city-block distance is the sum of distances along each dimension:

$$d = \frac{1}{n}\sum_{i=1}^{n} |u_i - v_i| \quad (2)$$

This is equal to the distance you would have to walk between two points in a city, where you have to walk along city blocks. The city-block distance is a metric, as it satisfies the triangle inequality. Again we only include terms for which both u_i and v_i are present, and divide by *n* accordingly.

C. Pearson Correlation

In this case, similarity between two items i and j is measured by computing the Pearson-r correlation $corr_{i,j}$. To make the correlation computation accurate we must first isolate the co-rated cases (i.e., cases where the users rated both i and j) . Let the set of users who both rated i and j are denoted by U then the correlation similarity is given by

$$Sim(i,j) = \frac{\sum_{u\in U} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u\in U} (R_{u,i} - \bar{R}_i)^2}\sqrt{\left(\sum_{u\in U} (R_{u,j} - \bar{R}_j)^2\right)}} \quad (3)$$

Here $R_{u,i}$ denotes the rating of user u on item i, $\bar{R}_i$ is the average rating of the i-th item.

D. Adjusted Cosine Similarity

One fundamental difference between the similarity computation in user-based CF and item-based CF is that in case of user-based CF the similarity is computed along the rows of the matrix but in case of the item-based CF the similarity is computed along the columns, i.e., each pair in the co-rated set corresponds to a different user .
Formally, the similarity between items i and j using this scheme is given by

$$Sim(i,j) = \frac{\sum_{u\in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u\in U} (R_{u,i} - \bar{R}_u)^2}\sqrt{\left(\sum_{u\in U} (R_{u,j} - \bar{R}_u)^2\right)}} \quad \mathbf{(4)}$$

Here $\bar{R}_u$ is the average of the u-th user's ratings.

IV. Neighbourhood Selection

After the similarity computation, CF algorithms have to select the most similar users for the active user. This is the important step since the recommendations are generated using the ratings of neighbours and therefore neighbourhood has an impact on the recommendation quality.The neighbourhood selection is done by selecting the top nearest-neighbours purely according to their similarities with the active user.

V. Generating Recommendation

The most important step in a collaborative filtering system is to generate the output interface in terms of prediction.Once we isolate the set of most similar items based on the similarity measures, the next step is to look into the target users ratings and use a technique to obtain predictions. The sum of all users' opinions is added to the active user's average rating. A positive result means that the users combined gave the item in question a higher rating than usual. If the result is negative instead, the predicted rating will be lower than the target user u's average ratings, indicating that the item is not as good as an average item. A list of n items with the highest predicted ratings is then returned to the user.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{k} (r_{u,i} - \bar{r}_u) \times sim(a,u)}{\sum_{u=1}^{k} sim(a,u)} \quad \textbf{(5)}$$

Here $\bar{r}_a$ is the mean rating user for a and $u_{1....}u_k$ are the k nearest neighbours to a. The*sim*(*a*,*u*) *is similarity* between *a* and *u*

VI. Implementation andEvaluation

Consider the rating matrix which consists of five users and seven item.The cells marked '-' indicate unknown values(the user has not rated that item).

Table 1:Rating Matrix (1-5 star rating scale)

	I1	I2	I3	I4	I5	I6	I7
U1	**5**	**3**	**2.5**	**-**	**-**	**-**	**-**
U2	**2**	**3.5**	**5**	**2**	**-**	**-**	**-**
U3	**2.5**	**-**	**-**	**4**	**4.5**	**-**	**5**
U4	**5**	**-**	**3**	**4.5**		**4**	**-**
U5	**4**	**3**	**2**	**4**	**3.5**	**4**	**-**

In our implementation all the above stated similarity functions which measures similarity between two users have been implemented in MATLABS.The application of equation 4 to our running example,constructs the rating similarity matrix i.e, cosine similarity which is depicted in Table 2:

Table 2:Cosine Similarity Matrix

	User1	User2	User3	User4	User5
User 1	1.0000	0.7548	0.2398	0.6112	0.6262
User 2	0.7548	1.0000	0.2526	0.6475	0.6248
User 3	0.2398	0.252	1.000	0.442	0.5937
User 4	0.6112	0.6475	0.4429	1.0000	0.8364
User 5	0.6262	0.6248	0.5937	0.8364	1.0000

Let's assume in our running example,that we want to predict the rating of user 5 on item 7.If we take into account the cosine similarity between two users depicted in Table 2 ,then we compute the predicted rating of a user for an item by using equation 5 and the neighbours average rating is computed through influence of neighbors on the user 5 :

$P_{5,7}$=2.9+ ((0-2.3)*0.8)+((0-1.5)*0.6)+((0-1.6)*0.6))/0.8+0.6+0.6

$P_{5,7}$=1.08

Hence,we predict the ratings on all the unrated items by using above all similarity functions.In our paper we implemented all the similarity functions described in above section on our rating matrix depicted in Table 1.

A. Evaluation Metrics

Several metrics have been proposed for accessing the accuracy of collaborative filtering methods.They are divided into two categories mainly:statistical accuracy metrics and decision-support accuracy metrics. In this Paper, we use the statistical accuracy metrics.

Statistical accuracy metric evaluates the accuracy of a prediction algorithm by comparing the numerical deviation of the predicted ratings from the respective actual user ratings. Some of them are MAE(Mean Absolute Error), RMSE(Root Mean Square Error). Both these were computed on the result data and provided the same conclusions. It amplifies the contributions of the absolute errors between the predictions and actual ratings and is defined as:

$$RMSE = \sqrt{\frac{\sum_{u} (pi(u) - ri(u))^2}{n}} \quad \textbf{(6)}$$

wheren is the total number of ratings over all users, $pi(u)$ is the predicted rating for user u on item i, and $ri(u)$ is the actual rating. The lower values of $RMSE$ entail better predictions.The table 3 describes the RMSE computed on above similarity functions.

Table 3:RMSE of Similarity Functions

Similarity Functions	RMSE
a. Euclidean	1.5411
b. Cityblock	1.5999
c. Pearson Correlation	3.5180
d. Adjusted Cosine	1.4727
e. Spearman Correlation	3.5180

VII. Results and Discussion

In this section we present our experimental results of applying user-based collaborative filtering techniques for generating predictions.Here accuracy of metrics is evaluated by comparing the RMSE of all the similarity functions.

Here implemented five similarity functions Euclidean,city block,pearson correlation,adjusted cosine,spearman as described in section 3 and tested on our data set. For each similarity function, we implemented the algorithm to compute the neighbourhood and used weighted sum algorithm to generate the prediction. We ran these experiments on our training data and used test set to compute RMSE. Figure 2 graphshows the experimental results. It can be observed from the results that cosine similarity RMSE is significantly lower in this case. Hence, cosine similarity function is producing quality and accurate recommendation.

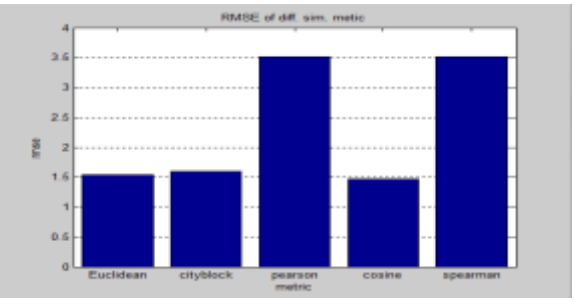

Fig2: Comparison of various Similarity Functions

VIII. Conclusion

Collaborative filtering approach is now one of the most successful technique.The memory based CF technique,neighbourhood based CF computes similarity between users or items and then use the weighted sum of ratings or simple weighted average to make predictions based on the similarity values.In the paper the computation of similarity between two users using various similarity functions is evaluated.From the results,the result can be concluded that from all the similarity functions,the cosine similarity function is most accurate method for generating recommendation. In future work clustering technique in recommendation system should be introduced to perform more efficient recommendations.

ACKNOWLEDGEMENT

I would like to thanks to all my friends and also my closest friend without his support this work will not be completed successfully and family who helped me in several critical situations . I am grateful to my Lord to give me energy to perform well.

REFERENCES

[1] J. Bennett and S. Lanning, "The netflix prize," in *KDD Cup and Workshop 07*, 2007.

[2] Amazon.com, "Q4 2009 Financial Results," Earnings Report Q4-2009 January 2010.

[3] L. H. Ungar and D. P. Foster.Clustering methodsfor collaborative filtering. In AAAI Workshop on Recommendation Systems, 1998.

[4] S. H. S. Chee, J. Han, and K.Wang.Rectree: An efficient collaborative filltering method.In Lecture Notes in Computer Science, pages 141{151. Springer Verlag, 2001.

[5] M. Connor and J. Herlocker.Clustering items for collaborative filtering.In Proceedings of ACM SIGIR Workshop on Recommender Systems, 2001.

[6] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Recommender systems for large scale e-commerce: Scalable neighborhood formation using clustering. In Proceedings of the 5th International Conference on Computer and Information Technology, 2002.

[7] G. R. Xue, C. Lin, Q. Yang, W. S. Xi, H. J. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing.In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 114{121. ACM New York, NY, USA, 2005.

[8] K. Miyahara and M. J. Pazzani.Collaborative filtering with the simple bayesian classifier.In Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence, pages 679{689, 2000.

[9] K. Miyahara and M. J. Pazzani.Improvement of collaborative filtering with the simple bayesian classifier 1. (11), 2002.

[10] S. Vucetic and Z. Obradovic.Collaborative filtering using a regression-based

approach.Knowledge and Information Systems, 7:1{22, 2005.

[11] D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. Society for Industrial Mathematics, 05:471{480, 2005.

[12]. Konstan, J., Miller, B., Maltz, D., Herlocker, J.Gordon, L., and Riedl, J. GroupLens: *ApplyingCollaborative Filtering to Usenet News. Communications of the ACM*, 40(3), 1997, 77-87.

[13] J. Breese, D. Heckerman, and C. Kadie.Empirical analysis of predictive algorithms for collaborative filtering.In Proc. of Uncertainty in Artificial Intelligence, 1998.

[14] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl.Item-based collaborative filtering recommendation algorithms.In Proc. of the international conference on World Wide Web, 2001.

[15] M. Sun, G. Lebanon, and P. Kidwell. Estimating probabilities in recommendation systems. In Proc. of the International Conference on Artificial Intelligence and Statistics,2011.

[16]. Hariri B B, Abolhassani H, and Khodaei A.: "A new Structural Similarity Measure for Ontology Alignment", in Proc. SWWS, 2006, pp.36-42.

[17]. Hongmei Wang, Sanghyuk Lee, and Jaehyung Kim.: " Quantitative Comparison of Similarity Measure and Entropy for Fuzzy Sets" ADMA '09 Proceedings of the 5th International Conference on Advanced Data Mining and Applications, pp. 688–695, 2009.

[18]. JouniSampo and PasiLuukka .: "Similarity Classifier with Generalized Mean; Ideal " Fuzzy Systems And Knowledge Discovery Lecture Notes in Computer Science, 2006, Volume 4223/2006, 1140-1147.

[19]. Sung-Hyuk Cha, "Comprehensive Survey on Distance/Similarity Measures between Probability Density Function", International Journal of Mathematical Models and Methods in Applied Sciences, Issues 4, Volume 1, 2007.

[20]. Randall Wilson D, Tony R. Martinez, Improved Heterogeneous Distance Functions Journal ofArtificial Intelligence Research ,Vol. 6, pp. 1-34, 1997.