Personalized Recommendation System with User Interaction based on LMF and Popularity Model

Aakanksha Sharaff
Department of Computer Science and Engineering
National Institute of Technology
Raipur, India
asharaff.cs@nitrr.ac.in

Sarthak Khurana
Department of Computer Science and Engineering
National Institute of Technology
Raipur, India
sarthakkhurana2@gmail.com

Keerthi Cheepurupalli
Department of Computer Science and Engineering
National Institute of Technology
Raipur, India
satyakeerthi1999@gmail.com

Thalendra Sahu
Department of Computer Science and Engineering
National Institute of Technology
Raipur, India
sthalendra@gmail.com

***Abstract*— With the popularization of social media and similar web-based applications, the information generated by the users of these applications is growing exponentially. Thus, more attention is required on the "recommender system". The recommender system fails to determine the real-time user interest. This paper uses an instant input mechanism to overcome this problem i.e. based on user interaction, a recommender system determines users' real-time interest. This paper uses "Significance analysis" to refine the recommender process performance. The most successful predictions are given to the client during communication to assess the real-time desires of the users and as soon as the model identifies the user it determines the expected recommendations for that user, thus, providing "Personalization of expectations." There are two problems in the recommender scheme "cold-start" or "potential-false termination". This problem is dealt using the user interaction which provides interactive recommendation although the system doesn't have any information related to the user. Potential-false dismissal occurs due to "tag sparsity". To overcome this problem, associations between various tags are taken into account, which refine the recommendations. The paper uses "low rank matrix factorization (LMF)" approach to determine the potential interests and personalize the recommendations during each iteration. This method overcomes the rating sparsity and is enhanced by embedding the similar user and resource information. For new users, real time input is used to recommend similar items.**

Keywords*— *Low rank matrix factorization (LMF), Singular Value Decomposition (SVD), Recommendation System.

I. Introduction

This research uses machine learning to build a custom movie scoring and recommendation framework based on previous movie reviews of the consumer. In movies, different people have different preferences, and that's not mirrored in a single score which is seen when a user search for a movie. The film scoring system helps users to quickly discover movies whatever their preferences may be. There are typically two types of existing recommender systems: content-based sorting and cooperative filtering. Among several method one method is tested in available venture, i.e. collaborative filtering. Upon going through some generic research papers it has been observed that collaborative filtering in terms of estimation error and computation time performs better than content-based filtering.

Collaborative sorting approaches measure distance interactions between users or objects are generally considered to be methods of neighbourhood, because of the nearness concept these methods are relying on. However, this methodology has two problems:

- Massive databases don't grow very well.
- Pure data-based methods becomes a practical issue.

For example, twenty from Chili Peppers songs are listened by one user and twenty other songs from Chili Peppers are listened by some other; there would be no difference between the actual user action matrixes. Mathematically, vectors' point product would be 0. It would be in completely separate neighbourhoods, although at least certain fundamental values appear pretty likely to be exchanged.

Using item features (such as genre) can aid, but not completely, fix this problem. Stealing an example from Joseph Konstan, regardless of the genre what if we both like great storytelling songs? How to solve this problem? The user needs a system that can extract raw vectors of taste and desired data. For cooperative sorting methods (neighbourhood models) based on raw information and low rank matrix factorization (LMF) [1] methods (factorization models) based on the latent features, it could make good suggestions.

IEEE ICSCAN 2020 ISBN 978-1-7281-6202-7

Recommenders of low-dimensional matrix try to capture the fundamental characteristics that motivate the raw data. From theoretical (like not a practical) perspective, this seems like the better approach to make recommendations dependent on the preferences of individuals. As Singular Value Decomposition (SVD) [2] can be directly estimated for gradient descent, this approach often applies significantly better for broader datasets. However, by using a lower-rank matrix, they are still likely to lose any important signals and although such approaches focused on factorization work extremely well, new methods are being investigated. Different types of probabilistic matrix factorization and many other methods emerged from these attempts.

II. Related Work

LMF [3] is an efficient and popular algorithm to make effective recommendations in real time applications. Recommendation Systems follow two approaches of filtering i.e., content-based and collaborative filtering [4]. These recommendation systems have gone through a major evolution [5] in the past decade. LMF is a matrix-based approach for performing collaborative filtering [6]. SVD [7] is a method of factoring any matrix. The result consists of three matrices; one out of them is a diagonal matrix. SVD is used to perform LMF in this research. Many approaches have evolved from SVD such as Accelerated SVD [8], Incremental SVD [9] and so on. Tag based analysis [10] is performed in some of the content-based recommender systems. But this analysis has a disadvantage of cold start; it is capable of providing recommendations for old users. User interaction is used for new users to avoid cold start. Genre is used as tag for indirect analysis to perform content-based filtering.

Cosine Similarity [11, 12] model is used by Google to make recommendations to its users. It is a model that uses simple mathematics to determine similarity index between various vectors using the cosine of angle between both. The disadvantages of this method are- it is not subjective to user interests and unsuitable for large datasets. There are hybrid recommendation algorithms [13, 14] which are generated by merging multiple algorithms. This research also uses hybrid algorithm involving LMF and Cosine Similarity algorithms. Evaluation of recommender system [15] is a controversial topic some of the researches involve computation of special parameters like precision, recall, and confusion matrix.

III. Proposed Work

A. Overview

The existing recommendation system uses the history of users to make a personalized recommendation which was sufficient when dealing with old users, but when the system encounters a new user it fails to make a recommendation to the user. This research proposes a model that gives special preference to the new users. A recommendation system has been developed following this approach. Generally, when someone talk about movie recommendations, the first thing that comes into the mind is the genre of the movie, and using the genres of movies it can easily recommend movies to a user, but the question is "Is it feasible?", Method developed in this paper try to answer these questions.

The proposed system mainly contains two methods:

1. Low Rank Matrix Factorization (for old users).
2. Popularity model (for new users).

These two models are divided into 11 functions. Out of these 5 are used for LMF and 6 are used for Popularity Model. The following are the functions used to provide recommendations to the old users using LMF preprocessing_for_old1(), preprocessing_for_old2(), lmf(), recommend_movies_old(), predictions_for_old(). The following functions used to provide recommendations to the new users using the popularity model preprocessing_for_new(), finding_similarity(), get_title_from_index(), get_index_from_title(), similar_movies(), recommend_movies_new(). Fig 1 represents the overall methodology behind the work. Fig 2 and Fig. 3 represents the functionalities of the functions.

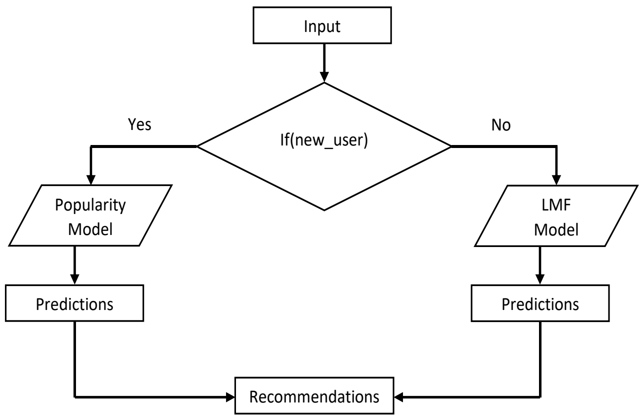

Fig. 1. Overall Methodology

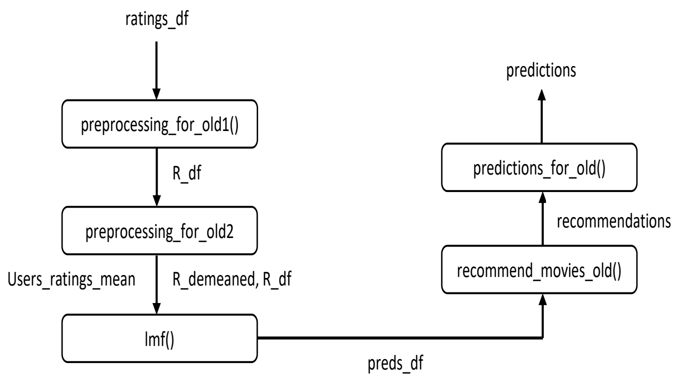

Fig. 2. LMF Model

ISBN 978-1-7281-6202-7

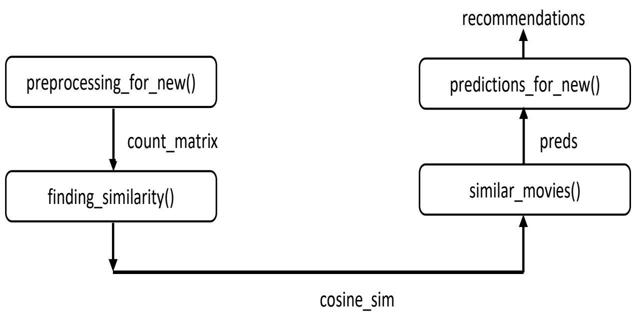

Fig. 3. Popularity Model

This research proposes a system that is a combination of two different models via, Low Rank Matrix Factorization and Popularity Model. This paper uses Low Rank Matrix Factorization for old users and popularity model for new users. This research uses this technique of Collaborative Filtering to make recommendations. This method falls under the category of Model based filtering. Many recommendation systems are built using other approaches from memory and model based filtering, but the problem occurs because some of memory based approaches cannot handle huge dataset, thus, generating scaling issues and some of the model based approaches can't handle sparse matrices, like in this case since most of the user-movie entry in the matrix is empty as in real life the user never rates all the movies, so, these approaches use features like genres to handle this issue, but is it enough? No, a method is needed that can determine preferences. and tastes from the raw data itself. This is where "Low Rank Matrix Factorization" comes into the picture. "Low Rank Matrix Factorization" is achieved using "Singular Value Decomposition". Generally, factorization of matrix is the process of determining multiple factors of matrix which are matrices as well. In other words, it is a process of breaking down of a matrix into simpler matrices. Suppose A is a matrix, then the following statement implies that B, C and D are its factors.

$$\mathbf{A = B.\ C.\ D} \quad \mathbf{(1)}$$

Singular value decomposition is an algorithm which can perform low rank decomposition of a matrix into simpler three matrices. Out of three, one is a diagonal matrix whereas others are unitary matrices.

$$\mathbf{R = U \sum V^t} \quad \mathbf{(2)}$$

Where, R represent matrix of user and ratings, U represent matrix of users and features, $\sum$ represent matrix of weights that is diagonal and V^t represent matrix of Movies and features respectively.

Since, the information about taste and preferences of user in unknown, the question arises "How this model is going to recommend movies to the user?" The answer to this question is the popularity model. The proposed model will prompt the user to enter its favourite movie and the model is then trained in accordance with that movie, since, the main goal of this study is to provide personalized recommendation to the users. Popularity model is a special method using which one can generate recommendation according to the interest of the users. For determination this model uses similarity index, which is calculated using vectors. In this model Cosine Similarity is used.

B. Cosine Similarity

Cosine similarity of two vectors is defined as the measure of cosine of angle between them. Thus, two vectors are said to be cosine similar if they overlap each other or the angle between them is "0 degrees". The closer the vectors are, the more similar they tend to be. The following is the relation for its calculation:

Suppose A and B are two vectors and the angle between them is α, then cosine similarity is given by

$$\mathbf{Cosine\ Similarity = \cos(\alpha) = A.B} \quad \mathbf{(3)}$$

Example- A graph with representation of two texts in vector form. Both of them would be equal if and only if the angle between them is zero. Thus, resulting cosine similarity will be 1.

C. Dataset Used

This research uses dataset from MovieLens. This dataset consists of 1 Million ratings (approximately). It consists of 4000 movies and 6000 users. The highest rating is 5 and the lowest rating is 1. The tabular form of few rows of the dataset is presented in Table 1 and Table 2.

TABLE 1. Contents of ratings.dat file from dataset

S.No	UserID	MovieID	Rating	Timestamp
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291

TABLE 2. Contents of movies.dat file

S.No	MovieID	Title	Genres
0	1	Toy Story (1995)	Animation|Children's|Comedy
1	2	Jumanji (1995)	Adventure|Children's|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama
4	5	Father of Bride Part II (1995)	Comedy

IV. RESULTS AND PERFORMANCE EVALUATION

Some developers believe that there doesn't exist any right way to evaluate the recommendation systems, their evaluation

ISBN 978-1-7281-6202-7

is dependent on their real-life use and the choice of users. e.g, a user in a movie recommendation system is recommended a movie (say) “Batman” based on the genres of movies he/she liked but this doesn’t mean this recommendation is correct, as it turns out the user only likes movies from Marvel Cinematic Universe and not from DC Universe.

Thus, evaluation of recommendation systems is meaningless until a user evaluates it. The “Netflix Prize” is a famous case of this scenario. Netflix Company promised to pay a million dollars to the team who would develop the best recommendation system for them; it had to be the robust solution. But as soon as the winners are declared, Netflix awarded them with the prize but it didn’t use their system at all because, the robustness of the system made it slow, thus it was never used by them in their websites. Recommendation systems cannot be evaluated in the normal way using normal parameters like precision, recall, etc. special metrics is used for this purpose.

A. *Metrics*

The metrics used for evaluation are different for both popularity model and LMF model. For popularity model two models will be compared: the old method to the new one. Precision, recall and f-score is calculated to evaluate the results.

B. *Popularity Model*

Traditional models determine the popular movie and then recommend those movies to the user. But these recommendations are not personalized. To overcome this drawback, popularity model based on user interaction has been used, which takes an input from user, e.g. favorite movie and then finds similar movies in the movies data frame. The recommendation is then made by selecting movies with higher similarity index, shown in Fig 4, i.e. it recommends the users with popular and similar movies in the frame.

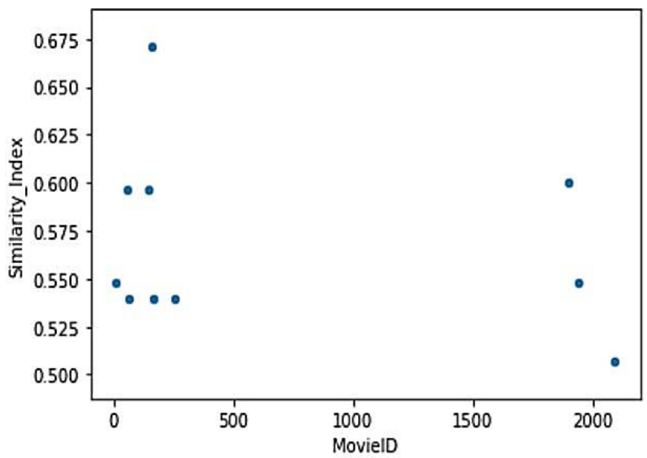

Fig. 4. MovieID vs Similarity Index

C. *Evaluation of LMF*

Tradition evaluation metrics like precision, recall, f-score can’t be used to evaluate recommendation systems. It will use special metrics like precision @ k, recall @ k, f-score@ k where k is the threshold rating. These metrics do not justify the correctness of the model. This research is conducted to develop a model, which is fully personalized i.e., consumer satisfaction. The analysis shows that this model needs higher number of ratings to recommend the user effectively. The correct evaluation is subjective to the interests of the user.

Precision @ k is defined as the number of movies that are recommended and are relevant to the number of movies that are recommended. The following formula can be used to calculate the precision @ k –

$$\textbf{Precision @ k} = \frac{|\textit{Recommended and Relevant}|}{|\textit{Recommended}|} \quad \textbf{(4)}$$

The results obtained by proposed model is represented in Fig. 5. This graph is plotted with x-axis as UserID and y-axis as Precision.

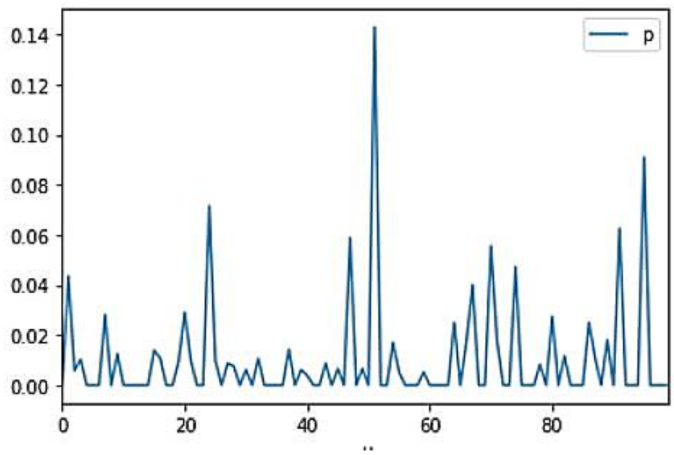

Fig. 5. UserID vs Precision

This graph is restricted to only 100 users. These values depend upon the number of ratings made by a user. Thus, using a bigger dataset can generate more precise results.

Recall @ k is defined as the number of movies that are recommended and are relevant to the number of movies that are relevant. The following formula can be used to calculate the recall @ k –

$$\textbf{Recall @ k} = \frac{|\textit{Recommended and Relevant}|}{|\textit{Relevant}|} \quad \textbf{(5)}$$

The recall obtained by proposed model is represented in Fig. 6. This graph is plotted with x-axis as UserID and y-axis as Recall. Just like precision, it also depends upon the size of dataset. Well dataset contains one million ratings but for a single user, values are low in number maximum to hundreds.

ISBN 978-1-7281-6202-7

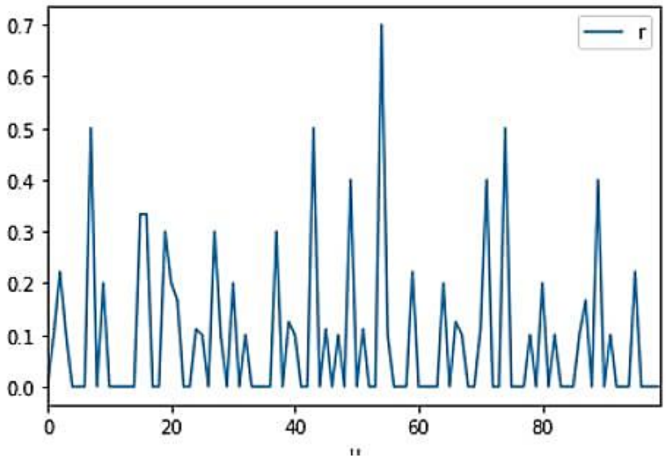

Fig. 6. UserID vs Recall

D. Results

Fig. 7 and Fig. 8 represents the results (predictions) obtained by using popularity and LMF model.

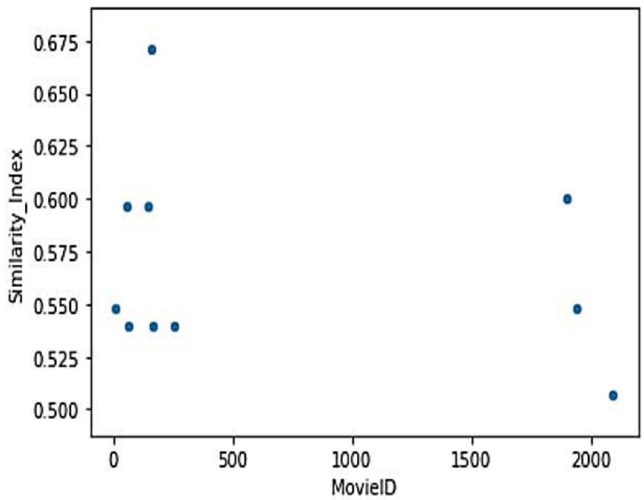

Fig. 7. Predictions using Popularity Model

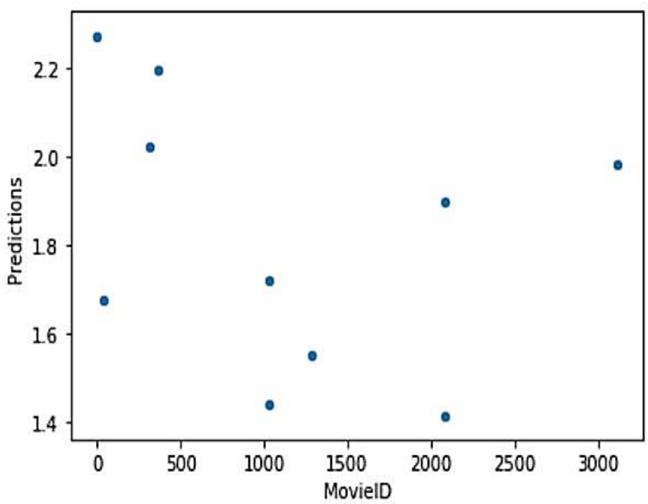

Fig. 8. Predictions using LMF

V. Conclusion

This paper presents a personalized recommendation model by using LMF and popularity model. This research is conducted to develop a model, which is fully personalized i.e., consumer satisfaction. The analysis shows that this model needs higher number of ratings to recommend the user effectively. The correct evaluation is subjective to the interests of the user. As the dataset consists of one million ratings but for a single user, values are low in number maximum to hundreds. So, the size of dataset is not good enough to explore each parameter. In near future, neural networks and deep-learning techniques can be used, some of which can generalize LMF algorithms via nonlinear neural architecture. Deep learning can be used in many different cases: social tagging, context-aware and sequence aware etc. Cosine similarity method do not work well and give accurate results when it doesn't have enough ratings for any movie or user's rating for movie is exceptionally very high or low. To improve the results for such low rated movies and users it can use adjusted cosine similarity to compute similarity. This recommendation system is not only bound for movies. It can be used anywhere like online music streaming sites and apps, online shopping stores, YouTube and Netflix like video streaming services, various news sites and social media apps like face book and Instagram, it can even have use on Google play store etc. Datasets can be continuously updated and thus making online actual rating recommendations to the users whose interests and habits are changing by day about films or any other item for which it is developed.

References

[1] Chen, B., Yang, Z., & Yang, Z, "An algorithm for low-rank matrix factorization and its applications", Neurocomputing, 275, pp. 1012-1020, 2018.

[2] Bokde, D., Girase, S., & Mukhopadhyay, D, "Matrix factorization model in collaborative filtering algorithms: A survey", Procedia Computer Science, 49, pp. 136-146, 2015.

[3] Xu, S., Zhang, C., & Zhang, J., "Bayesian deep matrix factorization network for multiple images denoising", Neural Networks, 2020.

[4] Katarya, R., & Verma, O. P., "An effective collaborative movie recommender system with cuckoo search", Egyptian Informatics Journal, 18(2), pp. 105-112, 2017.

[5] Sinha, B. B., & Dhanalakshmi, R., "Evolution of Recommender Paradigm Optimization over Time", Journal of King Saud University-Computer and Information Sciences, 2019.

[6] Ren, Y., Xu, J., Huang, J., & Chi, C., "Research on Collaborative Filtering Recommendation Algorithm for Personalized Recommendation System", 2019.

[7] Ranjan, A. A., Rai, A., Haque, S., Lohani, B. P., & Kushwaha, P. K., "An Approach for Netflix Recommendation System using Singular Value Decomposition", Journal of Computer and Mathematical Sciences, 10(4), pp. 774-779, 2019.

[8] Raghuwanshi, S. K., & Pateriya, R. K., "Accelerated singular value decomposition (asvd) using momentum based gradient descent optimization", Journal of King Saud University-Computer and Information Sciences, 2018.

[9] Zhou, X., He, J., Huang, G., & Zhang, Y., "SVD-based incremental approaches for recommender systems", Journal of Computer and System Sciences, 81(4), pp. 717-733, 2015.

[10] Xie, Q., Zhu, Y., Xiong, F., Li, L., Bao, Z., & Liu, Y., "Interactive resource recommendation with optimization by tag association and significance analysis", Neurocomputing, 2019.

ISBN 978-1-7281-6202-7

[11] Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X., "A new user similarity model to improve the accuracy of collaborative filtering", Knowledge-Based Systems, 56, pp. 156-166, 2014.

[12] Mahara, T., "A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment", Procedia Computer Science, 89, pp. 450-456, 2016.

[13] Tian, Y., Zheng, B., Wang, Y., Zhang, Y., & Wu, Q.,"College Library Personalized Recommendation System Based on Hybrid Recommendation Algorithm", Procedia CIRP, 83, pp. 490-494, 2019.

[14] Wang, Y., Deng, J., Gao, J., & Zhang, P., "A hybrid user similarity model for collaborative filtering", Information Sciences, 418, pp. 102-118, 2017.

[15] Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A., "Recommendation systems: Principles, methods and evaluation", Egyptian Informatics Journal, 16(3), pp. 261- 273, 2015.

 ISBN 978-1-7281-6202-7