Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/15/20230812

A content-based collaborative filtering algorithm for movies and TVS recommendation

Ziqi Wang

School of Management and Economics, Beijing Institute of Technology, Beijing, 102488, China

1120201047@bit.edu.cn

Abstract. With the rapid development of multimedia technology and the constant upgrading of film and television libraries, users' demand for movies and television is increasing. How to accurately and timely find favorite movies from massive movie and television resources according to user's preferences and needs has become a great challenge. In recent years, the recommendation of movies and TVs has attracted a lot of research interest from academia and industry. The existing recommendation algorithms mainly include content based and collaborative filtering. The former recommends projects through collaborative learning of others' interests, while the content-based method examines the rich context of the project. In this paper, to further improve the performance of recommendations, a content based collaborative filtering method is proposed to provide recommendations for movies and television. Specifically, we extract and vectorize feature and category information from movies based on TF-IDF and apply truncated SVD to reduce the dimensions of the rating and TF-IDF matrix to retain the most representative information. We calculate the cosine similarity between the vectors from these two matrices. The final recommendation is to list 10 movies based on the average similarity of content and ratings. Extensive experiments on Amazon review data have proven the effectiveness of this method.

Keywords: movie recommendation, content, collaborative filtering

1. Introduction

In recent years, Internet information and film websites have exploded, and film and television resources are unusually rich. However, various movies and television cannot be effectively integrated, which leads to so much information and makes it hard for people to quickly find the movies they like. To this end, how to accurately recommend the desired movie from the massive film and television resources has become a challenge, attracting a large amount of research interest from academia and industry. Accurate movie recommendation not only brings convenience to users, but also brings more profits and traffic to movie websites.

In the current digital era, recommendation plays an import role in our daily life, which aims at predicting the user choices and produce results according to user preference. According to the difference of algorithm designing, the existing recommendation systems are usually divided into recommender based on collaborative filtering, content, and Hybrid methods [1,2,3]. Content-based recommendation manage to list items similar to what users favored in the history as a result [4]. The text of items, like

DOI: 10.54254/2755-2721/15/20230812

description and category, is then transformed into an unordered bag of words and the examples represented as a vector of words [5]. Then, items will be recommended based on the similarity of contexts and attributes. In common cases, this kind of systems are used when there is abundant attribute information [6]. Therefore, other users play little role in this way. Unlike Content-based approach, Collaborative Filtering relies on the $m \times n$ user-item matrix, which contains m users and n items, to leverages the ratings of other users and calculate the similarities between items. The basic idea behinds it is that similar items receive similar ratings. Some famous systems, like Ringo/Firefly [7] and Recommender [8], are using this technique.

If used in isolation, these two approaches have their own shortcomings. For collaborative filtering, it may encounter problems like the sparsity and cold start. As for the content-based method, it just recommends items similar to what users have rated, leading to less novelty. For years, researchers have been exploring the hybrid technique to eliminate many of the weakness of each approach. Fab designs a partial hybridization approach. In this way, content-based methods are used to classify the peer group, while the ratings are leveraged in the recommendation process [6]. In recent years, more advanced techniques have been developed. For example, a hybrid recommendation approach for articles, introduced by Wang et al. [9], manages to incorporate social tag and friend information in scientific social network. A hybrid scholarly recommendation method, proposed by Sakib,N. et al.[10] integrates metadata in scientific papers. Other methods include hybrid collaborative filtering model integrating deep presentation learning and matrix factorization [11], recommendation algorithm combining user trust network with probability matrix factorization [12], and so on.

In this paper, we try to combine the content-based and collaborative filtering methods to recommend movies and TVs. The remaining section of this paper is organized as follows. The overall architecture of our design is given in Section II. Section III introduces the whole procedure of our approach in details. Finally, evaluation and future works are presented in Section IV.

2. Method

2.1. General system architecture

In this study, we try to use content-based together with collaborative filtering by averaging the similarity scores calculated with these two approaches. We manage to make full use of the advantages of content-based filters and reduces the effects of their shortcomings.

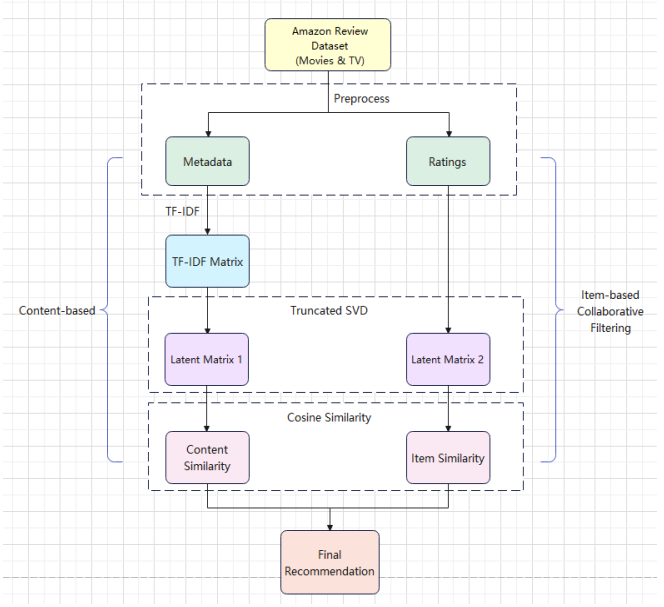

Figure 1. Overall Architecture of System.

DOI: 10.54254/2755-2721/15/20230812

Figure 1 depicts the procedure of our proposed method. For the metadata, which is about the detailed information about the movies and TVs, we use text mining approach to transform it into vectors in high dimensional space. For the ratings, we apply Truncated SVD method to it, just like in the content-based phrase, to reduce the dimensionality for efficiency when reserving as much important information as possible. Then, using the two matrices generated respectively, we calculate the similarities between movies. For final recommendation, we combine the similarity results by computing and ranking the average of them and return a list of top-10 similar movies as recommendations.

2.2. Original datasets

In the project, the recommendation system is designed based on the metadata and over 8 million ratings of about 20 thousand Movies and TVs, which are subsets of the complete Amazon review dataset (2018) [1]. The product metadata include 19 attributes like descriptions, category information, price, brand and so on. And ratings are recorded with user ID, product ID and time. Parts of these two datasets are shown in Table 1 and 2.

Table 1. Rating dataset.

	user	item	rating	timestamp
0	A3478QRKQDOPQ2	0001527665	5.0	1362960000
1	A2VHSG6TZHU1OB	0001527665	5.0	1361145600
2	A23EJWOW1TLENE	0001527665	5.0	1358380800

Table 2. Metadata dataset of movies and TVs.

	category	description	title	brand
0	[Movies & TV, Movies]	[Disc 1: Flour Power (Scones;Shortcakes;…)]	My Fair Pastry (Good Eats Vol.9)	Alton Brown
1	[Movies & TV, Movies]	[Barefoot Contessa Volume 2: On these three…]	Barefoot Contessa (with Ina Garten),…	Ina Garten
2	[Movies & TV, Movies]	[Rise and Swine (Good Eats Vol.7) includes…]	Rise and Swine (Good Eats Vol.7)	Alton Brown

2.3. Data preprocessing

Though there are quite a few records in the dataset, many of them are redundant with same information. Therefore, the first step we carried out was to drop the duplicated records. Then, since we are trying to apply the content-based method to the metadata of movies and TVs, we need to clean the columns which contain useful context, but the original structure is inappropriate. In our practice, columns of "category" and "description" are chosen to generate the "Bag of words" for each product. All punctuations are removed, and all terms are transformed into lowercase letters. After that, for convenience, we just reserve the cleaned columns. In our practice, except for the ID, title and bag of words, other columns have been dropped. The final cleaned metadata of movies and TVs are in Table 3.

Table 3. Cleaned metadata of movies and TVs.

item	title	Bag_of_words
0000695009	Understanding Seizures and Epilepsy	movies
0000143529	My Fair Pastry (Good Eats Vol.9)	disc1 flour power scones shortcakes…
0000143592	Rise and Swine (Good Eats Vol.7)	rise and swine good eats vol7 includes…

Similar operations are also carried out on the rating dataset, which contains lots of repeated records. Besides, for time-sequential effects are not taken into consideration in our simple model, we drop the

DOI: 10.54254/2755-2721/15/20230812

column of "timestamp" as well. Using the cleaned datasets, a larger table can be generated, containing all the information needed in our analysis in later phase. Note that because of the large size of the complete dataset, we just select 20000 records for training. Table 5 shows part of the final table.

Table 4. Final table after merging.

	item	title	Bag_of_words
3911304	B002DLB1IO	Anvil: The Story of Anvil	at 14 toronto school friends steve lips…
1035588	6305476098	The Confession	hired to defend a client who killed to…
3463376	B001AQR3LC	The Tudors: Season 3	henry tudor must overcome his despair…

From the merged table, we can get some rough information about the users and products. Surprisingly, most of the users have ratings that are less than five, while the most active one has commented on over 4000 movies. The distribution of users' ratings is shown in Table 5.

Table 5. Distribution of users' ratings.

Quantile	25%	50%	75%	Max
Ratings	1	1	2	4254

Besides, we also manage to find out 20 most active users, and the results are depicted in Figure 2. From the bar chart, we can see that the number of ratings of the top 1 user is almost twice as many as that of the second one. And for the users followed, the figures just decrease steadily.

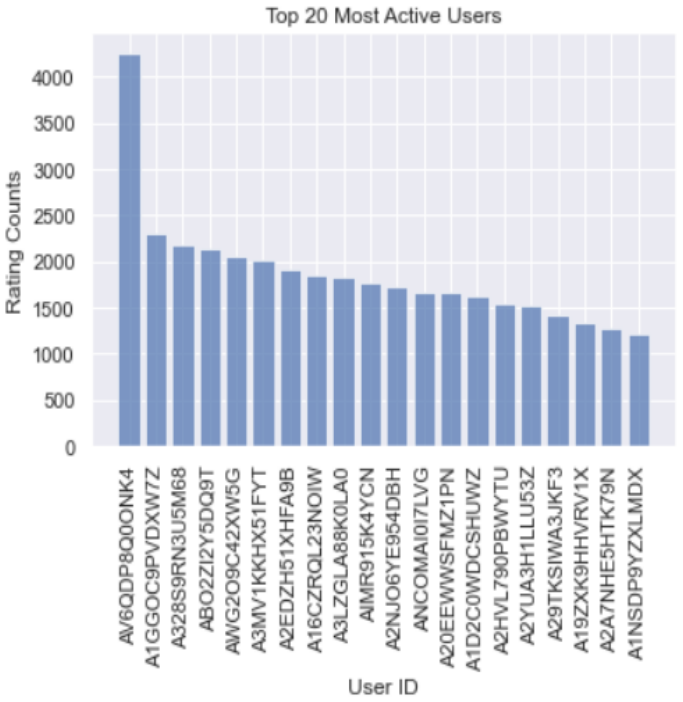

Figure 2. Top 20 Users.

Similarly, we explore the movies data. The results are shown in Table 6 and Figure 3. It seems that though over half of the movies have few ratings, the works Band of Brothers is really popular among the users, with about 50,000 ratings in total.

DOI: 10.54254/2755-2721/15/20230812

Table 6. Distribution of movies' ratings.

Quantile	25%	50%	75%	Max
Ratings	2	4	17	24543

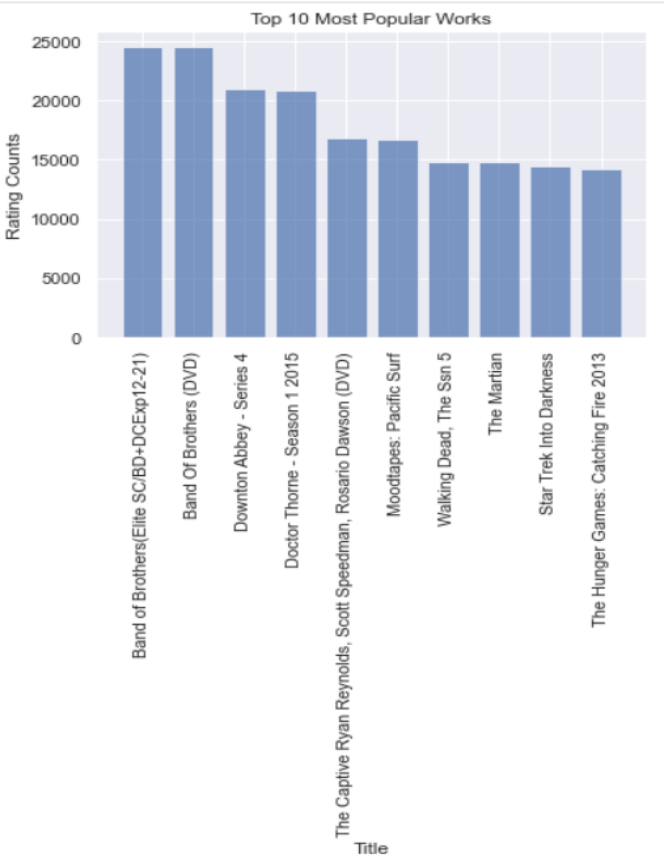

Figure 3. Top 10 Movies.

2.4. Content-based analysis

In our project, metadata is used in this phase. The categories and descriptions are the sources for generating the "bags of words". By removing useless information like punctuation and stop words in data preprocessing, we can get cleaned data prepared for analysis.

2.4.1. TF-IDF vectorization. Nowadays, TF-IDF (Term frequency-inverse document frequency) is one of the most famous term weighting schemes in the field of text mining. It has been used to measure word relatedness [13]. If one specific term appears really frequently in the document set, it will be assumed as a more common term which is less helpful to distinguish one document from the others. However, if it just appears in one document frequently, it is more likely to be regarded as the keyword of the document, and it should be more weighted.

In this approach, for a set of documents which contains m terms in total, a document D is transformed to an m-dimensional vector, and each dimension represents a term. Using TF-IDF, the term weight is calculated as:

$$w_i = tf_i \times \log(\frac{n}{df_i}) \quad (1)$$

Where n documents are in the set, tf_i represents the times of appearance of term t_i in document D and df_i is the number of documents in which term t_i occurs [14].

DOI: 10.54254/2755-2721/15/20230812

Using Tf-idf Vectorizer from Python, we can get a 10610×61703 matrix with rows of movies and columns of terms. Table 7 shows parts of the result. Since the complete matrix is too large to operate calculation on it, for the next step we use the truncated SVD to reduce its dimension.

Table 7. TF-IDF matrix for movies.

	0	1	2	3	4
3911304	0.0	0.040343	0.0	0.0	0.0
1035588	0.0	0.000000	0.0	0.0	0.0
3463376	0.0	0.000000	0.0	0.0	0.0

2.4.2. Truncated SVD. SVD (Singular Value Decomposition) is a popular method of dimensionality reduction [4]. It shrinks the space dimension from N to K $(K < N)$. For the $n \times d$ matrix M, SVD decomposes it into three other matrices:

$$M = U\Sigma V^T \tag{2}$$

Where Σ is an $k \times k$ diagonal matrix with nonnegative elements, U and V are $n \times k$ and $d \times k$ matrix respectively, and both of them consist of orthonormal columns. In this way, the value k is the rank of M[15].

2.5. Item-based collaborative filtering

For the ratings data, since the unnecessary column has been removed in the data preprocessing, now we just consider the problem of its large size. After transforming it into pivot table, it is in the format of user-item rating matrix, which is shown in Table 8. As we can see in the table, usually the rating matrix is sparse, containing lots of zeros.

Table 8. User-item Rating Matrix.

user item	**A0019420MGJRFO7TA5QC**	**A0283642BURXFWRSJIJT**
0005019281	0.0	0.0
0005119367	0.0	0.0
0307142493	0.0	0.0

2.6. Final recommendation

In item-based recommendation approaches, cosine similarity is usually used to measure how related two items are. Transformed into a high dimensional space, these two vectors' similarity is calculated based on the angle between them. In the fields of information retrieval and text mining, text documents, which are represented as vectors of terms, are also compared in this way [1]. The formula to calculate cosine similarity is:

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \times |\vec{b}|} \tag{3}$$

In our process, we calculate the cosine similarity of movies in the two matrices respectively. The content-based matrix shows how movies are similar in context, while collaborative filtering matrix sees them from a different perspective based on ratings they get. And in the final phrase, the similarities computed in two ways will be combined to give the recommendations. As a really primitive model, we just use the average of them for ranking the items.

DOI: 10.54254/2755-2721/15/20230812

3. Experiments and performance analysis

3.1. Evaluation metrics

Instead of predicting rating values of the users, our model ranks t items for users and recommend top-k items. The length of the recommended list becomes rather important. If the list is short, the user may miss relevant items and we will lose potential customers. In this case, it is called false-negative. However, if the list is really long, the user may be bored with so many repeated and irrelevant recommendations(false-positive).

Therefore, to evaluate the accuracy of this model, we use the indicators like precision, recall and F1-score. To calculate the indicators mentioned above, first let us think of a recommendation list with t items and denote the set of the recommended items as $S(t)$, and the true set of relevant items as G. Then, the precision will be calculated as follows [6]:

$$Precision(t) = \frac{|S(t) \cap G|}{|S(t)|} \tag{4}$$

And the recall is defined as:

$$Recall(t) = \frac{|S(t) \cap G|}{|G|} \tag{5}$$

To make a trade-off between them, $F_1 - score$ is calculated as:

$$F_1 = \frac{2 \times Precision(t) \times Recall(t)}{Precision(t) + Recall(t)} \tag{6}$$

In our project, $S(t)$ is the list of recommended movies, and G refers to all movies seen by audience of the given input. We randomly select 500 movies to evaluate the performance of top-10 recommendation list, and the evaluation function will return the average F_1 score of them. In our test, we get the result as around 0.94.

3.2. Effectiveness of Truncated SVD

To evaluate the performance of Truncated SVD, we conduct several experiments to see the performance of Truncated SVD on TF-IDF Matrix. Truncated SVD produces the closest rank-k approximation of a given input matrix [16]. Unlike the regular SVD, it can generate a factorization where the number K of columns can be specified (usually $K < rank(M)$). In practice, we use this method to extract the most representative features. Setting the parameter K as 3000, its performance is shown in Figure 4. It can be observed that majority of the original information can be reserved.

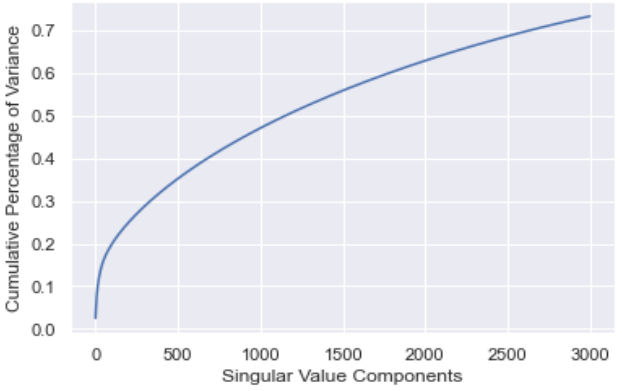

Figure 4. Performance of Truncated SVD on TF-IDF Matrix.

Similarly, Truncated SVD is also used in collaborative filtering analysis, with the parameter set as 3000. Its performance is illustrated in the Figure 5. All the results demonstrate the necessary of introducing the Truncated SVD into our method.

DOI: 10.54254/2755-2721/15/20230812

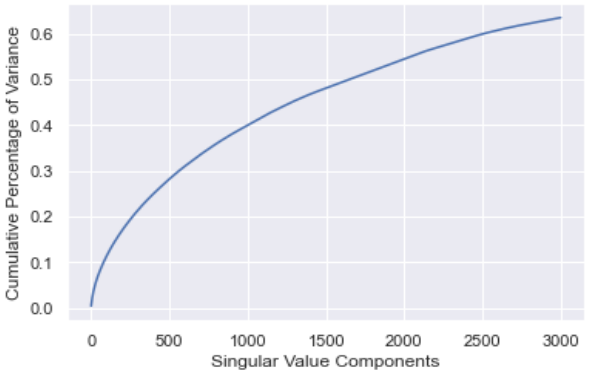

Figure 5. Performance of Truncated SVD on User-Item Matrix.

3.3. Performance analysis

For test, we input the ballet drama "The Flames of Paris" to this model, hoping to find its related works. And the recommendation list provided is in Table 9. Obviously, they are all related with ballet.

Table 9. The Recommended Movies.

Movie	Content Based	Collaborative	Final
Ballet 422	0.513553	3.422993e-07	0.256777
Ballet 201, Beyond the Basics-VHS	0.492126	-1.441504e-04	0.245991
Tchaikovsky-The Nutcracker/Maximova, Vasiliev, Boishoi VHS	0.450763	-1.353671e-04	0.225314
Prima Princessa Presents Swan Lake	0.445547	-1.333689e-03	0.222107
The Red Shoes	0.440672	-3.417913e-04	0.220165
New York City Ballet Workout VHS	0.437733	-1.021616e-03	0.218356
Balanchine Library-Balanchine Essays-Arabesque VHS	0.429910	-8.964244e-04	0.214507
Felia Doubrovska Remembered-From Diaghilev's Ballets Russes to Balanchine's School of American Ballet	0.404366	-1.908196e08	0.202183
The Merry Widow: Martins, McBride, New York City Ballet VHS	0.396656	-3.749068e04	0.198140
Beginner Ballet Barre	0.357443	-9.983903e04	0.178222

4. Conclusion and future work

Recommendation systems are widely used model and we have built movies and TVs recommendation system using content-based and item-based collaborative filtering approaches. As for evaluation, F_1 score is used to exam its accuracy. Though it seems it performs well, we should keep in mind that we

DOI: 10.54254/2755-2721/15/20230812

just use a very small proportion of the original dataset due to the restrictions of the hardware. Apart from that, the time and space complexity of the program are still problems, especially when it runs on a large-scale dataset. What's more, what we have done is just a primitive experiment, for all the methods are used separately, but not encapsulated in a so-called system. For future scope, the problems mentioned above need solving, and more advanced technique as well as models should be taken into consideration.

References

[1] Jannach D, Zanker M, Felfernig A and Friedrich, G. 2010. Recommender Systems: An Introduction (Cambridge: Cambridge University Press)

[2] Zhang S, Yao L, Sun A and Tay Y. 2019. J. Deep learning-based recommender system: A survey and new perspectives. ACM computing surveys (CSUR), 52(1), 1-38.

[3] Adomavicius G and Tuzhilin A. 2005. J. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, 17(6), 734-749.

[4] Lu Z, Dou Z, Lian J, Xie X, and Yang Q. 2015. Proc. Int. Conf. on Artificial Intelligence vol 29 no 1. Content-based collaborative filtering for news topic recommendation.

[5] Mooney R J and Roy L. 2000. Proc. Int. Conf. on 5th ACM Conf. on Digital libraries. Content-based book recommending using learning for text categorization. pp 195-204.

[6] Aggarwal C C. 2016. M. Recommender systems (Vol. 1) (Cham: Springer International Publishing)

[7] Shardanand U and Maes P. 1995. Proc. Int. Conf. On SIGCHI Conf. on Human factors in computing systems. Social information filtering: Algorithms for automating "word of mouth". pp 210-217.

[8] Hill W, Stead L, Rosenstein M and Furnas G. 1995. Proc. Int. Conf. On SIGCHI Conf. on Human factors in computing systems. Recommending and evaluating choices in a virtual community of use. pp 194-201.

[9] Wang G, He X and Ishuga C I. 2018. J. HAR-SI: A novel hybrid article recommendation approach integrating with social information in scientific social network. Knowledge-Based Systems, 148, 85-99.

[10] Sakib N, Ahmad R B, Ahsan M, Based M A, Haruna K, Haider J and Gurusamy S. 2021. J. A hybrid personalized scientific paper recommendation approach integrating public contextual metadata. IEEE Access, 9, 83080-83091.

[11] Dong X, Yu L, Wu Z, Sun Y, Yuan L and Zhang F. 2017. Proc. AAAI Conf. on artificial intelligence vol 31 no 1. A hybrid collaborative filtering model with deep structure for recommender systems.

[12] Yang F R, Zheng Y J and Zhang C. 2018. J. Hybrid recommendation algorithm combined with probability matrix Factorization. Computer Application, vol 38 no 3 pp 644–649.

[13] Yih W T and Qazvinian V. 2012. Proc. Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Measuring word relatedness using heterogeneous vector space models. pp 616-620

[14] Van Meteren R and Van Someren M. 2000. Using content-based filtering for recommendation. Proc. of the machine learning in the new information age: MLnet/ECML2000 workshop vol 30 pp 47-56

[15] Aggarwal C C, Aggarwal L F and Lagerstrom-Fife. 2020. Linear algebra and optimization for machine learning (Vol. 156) (Cham: Springer International Publishing)

[16] Frank M and Buhmann J M. 2011. Proc. IEEE Int. symposium on information theory. Selecting the rank of truncated SVD by Maximum Approximation Capacity. pp 1036-1040