A Research Paper on Data Analysis (rating distribution, time series analysis) & using The Recommendation Algorithm with Cosine similarity, Correlation matrix and SVD on Amazon dataset

Pranjal Chowdhury [1], ParthoProtim Sarkar [2]

[1](Computer Science &Engineering , Institute of Engineering & Management, Kolkata , India, pranjalprim@gmail.com)

[2](Computer Science & Engineering , Institute of Engineering & Management, Kolkata , India, sarker624452@gmail.com)

Abstract*—Nowadays . Data analysis has become a very important area for both companies and researchers as a consequence of the technological developments in recent years. Recommender system has a long history as a successful application in artificial intelligence. Collaborative filtering or recommender systems use a database about user preferences to predict additional topics or products a new user might like. This paper describes some algorithms designed for this task including cosine-based similarity algorithm and correlation-based similarity algorithm. we conclude that correlation based similarity algorithm acts better than Cosine based similarity algorithm . We adopted free-formatted rating-based into traditional 2-Dimensional SVD approach. We analyzed the effect of different rating similarity techniques to the 3-Dimensional SVD recommendation performance.*

***Keywords*—Recommendation Systems, Singular Value Decomposition, Collaborative filtering, correlation-based similarity algorithms, cosine-based similarity algorithm**

1. INTRODUCTION

Rating system is very important things now-a-days. It provides a good solution of all. We can get & understand that which product is good or not.Recommender systems do facilitate the selection of data items by users by issuing recommendations for items they might like. In particular, they aim at providing suggestions to users by estimating their item preferences and recommending those items featuring the maximal predicted preference. Typically, recommendation approaches are classified as content-based and collaborative filtering approaches. In content-based approaches, information about the features/content of the items is processed, and the system recommends items with features similar to items a user likes. In collaborative filtering approaches, we produce interesting suggestions for a user by exploiting the taste of other similar users.

Nowadays, recommendations have more broad applications, beyond products, like news recommendations [9], links recommendations [15, 13], and more innovative ones like query recommendations [3], health recommendations [14], open source software recommendations [7] and diverse venue recommendations [4]. For achieving efficiency, there are approaches that build user models for computing recommendations. For example, [10] applies subspace clustering to organize users into clusters and employs these clusters, instead of a linear scan of the database, for making predictions.

2. RELATED WORK

Maria Stratigi, Xiao Zhou Li, Kostas Stefanidis, and Zheying Zhang

effectiveness of this textual review sentiment-based recommender system is evaluated via a case study on the Amazon dataset. According to the findings, using sentiment score-based rating mechanism can provide more reasonable numeric score for the target items and therefore a more intuitive view of the item quality. In addition, the effectiveness of the recommender system based on sentiment rating is as high as that with regular numeric ratings.

3. MATERIALS AND METHODS

1. SVD RECOMMENDATION

A. Definition of SVD

Singular Value Decomposition is a matrix factorization technique which takes a rectangular matrix defined as A where A is an *m x n* matrix in which the *m* rows represents the users, and the *n* columns represents the items. The SVD theorem (1) states:

$$\mathbf{A}_{mxn} = \mathbf{U}_{mxm}\,\mathbf{S}_{mxn}\,\mathbf{V}^{T}_{nxn} \qquad (1)$$

$$\text{Where } \mathbf{U}^{T}\mathbf{U} = \mathbf{I}_{mxm}$$

$$\mathbf{V}^{T}\mathbf{V} = \mathbf{I}_{nxn}$$

Where the columns of U are the left singular vectors; S (the same dimensions as *A*) has singular values and is diagonal; and V^T has rows that are the right singular vectors. Calculating the SVD consists of finding the Eigenvalues and Eigenvectors of $\mathbf{AA}^{\mathbf{T}}$ and $\mathbf{A}^{\mathbf{T}}\mathbf{A}$. The Eigenvectors of $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ make up the columns of **V** , the Eigenvectors of $\mathbf{AA}^{\mathbf{T}}$ make up the columns of **U**. Also, the singular values in **S** are square roots of Eigenvalues from $\mathbf{AA}^{\mathbf{T}}$ or $\mathbf{A}^{\mathbf{T}}\mathbf{A}$. The singular values are the diagonal entries of the **S** matrix and are arranged in descending order. The singular values are always real numbers. If the matrix **A** is a real matrix, then **U** and **V** are also real.

Matrix **S** is a diagonal matrix having only r nonzero entries, which makes the effective dimensions of **U**, **S** and **V** matrices $m \times r$, $r \times r$, and $r \times n$, respectively. The diagonal entries $(s_1, s_2, \ldots, s_r)$ of S have the property that $s_i > 0$ and $s_1 \geq s_2 \geq \ldots \geq s_r$

B. Recommendation Using SVD

In order to remove noise data from a large and sparse database, some dimensionality reduction techniques are proposed[1, 2, 3]. SVD is a successful dimensionality reduction technique which is used for making recommendations. SVD-based recommendation algorithms produce high quality recommendations, but has to undergo computationally very expensive matrix factorization steps[1].

SVD provides the best low-rank linear approximation of the original matrix. It is possible to reduce dimensions by selecting greatest k singular values. The value of k may change according to the size and the structure of data.

The reduced matrix $\mathbf{S_k}$ is constructed by retaining the first $\boldsymbol{k}$ singular values. The matrices **U** and **V** are also reduced to produce matrices $\mathbf{U_k}$ and $\mathbf{V_k}$, respectively. The matrix $\mathbf{U_k}$ is produced by removing $\boldsymbol{(r - k)}$ columns from the matrix **U** and matrix $\mathbf{V_k}$ is produced by removing $\boldsymbol{(r - k)}$ rows from the matrix **V**. Multiplying these three reduced matrices, the matrix $\mathbf{A_k}$is obtained. The reconstructed matrix $\mathbf{A_k}$is a matrix that isthe closest approximation to the original matrix *A*. Figure 1 demonstrates this process.

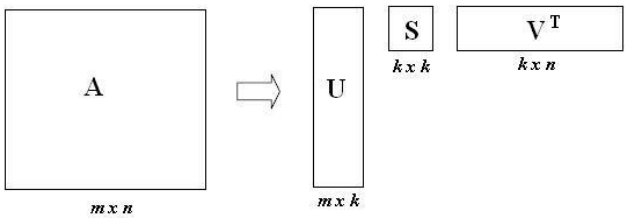

Fig. 1. Dimensionality Reduction Process in SVD

Low-rank approximation of the original matrix is better than the original matrix itself. Filtering out of the small singular values can be introduced as removing "noise" data in the matrix. Each customer and product is represented by its corresponding Eigenvector in SVD-based recommender systems. For instance, for a movie recommender system, users who rated similar products are mapped into the space spanned by the same Eigenvectors [1, 2].

2.COLLABORATIVE FILTERING (CF)

One of the most successful technologies for recommender systems, called collaborative filtering, has been developed and improved over the past decade to the point where a wide variety of algorithms exist for generating recommendations [10,11]. It aims at finding the relationships among the new individual and the existing data in order to further determine the similarity and provide recommendations [8]. In fact, it exploits user ratings of products in order to identify additional products that the new user may like as well.

Collaborative filtering systems are being applied to larger and larger sets of items. With large numbers of items in the prediction domain, we see the occurrence of 3 significant negative phenomena:

1. Since users have limited resources to experience items (read articles, see movies, listen to music), the density of user ratings on items decreases. It becomes less likely that any significant number of a user's neighbors will have experienced the item for which a prediction is being requested.
2. While the density will decrease, the number of items that must be considered in each user-to-user correlation will still increase, increasing the amount of time necessary to compute the neighborhood.
3. As the number of items in the prediction domain gets large, the diversity of those items will also increase. As this diversity increases, it becomes less likely that a user's opinions on all other items will be relevant to their opinion on a single given item [12]. A very common method for solving these problems is clustering which is used in the most popular recommendation sites like amazon.com. To find customers who are similar to the user, cluster models divide the customer into a lot of segments and treat the task as a classification problem. The algorithm's goal is to assign the user to the segment containing the most similar customers. It then uses the purchases and ratings of the customers in the segment to generate recommendations. There exits many research in this area [13].

RecTree (which stands for Recommendation Tree) addresses the scalability problem with a divide-and-conquer approach. The method first performs an efficient k-means-like clustering to group data and creates neighborhood of similar users and then performs subsequent clustering based on smaller, partitioned databases. Since the progressive partitioning reduces the search space dramatically, the search for an advisory clique will be faster than scanning the entire database of users. In addition, the partitions contain users that are more similar to each other than those in other partitions. This characteristic allows RecTree to avoid the dilution of opinions from good advisors by a multitude of poor advisors, and thus, yielding a higher overall accuracy [14].

In other work sparsity and dimensionality reduction are addressed by first clustering items based on user access patterns so as to attempt to minimize the apriori probability that recommendations will cross cluster boundaries and then recommending only within clusters. The inherent dynamic nature of the problem is addressed by explicitly modeling the data as a time series; and is showed how this representational expressivity fits naturally into a maxent framework [15].

Collaborative filtering approaches are often classified as memory-based and model-based. In the memory-based approach, all rating examples are stored *as-is* into memory. In the prediction phase, similar users or items are sorted based on the memorized ratings. Based on the ratings of these similar users or items, a recommendation for the test user can be generated. Examples of memory-based

collaborative filtering include user-based methods and item-based methods. The advantage of the memory-based methods over their model-based alternatives is that fewer parameters have to be tuned; however, the data sparsity problem is not handled in a principled manner. In the model-based approach, training examples are used to generate a model that is able to predict the ratings for items that a test user has not rated before. Examples include decision trees, aspect models, and latent factor models. The resulting compact models solve the data sparsity problem to a certain extent. However, the need to tune an often significant number of parameters has prevented these methods from practical usage.

A. User-based Collaborative Filtering

User-based collaborative filtering predicts a test user's interest in a test item based on rating information from similar user profiles. Each user profile (row vector) is sorted by its dissimilarity toward the test user's profile. Ratings by more similar users contribute more to predicting the test item rating. In the top-N case, a set of top-N similar users toward test user can be generated [16].

B. *Item-based Collaborative Filtering*

Item- based approaches apply the same idea, but use similarity between items instead of users. The unknown rating of a test item by a test user can be predicted by averaging the ratings of other similar items rated by this test user. Again, each item (column vector) is sorted and re-indexed according to its dissimilarity toward the test item in the user-item matrix, and ratings from more similar items are weighted stronger.

In tourism recommender system and especially for destination recommendation, we exactly use user-based method because we do not have very detailed information about the previous travels of a current tourist. Most users of this system rarely use it because people generally travel a few times a year; thus, neither enough data about their preferences nor their ratings of travel's items is gained. Due to the reasons mentioned above, neither can item-based method be used nor can users' ratings of a new item in our recommender system be calculated.

In addition, we just use necessary and important data because most people are not patient enough to fill in the form, so we use data like age, budget, etc. to find the most similar tourist to the current tourist and use its travel experience for a new tourist [16].

3.**Cosine similarity**

Cosine similarity is a metric used to determine how similar two entities are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space.

When we say two vectors, they could be two product descriptions, two titles of articles or simply two arrays of words.

$$\vec{a}\cdot\vec{b}=\sum_{i=1}^{n}a_i b_i = a_1b_1 + a_2b_2 + \cdots + a_nb_n$$

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2}$$

$$\|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

Mathematically, if 'a' and 'b' are two vectors, cosine equation gives the angle between the two.

4. RESULTS AND DISCUSSION

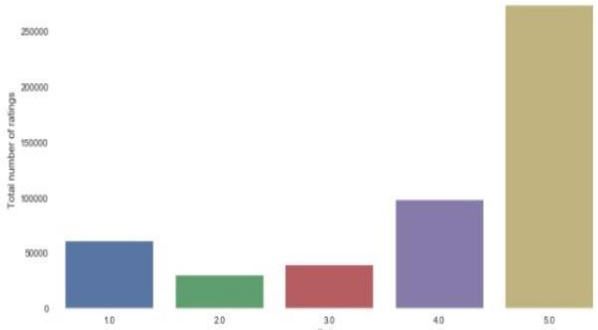

Fig : Total number of ratings vs rating

Fig: Number of rate product per user

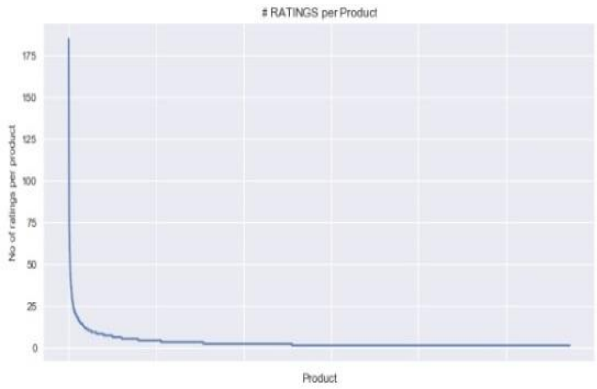

Fig: Ratings per product

```
['0594481813',
 '1400532736',
 '9625993428',
 'B00000O2F',
 'B000001OKH',
 'B000001OMJ',
 'B000001OMT',
 'B000001OMV',
 'B000001ON0',
 'B000003055',
 'B00000J1F3',
 'B00000J1QK',
 'B00000J1UA',
 'B00000J47V',
 'B00000J4DT',
 'B00000J4E4',
 'B00000J4GE',
 'B00000JBHE',
 'B00000JBPN',
 'B00000K2YR',
 'B00000K2ZO',
 'B00000K3QL',
 'B00000K3RG',
 'B00001MXWD']
```

```
len(Recommend)
```

```
544
```

Fig: collaboration products id

	UserId	Asin	Rating
6	A3J3BRHTDRFJ2G	0511189877	2.0
118	AT09WGFUM934H	0594481813	3.0
178	A17HMM1M7T9PJ1	0970407998	4.0
189	A2IDCSC6NVONIZ	0972683275	5.0
295	A23A7OH3AM0PDA	0972683275	4.0
389	A6J8D9V5S9MBE	0972683275	5.0
406	A39Z4OU2C7ENWH	0972683275	3.0
631	A3TAS1AG6FMBQW	0972683275	5.0
886	AZFF4CX9MQ4AE	0972683275	5.0
889	A2HI35H2BOCV9R	0972683275	3.0

Fig: User ID & Ratings

UserId	A100UD67AHFODS	A100WO06OQR8BQ	A101OAAMZYWQ3U	A1027EV8A9PV1O	A1029ESOR657OQ	A1
Asin						
0511189877	0	0	0	0	0	0
0594481813	0	0	0	0	0	0
0970407998	0	0	0	0	0	0
0972683275	0	0	0	0	0	0
1400501466	0	0	0	0	0	0

5 rows × 4267 columns

Fig: User id rating in columns

5. CONCLUSION

Various methods of recommender systems were studied and used to implement the amazon recommender system . Among the various recommendation methods collaborative filtering was introduced as the most proper one for this system. And among the collaborative filtering similarity algorithms, correlation-based similarity, and cosine-based similarity were selected .Using experimental results the amazon recommender system was implemented. Correlation based similarity was introduced as the most accurate one for prediction and recommendation . In order to present a more complete and more precise amazon recommender system, it is propose to focus on other various recommendation methods specially clustering method and users classification in future works . From the experiments, we observe the contribution of rating into 2-Dimentional SVD method . By having semantically grouped rating, SVD-based approach will produce more accurate results.

REFERENCES

[1] E. Bigdeli and Z. Bahmani, "Comparing accuracy of cosine-based similarity and correlation-based similarity algorithms in tourism recommender systems," 2008 4th IEEE International Conference on Management of Innovation and Technology, Bangkok, 2008, pp. 469-474, doi: 10.1109/ICMIT.2008.4654410.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] Anuranjan Kumar, Sahil Gupta, S. K Singh, K. K. Shukla, "Comparison of various metrics used in collaborative filtering for recommendation system", Contemporary Computing (IC3) 2015 Eighth International Conference on, pp. 150-154, 2015.K. Elissa, “Title of paper if known,” unpublished.

[3] Luisa Hernández, Heitor Costa, "Identifying Similarity of Software in Apache Ecosystem -- An Exploratory Study", Information Technology - New Generations (ITNG) 2015 12th International Conference on, pp. 397-402, 2015.

[4] Zheng Wan, "Personalized Tourism Information System in Mobile Commerce", Management of e-Commerce and e-Government 2009. ICMECG '09. International Conference on, pp. 387-391, 2009.

[5] Rudolf Turnip, Dade Nurjanah, Dana Sulistyo Kusumo, "Hybrid recommender system for learning material using content-based filtering and collaborative filtering with good learners' rating", e-Learning e-Management and e-Services (IC3e) 2017 IEEE Conference on, pp. 61-66, 2017.

[6] Nhan Nguyen-Thanh, Dana Marinca, Kinda Khawam, Steven Martin, Lila Boukhatem, "Multimedia Content Popularity: Learning and Recommending a Prediction Method", Global Communications Conference (GLOBECOM) 2018 IEEE, pp. 1-7, 2018.

[7] Volume Removed - Publisher's Disclaimer", Energy Procedia, vol. 13, pp. 1, 2011.

[8] Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC (2010).

[9] Hutto, C.J., Gilbert, E.: VADER: A parsimonious rule-based model for sentiment analysis of social media text. In: ICWSM (2014)

[10] . Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: The Adaptive Web, Methods and Strategies of Web Personalization (2007)

[11] Yin, Z., Gupta, M., Weninger, T., Han, J.: LINKREC: a unified framework for link recommendation with user attributes and graph structure. In: WWW (2010)

[12] M. O'Conner, and J. Herlocker, Clustering Items for Collaborative Filtering. ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation, University of California, Berkeley, August 19, 1999.

[13] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, John Riedl: Item-based collaborative filtering recommendation algorithms. WWW 2001: 285-295.

[14] Berry, M. W., Dumais, S. T., and O'Brian, G. W. (1995). Using Linear Algebra for Intelligent Information Retrieval. SIAM Review, 37(4).

[15] . Koskela, M., Simola, I., Stefanidis, K.: Open source software recommendations using github. In: TPDL (2018)

[16] Eirinaki, M., Abraham, S., Polyzotis, N., Shaikh, N.: Querie: Collaborative database exploration. IEEE Trans. Knowl. Data Eng. 26(7), 1778–1790 (2014)