Recommender Systems

Peng Xiong

***Abstract*—Recommendations have become a popular way of getting different opinions from people on things, i.e., movies, consumer products, dating, etc. Let us ask ourselves, can we make an automated system that can help with this issue? Recommendation systems have become the fore front of giving recommendations to people. These systems can utilize mathematical techniques/machine learning algorithms, such as matrix factorization, to give recommendations. The techniques of these mathematical models/machine learning techniques are giving these recommendation systems its capabilities to help society have more options. With this, recommendation systems are improving and starting to gain attraction from businesses to ordinary people.**

Contents

I. Introduction

When someone asks for recommendations on peculiar things, e.g., kitchenware, videogames, movies and tv series, they ask other peoples' opinion on their favorite recommendations. The excessive amount of data and internet usage has allowed e-commerce businesses, such as Amazon, Netflix, and Google, to utilize recommender systems to attract their users on their platforms in order to give the recommendations to them. The usage of these systems have many different approaches to for giving good recommendations.

A. Introduction to recommendation systems

Recommendation systems is an application of machine learning that helps users find what items the user likes. It utilizes data input from the user and gives an output of results. Recommendation systems utilize three models to execute its objectives: Content based filtering, Collaborative filtering (Memory based approach and Model based approach), and Hybrid filtering [12] [9]. Content based filtering uses data that is provided by the user and the description of the item which finds similar items relating to the user's original choice [12]. Collaborative filtering collects data from other users that are similar to the particular users' preferences [2]. Hybrid filtering is a combination of other filtering models.

B. Content-based Filtering

Content-based filtering is utilizing the user's input which finds items relating to the user's previous choices. This filtering model can provide the recommendations regardless of having no user ratings of the product. It only needs a single user in order to give recommendations. However, it does have drawbacks to giving recommendations. It needs enough ratings in order to give recommendations so that the model can give effective recommendations [12]. The authors imply a limitation where the implementation is difficult on larger database since users have different opinions on different items [12]. Content-based filtering is a simple solution to giving recommendations, but collaborative filtering can remedy some of those drawbacks.

C. Collaborative Filtering

Collaborative filtering takes data from other users' preferences and uses it to recommend other products to a particular user. It utilizes two approaches: memory and model. In A Comparative Study on Recommender Systems Approaches, the authors (Najmani, El habib, Sael, Zellou, 2019) explain that memory-based approach "find similar users based on cosine similarity or pearson correlation and take weighted avg. of ratings" (pg. 3) while model-based approach utilizes "different data mining, machine learning algorithms to find user ratings of unrated items, for example, PCA (Principal Component Analysis), SVD (Singular Value Decomposition), neural networks, matrix factorization, etc." (pg. 3). Comparing this to the content-based filtering, this would give serendipitous results to the users, but it does have drawbacks. Collaborative filtering needs a large number of users with similar interests to accurately give recommendations to the particular user. This problem is called cold start in recommendation system.

D. Hybrid Filtering

Hybrid filtering is a combination of two or more filtering approaches to increase the capabilities of the recommendation system. For instance, the combination of collaborative filtering and content-based filtering can increase the advantages such as improving the accuracy and efficiency of the system. It solves the key issues that are presented in content based filtering and collaborative filtering. On the other hand, there are drawbacks for its hybridization. By combining two or more filtering approaches, the hybrid filtering causes higher complexity with the recommendation system, and the calculation of the cost becomes more expensive [12].

E. Machine learning

The introduction of recommendation systems stems from the applications of machine learning (ML). In an application of machine learning, it utilizes a combination of mathematics, statistics, computer science, biology, psychology, etc. For instance, each filtering model (Content based filtering, Collaborative filtering, and Hybrid filtering) utilize a list of techniques to find recommendations for the user: TF-IDF (Term frequency-inverse document frequency), Clustering, Naïve bayes, Probability, KNN (k-nearest neighbors), SVD (Singular Value Decomposition), Jaccard index, Neural Networks, Matrix Factorization, Apriori algorithm, Vector distance algorithm, and PCA (Principal Component Analysis) [12].

II. Technical Analysis

People have a general understanding if someone were to explain what is a recommendation system, but would they be able to explain how it works? Recommendation systems, aka recommender systems, have a complex system and it is difficult to understand it. The system is based on different mathematical techniques and machine learning algorithms to find recommendations. We will dissect the mathematical techniques and machine learning algorithms and compare and contrast them. Recommender systems are useful in society, but the mathematics and machine learning algorithms are important to understand their usefulness.

A. Collaborative Filtering

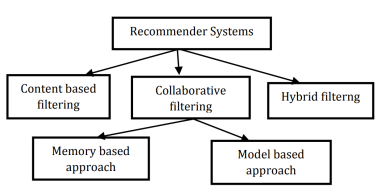

Fig. 1. This shows the recommendation system model. It is broken into three filtering methods where collaborative has two based approaches.

Collaborative filtering filters its recommendations on item-based recommendations [15]. In recommendation systems, there are three types of filtering: Content based filtering, Collaborative filtering, and Hybrid filtering [7]. We will focus on collaborative approach which takes two approaches: memory and model. The memory model utilizes a users based on cosine similarity or Pearson correlation:

$$similarity = \frac{A \times B}{||A|| \times ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (2)$$

This takes the weighted average of ratings while model base utilizes machine learning algorithms and mathematical techniques [12] [1]. We will be talking more about the model base approach where mathematics and machine learning will be introduced.

B. Matrix Factorization

One of the collaborative filtering methods used in the model-based approach is matrix factorization. Matrix factorization uses a matrix (an array that is a m×n where it is arrange by m rows and n columns) which the method uses two matrices where one matrices, matrix A, contains the users and the other one, matrix B, contains the ratings of the items [3] [18]. These two matrices are multiplied where the output is another matrices, matrix C, that is the output of the new ratings [14]. Matrix factorization can be difficult to understand in an abstract way, so we want to give an example to simplify it. The example that we will be using is recommending movies.

	Comedy	Action
Abagail	V	X
Becky	X	V
Cal	V	X
Derek	V	V
Erick	X	X

+

	Comedy	Action
M1	1	1
M2	3	2
M3	1	4
M4	3	2
M5	2	1
M6	1	4

	M1	M2	M3	M4	M5	M6
Abagail	1	3	1	3	2	1

Fig. 2. There are two matrix: matrix A (left) and matrix B (right). V represents what the users like and X represents what the users do not like. The columns for the movies show us the ratings that goes from 1 (bad) to 5 (good). Matrix C (bottom) is the factorization of both matrix A and matrix B.

In figure 2, we have two matrix: matrix A (represents the users) and matrix B (represents the movies). Matrix factorization takes a value from matrix A and matrix B and finds the results of each users favorite movie. Each movie is selected by the users favorite genre. For example, Abagail likes comedy which movies that have comedy are put into a new matrix.

By using matrix factorization on matrix A and matrix B, figure 3 gives us are output in matrix C. There are some exceptions that needs to be explained. For instance, Derek likes comedy and action which means that the ratings of comedy and action are added together. However, Erick does not like comedy nor action which means that no ratings need to be added. As you see, the results of matrix C shows the users ratings of each movie. We now have an understanding

	M1	M2	M3	M4	M5	M6
Comedy	1	3	1	3	2	1
Action	1	2	4	2	1	4

	Comedy	Action
Abagail	V	X
Becky	X	V
Cal	V	X
Derek	V	V
Erick	X	X

	M1	M2	M3	M4	M5	M6
Abagail	1	3	1	3	2	1
Becky	1	2	4	2	1	4
Cal	1	3	1	3	2	1
Derek	2	5	5	5	3	5
Erick	0	0	0	0	0	0

Fig. 3. This shows the results by using matrix factorization of both matrix A and matrix B which gives us matrix C.

of matrix factorization, but a question lingers: how do we know the factorization is correct of knowing that the users like comedy or action? Here, we start to apply the machine learning to this factorization.

C. Machine Learning

Matrix factorization utilizes a mathematical optimization where the machine finds the solution until it is close enough to the specific solution. The mathematical optimization that is used to find the solution is called Gradient Descent (a first-order iterative algorithm that finds the local minimum) [6]. This method can access through the same factorization process by comparing the matrix with random values to matrix C.

	M1	M2	M3	M4	M5	M6
F1	1.3	0.3	3.1	2.1	2.5	2.2
F2	0.9	2.3	4.6	0.4	1.1	3.5

	F1	F2
Abagail	0.5	0.1
Becky	0.22	0.8
Cal	0.42	0.18
Derek	0.30	0.67
Erick	0.71	0.12

	M1	M2	M3	M4	M5	M6
Abagail	0.98	0.17	2.01	1.09	1.36	1.45
Becky	1.0	1.90	4.32	0.78	1.43	3.28
Cal	0.70	0.53	2.12	0.95	1.24	1.55
Derek	0.64	1.55	4.01	0.89	1.45	3.00
Erick	1.03	0.48	2.75	1.53	0.90	1.98

	M1	M2	M3	M4	M5	M6
Abagail	1	3	1	3	2	1
Becky	1	2	4	2	1	4
Cal	1	3	1	3	2	1
Derek	2	5	5	5	3	5
Erick	0	0	0	0	0	0

Fig. 4. The picture shows an application of the mathematical optimization to finding the correct response. In the middle is a new matrix D.

In Gradient Descent, matrix A and matrix B have random values where they are hyper parameters (a controlling parameter where it controls the learning process of the test set). The values for matrix D are given by multiplying feature 1 (F1) and feature 2 (F2) from both the movies and the users from which they are added together.

$$(F1_1 \times F1_2) + (F2_1 \times F2_2) \tag{3}$$

Gradient Descent is applied by comparing the values to each column from matrix D to matrix C. For example, in figure 4, we compare Abagail's values from matrix D to matrix C. If one of the values in matrix D is not close enough to the expected value from matrix C, then the user can increase or decrease the values of F1 and F2 from both matrix A and matrix B. From there you get the results of the features that the user is going to like. From these results, matrix factorization and machine learning techniques like gradient descent gives us the recommended results to the user. We understand how the machine learning technique of gradient descent works, but how well does matrix factorization perform with recommendations?

D. Performance of Matrix Factorization

The factorization's performance has a beneficial cost. It has the ability to lower the number of recommendations that is given to the users. For instance, if the movies were just recommended to the user without matrix factorization, the storage would have too many recommendations that the filtering system can hold. We want to explain this effectiveness that matrix factorization can have with recommendations.

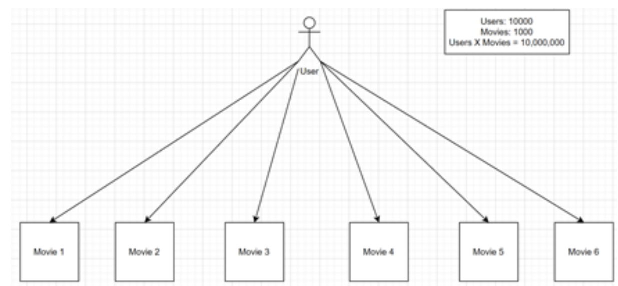

Fig. 5. The picture depicts a particular user in which it shows how many recommendations would be given.

In figure 5, we have a user that has 7 movies that can be recommended to them. By multiply the user and movies together, the storage would be 7. However, if we had more users and movies together, the total amount would be massive. Looking at the little example box, the number of parameters is massive which the storage can be an issue to the computation. In contrast, if we were to put features into the figure, the computation becomes less massive.

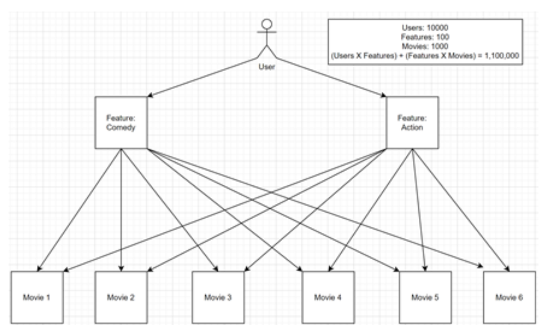

Fig. 6. This picture contains features which the computation becomes less massive.

Figure 6 shows a much better understanding of the filtering in which the model has a better storage amount than figure 5. Looking at the example box, the storage is big, but, when comparing it to the previous storage in figure 5, the storage is less massive and helps makes the recommendations less complex. This beneficial technique is a driving factor of why this technique is effective.

$$\overset{A}{\begin{bmatrix} 1 & 3 & 2 & 0 \\ 0 & 0 & 3 & 5 \\ 3 & 5 & 4 & 1 \\ 2 & 5 & 4 & 3 \end{bmatrix}} = \overset{U}{\begin{bmatrix} -0.30 & 0.27 & 0.28 & -0.86 \\ -0.32 & -0.89 & -0.20 & -0.23 \\ -0.61 & 0.35 & -0.69 & 0.10 \\ -0.65 & -0.01 & 0.62 & 0.42 \end{bmatrix}} \times \overset{\Sigma}{\begin{bmatrix} 11.23 & 0 & 0 \\ 0 & 5.10 & 0 \\ 0 & 0 & 0.48 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -0.30 & -0.64 & -0.59 & -0.37 \\ 0.25 & 0.49 & -0.15 & -0.81 \\ -0.68 & 0.57 & -0.40 & 0.20 \\ 0.61 & 0.10 & -0.68 & 0.38 \end{bmatrix}}$$

Fig. 7. The picture shows an example of SVD. Matrix A is the result of finding the three sub matrices where the results of the three sub matrices are compared to matrix A.

E. Singular Value Decomposition

Matrix factorization is a generalize factorization to finding recommendations, but their are other types of factorization that we want to explore. One type of matrix factorization that is a sub factorization is called the singular value decomposition (SVD). This method uses the original matrix which reduces it to three sub matrix: U, Sigma (Σ), and V^T. To put this in recommendation systems term, the original matrix is the $Data_m \times m$ where U is m×m matrix, Sigma (Σ) is $m \times n$ matrix that has all zeros besides the diagonal line which contains values that are called singular values (square roots of non-negative eigen values), and V^T is an $n \times n$ transpose matrix of V (an operator where it turns the original matrix to a diagonal matrix) [5] [4] [16].

$$Data_{m\times m} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n} \rightarrow A = U\Sigma V^T \tag{4}$$

The mathematics behind SVD is more complex on how it recommends the users, so we will break down the mathematics behind it. When given the original matrix, $Data_m \times m = A$, SVD will decompose A to find the three sub matrix. U is found by looking for the eigen vectors (special vectors that correlates to a linear system of equations) of the matrix of AA^T where the matrix A is multiplying to the transpose matrix of A, and, similarly, V is found by looking for the eigen vectors of the matrix of A^TA [16]:

$$U \approx AA^T = \begin{bmatrix} a_1 & \cdots & a_n \\ b_1 & \cdots & b_n \\ c_1 & \cdots & c_n \end{bmatrix} \times \begin{bmatrix} a_1 & b_1 & c_1 \\ \vdots & \vdots & \vdots \\ a_n & b_n & c_n \end{bmatrix}$$

$$V \approx A^TA = \begin{bmatrix} a_1 & b_1 & c_1 \\ \vdots & \vdots & \vdots \\ a_n & b_n & c_n \end{bmatrix} \times \begin{bmatrix} a_1 & \cdots & a_n \\ b_1 & \cdots & b_n \\ c_1 & \cdots & c_n \end{bmatrix}$$

With the eigen vectors, we find the eigen values (a special set of scalars (an element of a value that is defined under the vector space) that relates to a systems of linear equations) of U and V by using the formula to find eigen values:

$$A - \lambda I = 0 \tag{5}$$

For Σ, the values are found by square rooting the eigenvalues of either AA^T or A^TA (the results would be orthogonal). From here, we do the same process for A^TA, but we start using the singular values to row reduce each of the matrices. From there, we have the original that is decomposed through the singular value decomposition method . We went through the SVD process, and I will interpret the results of the matrix given to us.

Using figure 7 as are example, the results tells us the recommendations that are necessary to the users. Matrix A is the user to movies where each user has a rating from 1 to 5 for each movie. Matrix U is a user-to-concepts where concepts are the specified groups of what users like in terms to features. This can also be implied to matrix Σ and matrix V^T where matrix Σ implies to the strengths-to-concepts (finding the strongest similarities between the user and the movies with the genre or feature) and matrix V^T implies to the movie-to-concepts (finding the similarities between the movies to the genre or features). The original matrix is compared to each of the sub matrices and finds which concepts is the best for recommendation for the user. When you compare this to the matrix factorization, you can see a difference in its approach mathematically to finding the recommendations. It does not need to use an optimization in order for the developer to adjust values to find the correct percentage. However, Singular Value Decomposition needs to find the three sub matrices in order to find the best recommendations to the users. Knowing how singular value decomposition works, we can compare this method to matrix factorization.

F. Comparison and Contrast of Singular Value Decomposition and Matrix Factorization

Even though Singular Value Decomposition is a sub factorization of Matrix Factorization which uses the same concept to find the recommendations, the two have some similarities and differences on certain approaches. Starting off with the differences, a difference with Singular Value Decomposition and Matrix Factorization is how SVD and Matrix Factorization factorizes the matrices and interpret results. As it was explained in the previous paragraph, it takes the original matrix and finds the three matrices—U, Σ, and V^T—and analyzes each of the three matrices correlations, whereas Matrix Factorization has two matrices that utilizes machine learning techniques, such as Gradient Descent, to find the optimal solution of each feature to give to the user. On a learning and an implementation level, matrix factorization can be easily understanding and easy to implement whereas SVD utilizes a lot of linear algebra, such as eigen values and eigen vectors, which can be hard to understand. A similarity that both of these factorizations have is being able to factorize their matrices to find the optimal answer. Both of the factorizations are great mathematical techniques to finding the recommendations to the user.

G. Cold Start

When people hear about cold start, they usually think about cars that cannot start during sub zero cold temperatures. In recommendation systems, specifically in collaborative filtering, cold starts have similar concepts as the definition implies, but it is data that becomes the issue. We have talked about the in-depth mathematical techniques used in Collaborative Filtering. The data of users and items needs enough rating information in order to be effective in recommending [11]. To an item that has no ratings and to new users, it can be difficult to recommend.

H. Issue

Cold start is a data issue for the machine where it can not give effective recommendations to the user due to insufficient amount of information on any users. This issue is more directed to new users and new items [17]. This becomes an issue with the mathematical techniques because of insufficient information will not show the results [11]. Implied by the definition of cold start, it is impeded by the lack of ratings from each product that can not be recommended to the user. For instance, if a new shoe does not have any ratings, the new shoe will not be recommended to any user. The problem gets complex for the filtering model to find recommendations, but There are solutions that can help remedy the cold start issue

I. Solution

Many of the issues can be solve through different techniques within collaborative filtering. Having a enough knowledge of matrix factorization, there is a proposal of a solution that tries to remedy the issue. We looked at different proposals from each researchers that utilizes the same mathematical techniques and machine learning algorithms with different modifications.

Ocepek, Uroš, Jože Rugelj, and Zoran Bosnić (2015) propose a solution by imputing missing ratings to improve the matrix factorization performance [13]. They would impute the values of the attributes by looking for the mean value, the most frequent, and linear regression (a supervised learning model that predicts a value based on another value) and regression tree (a supervised learning method to base on making decisions) [13]. This approach would remedy the issue of cold start because it computes the distance of a new user and a user on how much similar interests that they have.

Another solution, similar to the previous paragraph, would be giving a survey to new users which would give information to the users and learning what they like in general [17]. For instance, in figure 8, a new user would be given a survey to fill out on their favorite interests which the machine would find items for the new user based on common interests with other users. This would allow the recommender system to have sufficient information on a new user. However, we solve an issue with a new user, but a survey would not be effective for a new item. Plus, there is a problem if the user does not actually do the survey. For instance, they do not consent to giving information on themselves or not wanting to waste or not having time enough time [20]. To the solve the issue, the surveys should be as simple as possible.

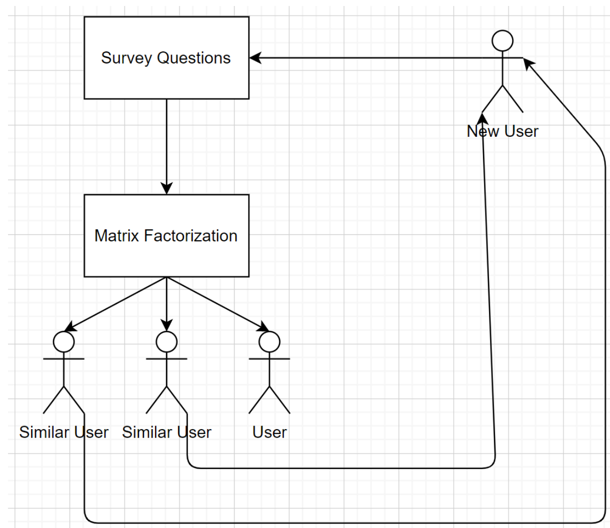

Fig. 8. The image shows a new user giving information about their interests. The information would be send through a model, i.e, matrix factorization, and find similar users to recommend items based on similar users.

III. The Future of Recommender Systems

The rise of the internet and the excessive amount of data has allowed more e-commerce and businesses to find ways to attract more people, and its popularity has gotten big that companies are finding ways to improve it. Within the next couple of years, it is becoming more utilize to have better recommendations. In this section, I want to show the trends that arising within recommendation systems.

A. Research

With the popularity gaining attention, recommender systems are improving each year with new research. Recommender system, as a whole, have different ways of being implemented, but the systems do have drawbacks in which there are solutions. Also, these filtering types can also be improved to optimize recommendations to other users. For instance, some areas of research consist of finding new ways how to recommend, indirect feed backs, hybridization of filtering methods, and full-page optimization [2]. Taking on how users choose to personalize their recommendations as an example, recommender systems recommend based on what things to recommend. Instead, it is about how the users should be recommended. For instance, taking the Netflix prize as an example, users will be presented with details, i.e., pictures, description, metadata, etc. about particular movies. The users can choose which details are presented to them to optimize their recommendations [2]. This example relates to an issue of avoiding low quality modeling in which it is finding quality recommendations.

B. Quantum Machine Learning

With quantum computation on the rise, machine learning is integrating its algorithms with quantum algorithms. With recommender systems being an application of machine learning, this will help optimize recommender systems to find recommendations with accuracy and precision. We will be explaining more about the machine learning and quantum computation.

Adding some context, quantum computation is utilizing quantum mechanics, such as superpositions (when two or more stimuli can be combined) and entanglement (when a particle is correlated to other particles), to perform computation where it utilizes quantum circuits which is based on quantum bits (qubit) [19] [8]. In contrast to classical computation, it utilizes programs which uses assemblers that is based on bits (binary number that are represented as a 0 or a 1). Quantum computation is unique because it utilizes algorithms that can process information faster than what a classical computer can compute. For instance, Shor's algorithm (a factorization algorithm that finds prime factors through fast multiplication) can be applied through computer science fields such machine learning.

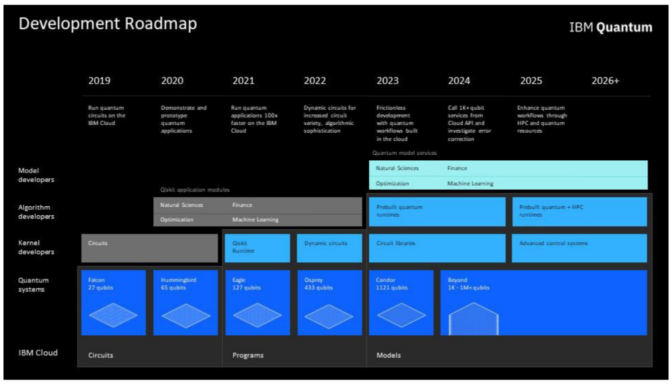

Fig. 9. The road map for IBM's quantum computation future in the next couple of years [10].

People might be asking themselves: how does this relate to recommender systems? As we said before, recommender systems are an application of machine learning, so quantum machine learning will give influence on the next generation of recommender systems. Also, we have seen an example of a machine learning algorithm that have been applied—Gradient Descent. These algorithms will be applied to quantum algorithms, such as Shor's, to optimize the recommendations. However, quantum machine learning is at its beginning phase, but there is more research through. For example, IBM's quantum computer is increasing its qubits and the applications within the next couple of years to optimize their systems for improvement [10]. This would mean that the improvement of different applications, such as recommender systems.

IV. Conclusion

We have understood the complex mathematical techniques and machine learning algorithms used to recommend things to people. Matrix Factorization, as well as SVD, is a useful tool when it comes to finding the optimal recommendations to the users. It is a useful tools used in collaborative filtering which can find optimal solutions. However, it does have its faults, such as cold starts, but we have shown that their are modifications to remedy this issue. The technical aspects of the collaborative filtering shows the complex models that can be complicated to understand, but, with a step-by-step process of understanding it, the models are not as intimidating as it seems. These filtering techniques has shown effective usage in being able to give the recommendation that it is supposed to give to the users. Technology is growing, and recommendation systems are growing increasingly popularity to helping e-commerce and other businesses to attract people with products.

References

[1] Neha Agarwal. Incremental associative memory model algorithm for highly scalable recommender systems. *International Journal of Computer Applications*, 68(21), 2013.

[2] Xavier Amatriain and Justin Basilico. Past, present, and future of recommender systems: An industry perspective. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 211–214, New York, NY, USA, 2016. Association for Computing Machinery.

[3] Dheeraj Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. Matrix factorization model in collaborative filtering algorithms: A survey. *Procedia Computer Science*, 49:136–146, 2015.

[4] Vito Xituo Chen and Tiffany Y. Tang. Incorporating singular value decomposition in user-based collaborative filtering technique for a movie recommendation system: A comparative study. In *Proceedings of the 2019 the International Conference on Pattern Recognition and Artificial Intelligence*, PRAI '19, page 12–15, New York, NY, USA, 2019. Association for Computing Machinery.

[5] Guan-Shen Fang, Sayaka Kamei, and Satoshi Fujita. Rating prediction with topic gradient descent method for matrix factorization in recommendation. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 8(12):469–476, 2017.

[6] Rainer Gemulla, Erik Nijkamp, Peter J. Haas, and Yannis Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 69–77, New York, NY, USA, 2011. Association for Computing Machinery.

[7] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[8] Weiwen Jiang, Jinjun Xiong, and Yiyu Shi. When machine learning meets quantum computers: A case study. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, ASPDAC '21, page 593–598, New York, NY, USA, 2021. Association for Computing Machinery.

[9] Mohamed Khoali, Abdelhak Tali, and Yassin Laaziz. Advanced recommendation systems through deep learning. In *Proceedings of the 3rd International Conference on Networking, Information Systems Security*, NISS2020, New York, NY, USA, 2020. Association for Computing Machinery.

[10] Kevin Krewell. Ibm reveals five year quantum development roadmap, Feb 2021.

[11] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, ICUIMC '08, page 208–211, New York, NY, USA, 2008. Association for Computing Machinery.

[12] Kawtar Najmani, Benlahmar El habib, Nawal Sael, and Ahmed Zellou. A comparative study on recommender systems approaches. In *Proceedings of the 4th International Conference on Big Data and Internet of Things*, BDIoT'19, New York, NY, USA, 2019. Association for Computing Machinery.

[13] Uroš Ocepek, Jože Rugelj, and Zoran Bosnić. Improving matrix factorization recommendations for examples in cold start. *Expert Systems with Applications*, 42(19):6784–6794, 2015.

[14] Li Pu and Boi Faltings. Understanding and improving relational matrix factorization in recommender systems. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, page 41–48, New York, NY, USA, 2013. Association for Computing Machinery.

[15] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, page 285–295, New York, NY, USA, 2001. Association for Computing Machinery.

[16] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth international conference on computer and information science*, volume 1, pages 27–8. Citeseer, 2002.

[17] Lei Shi, Wayne Xin Zhao, and Yi-Dong Shen. Local representative-based matrix factorization for cold-start recommendation. *ACM Trans. Inf. Syst.*, 36(2), August 2017.

[18] Harald Steck. Gaussian ranking by matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, page 115–122, New York, NY, USA, 2015. Association for Computing Machinery.

[19] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 217–228, New York, NY, USA, 2019. Association for Computing Machinery.

[20] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, page 315–324, New York, NY, USA, 2011. Association for Computing Machinery.

APPENDIX

My name is Peng Xiong and I am an upcoming graduating senior at the College of Saint Benedict's and Saint John's University (CSB/SJU). I am majoring in computer science with a minors in mathematics. My research paper is based on recommender system which was written based on my experiences on classes and lessons that I have learned over the years.

The classes that I took were MATH 338 (Numerical Analysis), MATH 239 (Linear Algebra), MATH 119 (Calculus 1), MATH 120 (Calculus 2), CSCI 317H (Artificial Intelligence), FYS (First Year Seminar), etc. Some courses that I took were not related to computer science and mathematics, but they helped improve my writing skills for this paper.

The courses, that are not computer science and mathematics, had a stronger impact on understanding my topic. Recommender systems has other focuses on society besides computer science and mathematics. For instance, my introduction to anthropology helped me understand cultural understanding with the combination of science and sociology. This major experience opened my mind in different ways to learning to see how computer science can impact society, such as recommender systems.

The courses for mathematics and computer science had necessary information that I needed to understand recommender systems. For instance, Numerical Analysis introduce me SVD and Matrix Factorization where we learned how to apply it, and the optimizations, such as gradient descent, that are used in machine learning algorithms. Plus, Calculus 1 and Calculus 2 also helped me understand Numerical Analysis because optimizations was based on calculus.

Recommender systems have allowed me to apply my liberal arts education to an extent. I have reflected on the certain courses that allowed me to dive deeper into this topic. Plus, it allowed me to give better demonstration of writing a better paper. For instance, in FYS, I did a paper on the ethics of Artificial Intelligence, but it was only a generalizing paper which dived into some ethical parts without any technical paper. However, the class gave me the foundations of writing a paper, looking for peer review sources, and correcting my mistakes.