RECOMMENDATION SYSTEMS USING COLLABORATIVE FILTERING

* **BY GROUP 18**

1st **Kadiyala Monish Mithra**
dept. of CSE. (Artificial Intelligence)
AMRITA VISHWA VIDYAPEETHAM
AM.EN.U4AIE20038
KOLLAM , KERALA

2nd **Konijeti Sri Vyshnavi**
dept. of CSE. (Artificial Intelligence)
AMRITA VISHWA VIDYAPEETHAM
AM.EN.U4AIE20042
KOLLAM , KERALA

3rd **Seshabhattar Sri Vastav**
dept. of CSE. (Artificial Intelligence)
AMRITA VISHWA VIDYAPEETHAM
AM.EN.U4AIE20065
KOLLAM , KERALA

4th **Srungarapu Jatin Abit Sai**
dept. of CSE. (Artificial Intelligence)
AMRITA VISHWA VIDYAPEETHAM
AM.EN.U4AIE20069
KOLLAM , KERALA

5th **Yerramreddy Dhanvanth Reddy**
dept. of CSE. (Artificial Intelligence)
AMRITA VISHWA VIDYAPEETHAM
AM.EN.U4AIE20077
KOLLAM , KERALA

***Abstract*—We are living in a digital world . Now a days online platforms like Amazon prime, Netflix etc have been increased for movies , web series etc.**

Everyone loves movies irrespective of age, gender, race, color, or geographical location. We all in a way are connected to each other via this amazing medium. Yet what most interesting is the fact that how unique our choices and combinations are in terms of movie preferences. Some people like genre-specific movies be it a thriller, romance, or sci-fi, while others focus on lead actors and directors. When we take all that into account, it's astoundingly difficult to generalize a movie and say that everyone would like it. But with all that said, it is still seen that similar movies are liked by a specific part of the society.

So here's where we as data scientists come into play and extract the juice out of all the behavioral patterns of not only the audience but also from the movies themselves. So without further ado let's jump right into the basics of a recommendation system.

Practically, recommender systems encompass a class of techniques and algorithms which are able to suggest "relevant" items to users. Ideally, the suggested items are as relevant to the user as possible.

Items are ranked according to their relevancy, and the most relevant ones are shown to the user. The relevancy is something that the recommender system must determine and is mainly based on historical data.

Recommender systems are generally divided into two main categories: collaborative filtering and content-based systems.

Collaborative Filtering is the most common technique used when it comes to building intelligent recommender systems that can learn to give better recommendations as more information about users is collected.

***Index Terms*—Recommender Systems , Collaborative Filtering**

I. INTRODUCTION

A Recommender System predicts the likelihood that a user would prefer an item. Based on previous user interaction with the data source that the system takes the information from (besides the data from other users, or historical trends), the system is capable of recommending an item to a user. Think about the fact that Amazon recommends you books that they think you could like; Amazon might be making effective use of a Recommender System behind the curtains. This simple definition, allows us to think in a diverse set of applications where Recommender Systems might be useful. Applications such as documents, movies, music, romantic partners, or who to follow on Twitter, are pervasive and widely known in the world of Information Retrieval.

Such amazing applications, carry a huge amount of theory behind them. While theory can be a little bit intimidating and dry, basic understanding of data structures, a programming language, and a little bit of abstraction is all you need to understand the basics of recommender systems.

II. WHAT ARE RECOMMENDER SYSTEMS?

Recommender systems aim to predict users' interests and recommend product items that quite likely are interesting for them. They are among the most powerful machine learning systems that online retailers implement in order to drive sales.

Data required for recommender systems stems from explicit user ratings after watching a movie or listening to a song, from implicit search engine queries and purchase histories, or from other knowledge about the users/items themselves.

Essentially, a Recommendation System is a filtration program whose great objective is to anticipate the "rating" or "inclination" of a client towards an area explicit thing or thing. For our situation, this space explicit thing is a film, thusly the fundamental focal point of our suggestion framework is to channel and foresee just those motion pictures which a client would favor given a few information about the client oneself.

III. Why do we need recommender systems?

We now live in what some call the “era of abundance”. For any given product, there are sometimes thousands of options to choose from. Think of the examples above: streaming videos, social networking, online shopping; the list goes on. Recommender systems help to personalize a platform and help the user find something they like.

The easiest and simplest way to do this is to recommend the most popular items. However, to really enhance the user experience through personalized recommendations, we need dedicated recommender systems.

Companies using recommender systems focus on increasing sales as a result of very personalized offers and an enhanced customer experience.

Recommendations typically speed up searches and make it easier for users to access content they’re interested in, and surprise them with offers they would have never searched for.

What is more, companies are able to gain and retain customers by sending out emails with links to new offers that meet the recipients’ interests, or suggestions of films and TV shows that suit their profiles.

The user starts to feel known and understood and is more likely to buy additional products or consume more content. By knowing what a user wants, the company gains competitive advantage and the threat of losing a customer to a competitor decreases.

Providing that added value to users by including recommendations in systems and products is appealing. Furthermore, it allows companies to position ahead of their competitors and eventually increase their earnings.

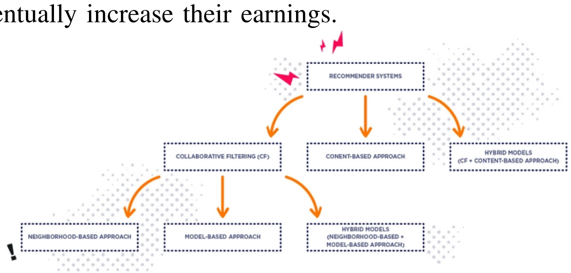

Fig. 1. Types Of Recommender Systems.

IV. How does a recommender system work?

Recommender systems function with two kinds of information:

- Characteristic information:
  - This is information about items (keywords, categories, etc.) and users (preferences, profiles, etc.).
- User-item interactions:
  - This is information such as ratings, number of purchases, likes, etc.

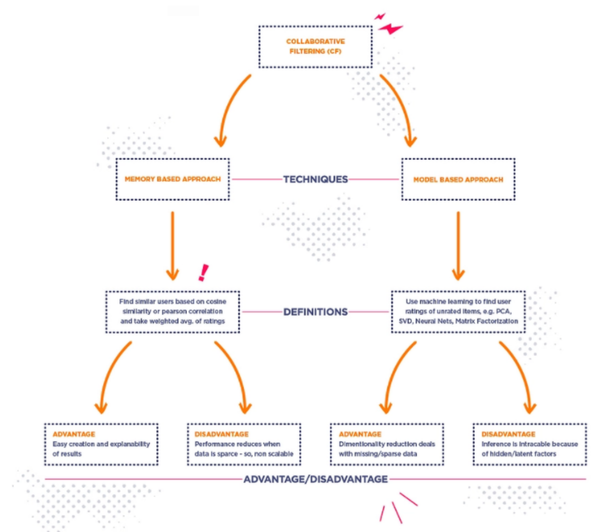

Fig. 2. Types Of Collaborative Filtering

Based on this, we can distinguish between three algorithms used in recommender systems:

- Content-based systems :
  - which use characteristic information.
- Collaborative filtering systems:
  - which are based on user-item interactions.
- Hybrid systems :
  - which combine both types of information with the aim of avoiding problems that are generated when working with just one kind.

A. Content-based systems

These systems make recommendations using a user’s item and profile features. They hypothesize that if a user was interested in an item in the past, they will once again be interested in it in the future. Similar items are usually grouped based on their features. User profiles are constructed using historical interactions or by explicitly asking users about their interests. There are other systems, not considered purely content-based, which utilize user personal and social data.

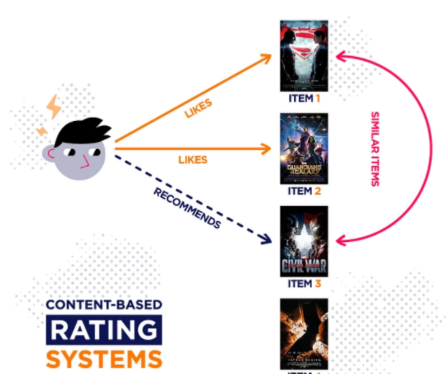

Fig. 3. Content Based Rating Systems

One issue that arises is making obvious recommendations because of excessive specialization (user A is only interested in categories B, C, and D, and the system is not able to recommend items outside those categories, even though they could be interesting to them).

Another common problem is that new users lack a defined profile unless they are explicitly asked for information. Nevertheless, it is relatively simple to add new items to the system. We just need to ensure that we assign them a group according to their features.

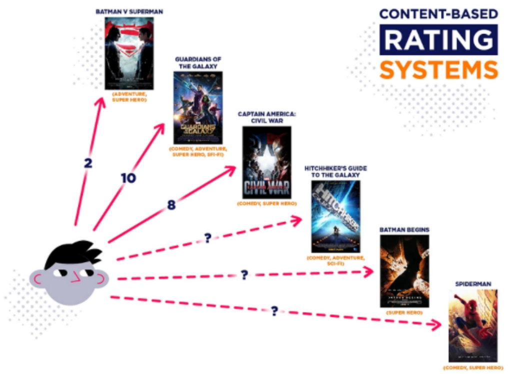

Fig. 4. Content Based Rating Systems

B. *Collaborative Filtering Systems*

Collaborative filtering is currently one of the most frequently used approaches and usually provides better results than content-based recommendations. Some examples of this are found in the recommendation systems of Youtube, Netflix, and Spotify.

These kinds of systems utilize user interactions to filter for items of interest. We can visualize the set of interactions with a matrix, where each entry (i, j)(i,j) represents the interaction between user ii and item jj. An interesting way of looking at collaborative filtering is to think of it as a generalization of classification and regression. While in these cases we aim to predict a variable that directly depends on other variables (features), in collaborative filtering there is no such distinction of feature variables and class variables.

In short, collaborative filtering systems are based on the assumption that if a user likes item A and another user likes the same item A as well as another item, item B, the first user could also be interested in the second item. Hence, they aim to predict new interactions based on historical ones. There are two types of methods to achieve this goal: memory-based and model-based.

V. Collaborative Filtering

Collaborative filtering needs a set of items that are based on the user's historical choices. This system does not require a good amount of product features to work. An embedding or feature vector describes each item and User, and it sinks both the items and the users in a similar embedding location. It creates enclosures for items and users on its own.

Other purchaser's reactions are taken into consideration while suggesting a specific product to the primary user. It keeps track of the behavior of all users before recommending which item is mostly liked by users. It also relates similar users by similarity in preference and behavior towards a similar product when proposing a product to the primary customer.

Two sources are used to record the interaction of a product user. First, through implicit feedback, User likes and dislikes are recorded and noticed by their actions like clicks, listening to music tracks, searches, purchase records, page views, etc.

On the other hand, explicit feedback is when a customer specifies dislikes or likes by rating or reacting against any specific product on a scale of 1 to 5 stars. This is direct feedback from the users to show like and dislike about the product. It includes both positive and negative feedback.

Collaborative Filtering is the most famous application suggestion engine and is based on calculated guesses; the people who liked the product will enjoy the same product in the future. This type of algorithm is also known as a product-based collaborative shift. In this Filtering, users are filtered and associated with each User in place of items. In this system, only users' behavior is considered. Only their content and profile information is not enough. The User giving a positive rating to products will be associated with other User's behavior giving a similar rating.

A. *Types of Collaborative Filtering*

There are two types of the collaborative filtering process:

1) Memory-based collaborative filtering
2) Model-based collaborative filtering

1) Memory-based Collaborative Filtering Memory-based CF is one method that calculates the similarity between users or items using the user's previous data based on ranking. The main objective of this method is to describe the degree of resemblance between users or objects and discover homogenous ratings to suggest the obscured items. Memory-based CF consist of the following two methods:
   1) User-based Collaborative Filtering: In this method, the same user who has similar rankings for homogenous items is known. Then point out the user's order for the item to which the user is never linked.
   Let's understand this with an example. Consider Harry and Jack are given ranking on few movies:
   Harry: Toy Story= 4, Coco= 2, Zootopia=3.
   Jack: Toy Story= 4, Coco= 2, Zootopia=?
   Now we need to find out the rating of Zootopia, which Jack has never viewed. For this, we need to follow the given steps:
   Identify the target user (according to this example, Jack is the target user) Find the same user who has ratings like the target user. Explore the interacted items. Forecast the ranking of unseen things of the target user. If the forecasted rankings are higher than the threshold, then suggest them to the target user.

2) Item-based Collaborative Filtering: In item-based CF, we find the same items that the target user has already viewed.

Jack: finding nemo=4, Moana= 3, Toy Story=4.

Identify the target user. Find the matched items which have the same ratings as items the target user rated. Forecast the rankings for the same items. If the forecasted rankings are higher than the threshold, then suggest them to the target user. Though, the Item-based model shows better consequences as compared to the user-based method as the resemblance between items seems to be consistent than the users.

A numerical measure using a similarity matrix is the most common technique. It involves Dot product, Cosine similarity, Pearson similarity, and Euclidean distance.

2) Model-based Collaborative Filtering

Model-based collaborative filtering is not required to remember the based matrix. Instead, the machine models are used to forecast and calculate how a customer gives a rating to each product. These system algorithms are based on machine learning to predict unrated products by customer ratings. These algorithms are further divided into different subsets, i.e., Matrix factorization-based algorithms, deep learning methods, and clustering algorithms.

Normally, the simple cluster algorithm is used like K-Nearest Neighbor to identify the nearest embedding or neighbor consisting of a similar matrix used for a product or a customer embedding. The matrix factorization technique is different from analyzing and exploring the rate of rating matrix in an algebra context and has two main goals. First, the initial ambition is to reduce the rating matrix dimension. This approach's second ambition is to identify perspective features under the rating matrix, which will provide several recommendations.

In Collaborative Filtering, two more frequent techniques are used. The model-based technique applies a statistics system and machine learning approach for minimizing the rating matrix. Still, the model-based approach does not produce expected results compared to CBF and CF approaches. An extensive database can be handled and infrequent matrices.

Collaborative Filtering is a straightforward interpretation of how these algorithms use crowd data. A large amount of data is gathered from different people and used for creating customized suggestions and preferences of a single user. These methods were developed in the 1990s and 2000s. Social media has brought innovation, and data availability has increased access to information from different sources. The recommended system has begun to use the social network in account in inclusion to similarity.

VI. Examples Of Collaborative Filtering

One of the best examples of collaborative filtering can be seen in the area of E-Commerce. When you browse an e-commerce website, you can see that it shows some recommended products to you. Some of the items there are precisely the same as what you were looking for. Now a question may arise about how the website knows what your interests are. It is all just because of collaborative filtering.

In social connection sites, Friends' suggestion is also very common. For example, on Facebook, a section is displayed known as People you may know; it is a very outstanding feature and shows a list of people to add them as a friend. Based on social connection data, this system educates and guesses the missing edges, like if you are friends with 10th out of 11th densely associated people, it is like you must befriend with 11th. Social connections are built by using the algorithms of collaborative Filtering.

Let's take one more example. Bob and Alice have the same interest in playing. Bob played it and enjoyed the game a lot. Alice did not play that game yet, but the system has determined that Bob and Alice have the same interest, so the system recommends that game to Alice. Collaborative Filtering can be performed by recommender systems using the same product. The other buyer will like the same item.

One more example of n X m this matrix is made up of the buyer's rating n refer to buyer and m refer to the item or object. Every element of this matrix is (k,l) how User l rated product k. We are dealing with movie's show ratings, and every rating should be several between 1 to 5 where follow 1-star rating to 5 stars rating. If the User did not rate a particular movie or the movie l is rated by user k.

One scenario of collaborative filtering is to suggest famous and interesting or popular information judged by the area of people. Then, the stories are shown on the front page of Reddit, which are voted positively by a group of people. As the group of people becomes more diversified, the publicized stories will show a better community interest. Wikipedia is also an application of collaborative Filtering.

Collaborative filtering does not need content extraction and analysis. People will be able to evaluate information accurately as compared to counted functioning. Complex objects or multimedia like music, movies, and images start working well.

VII. Collaborative Filtering vs Content-based Filtering

A. Advantages

A number of advantages are provided by Collaborative Filtering over Content-Based filtering. Some of them are:

- For telling the whole story, the item's content is unnecessary, like movie genre/ type.
- If the information of a product is not available, the product can be rated easily without delay in buying the product.
- Content-focused does not give any adaptability to the user's preferences and aspects.
- Collaborative filtering relies on other buyer's ratings to identify the connections between the buyers and provide the best suggestion based on the user's similarities. As

a comparison, the Content-based method just needs to analyse the user's profile and items.
- Collaborative filtering gives suggestions because most of the unknown buyers have a similar taste to you. Still, in Content-based, you will get the recommendations of items based on product features.
- In contrast to Collaborative filtering, new products are suggested without any specifications by many buyers.
- The cold start is the main problem of Content-based, and it arises when the recommendation system is made up of very few rating records. In this case, content-based filtering is an excellent alternative to this problem.
- Content-based has drawbacks, like the keyword used in the content for representing the item may be not representative. This approach also suffers in making perfect recommendations to the buyers with the very ratings.

B. *Disadvantages*

Numbers of the drawback of these systems are mentioned below.

- The content-based system is only a design suggestion based on the current interest of the user. Therefore, you can also say that this system is only limited to buyers' existing desires or interests.
- Since the item representation of the features is hand-setup comparatively, it requires enough domain knowledge; therefore, this model is only with the excellent hand setup features.
- If the content of the product is not good enough to describe the product precisely so the made recommendation will be false at the end
- The content-based approach provides a limited amount of innovation since the item and profile features should be matched. You need to be surprised by an excellent content-based filtering method.
- The system's correct recommendation cannot be provided unless strong user profile information is put in the system.
- There are many pros and cons of every system, whether a content-based filtering system or a collaborative filtering system. As a result, many organizations have adopted a hybrid system to merge the advantages of these systems, as mentioned earlier, and try their best to provide more accessible and more accurate suggestions to their users.

VIII. Advancements

There are many pros and cons of every system, whether a content-based filtering system or a collaborative filtering system. As a result, many organizations have adopted a hybrid system to merge the advantages of these systems, as mentioned earlier, and try their best to provide more accessible and more accurate suggestions to their users.

A. *Hybrid Filtering*

A hybrid approach is a mixture of collaborative and content-based filtering methods while making suggestions; the film's context also considers. The user-to-item relation and the user-to-user relation also play a vital role at the time of the recommendation. This framework gives film recommendations as per the user's knowledge, provides unique recommendations, and solves a problem if the specific buyer ignores relevant data. The user's profile data is collected from the website, film's context also considers the user's watching film and the data of the scores of the movie.

The data consist of aggregating similar calculations. This method is called the hybrid approach, in which both methods are used to produce the results. When this system is compared with other approaches, this system has higher suggestions accuracy. The main reason is the absence of information about the filtering's domain dependencies and the people's interest in a content-based system.

When these two approaches work together, you will get more knowledge, leading to better results; it explores the new paths to significant underlying content and collaborative filtering methods with buyer behavior data.

This system has taken to implement both the systems and overcome most of the weaknesses of each system's algorithms and improves the system's performance. Classification and cluster techniques are used for getting more excellent recommendations, thus growing accuracy and precision. Our method can be lengthier than other rules to recommend video, song, newsbooks, venue, e-commerce site, tourism, etc.

IX. DataSet

The topic collaborative Filtering deals with the users , items and the result prediction using various methods. We are creating a recommender system for giving recommendations on movies watched by various users.So we are including a matrix taking users and movies of dimensions **943** and **1682** which gives the ratings of various users to given movies and which are not been rated are taken as zero.So that the collaborative filtering model will be taking it for predicting the rating.The predicted rating will match approximately but not perfectly, the difference in the approximation is taken as the accuracy on the model we designed.

X. Methods

A. *Alternating Least Squares*

Alternating Least Squares (ALS) is a the model we'll use to fit our data and find similarities. But before we dive into how it works we should look at some of the basics of matrix factorization which is what we aim to use ALS to accomplish.

1) Matrix factorization: The idea is basically to take a large (or potentially huge) matrix and factor it into some smaller representation of the original matrix. You can think of it in the same way as we would take a large number and factor it into two much smaller primes. We end up with two or more lower dimensional matrices whose product equals the original one. When we talk about collaborative filtering for recommender systems we want to solve the problem of our original matrix having millions of different dimensions, but our "tastes" not being nearly as complex. Even if i've viewed hundreds of items they might just express a couple of different tastes. Here we can actually use matrix factorization to mathematically reduce the dimensionality of our original "all users by all items" matrix into something much smaller that represents "all items by some taste dimensions" and "all users by some taste dimensions". These dimensions are called latent or hidden features and we learn them from our data. Doing this reduction and working with fewer dimensions makes it both much more computationally efficient and but also gives us better results since we can reason about items in this more compact "taste space". If we can express each user as a vector of their taste values, and at the same time express each item as a vector of what tastes they represent. You can see we can quite easily make a recommendation. This also gives us the ability to find connections between users who have no specific items in common but share common tastes.

If we can express each user as a vector of their taste values, and at the same time express each item as a vector of what tastes they represent. You can see we can quite easily make a recommendation.

2) Matrix factorization Implicit data: There are different ways to factor a matrix, like Singular Value Decomposition (SVD) or Probabilistic Latent Semantic Analysis (PLSA) if we're dealing with explicit data. With implicit data the difference lies in how we deal with all the missing data in our very sparse matrix. For explicit data we treat them as just unknown fields that we should assign some predicted rating to. But for implicit we can't just assume the same since there is information in these unknown values as well. As stated before we don't know if a missing value means the user disliked something, or if it means they love it but just don't know about it. Basically we need some way to learn from the missing data. So we'll need a different approach to get us there.

3) Back to ALS: ALS is an iterative optimization process where we for every iteration try to arrive closer and closer to a factorized representation of our original data.

We have our original matrix R of size u x i with our users, items and some type of feedback data. We then want to find a way to turn that into one matrix with users and hidden features of size u x f and one with items and hidden features of size f x i. In U and V we have weights for how each user/item relates to each feature. What we do is we calculate U and V so that their product approximates R as closely as possible: R is congruent to U x V.

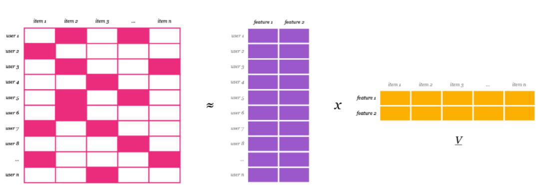

Fig. 5. Alternating Least Squares matrix selection process

By randomly assigning the values in U and V and using least squares iteratively we can arrive at what weights yield the best approximation of R. The least squares approach in it's basic forms means fitting some line to the data, measuring the sum of squared distances from all points to the line and trying to get an optimal fit by minimising this value.

With the alternating least squares approach we use the same idea but iteratively alternate between optimizing U and fixing V and vice versa. We do this for each iteration to arrive closer to R = U x V.

The approach we're going to use with our implicit dataset is the one outlined in Collaborative Filtering for Implicit Feedback Datasets by Hu, Korenand and Volinsky (and used by Facebook and Spotify). Their solution is very straight forward so i'm just going to explain the general idea and implementation but you should definitely give it a read.

Their solution is to merge the preference (p) for an item with the confidence (c) we have for that preference. We start out with missing values as a negative preference with a low confidence value and existing values a positive preference but with a high confidence value. We can use something like play count, time spent on a page or some other form of interaction as the basis for calculating our confidence.

- We set the preference (p):

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases}$$

Fig. 6. Preferences

  Basically our preference is a binary representation of our feedback data r. If the feedback is greater than zero we set it to 1.
- The confidence (c) is calculated as follows:

$$c_{ui} = 1 + \alpha r_{ui}$$

Fig. 7. Confidence

Here the confidence is calculated using the magnitude of r (the feedback data) giving us a larger confidence the more times a user has played, viewed or clicked an item. The rate of which our confidence increases is set through a linear scaling factor alpha. We also add 1 so we have a minimal confidence even if alpha x r equals zero. This also means that even if we only have one interaction between a user and item the confidence will be higher than that of the unknown data given the alpha value.

The goal now is to find the vector for each user (xu) and item (yi) in feature dimensions which means we want to minimize the following loss function:

$$\min_{y_*, y_*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \| x_u \|^2 + \sum_i \| y_i \|^2 \right)$$

Fig. 8. minimum loss of function

As the paper notes, if we fix the user factors or item factors we can calculate a global minimum. The derivative of the above equation gets us the following equation for minimizing the loss of our users:

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u)$$

Fig. 9. Derivative of the above equation gets us the following equation for minimizing the loss of our users

And this for minimizing it for our items:

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i)$$

Fig. 10. Minimizing items

One more step is that by realizing that the product of Y-transpose, Cu and Y can be broken out as shown below:

$$Y^T C_u Y = Y^T Y + Y^T (C_u - I) Y$$

Fig. 11. Product

Now we have Y-transpose-Y and X-transpose-X independent of u and i which means we can precompute it and make the calculation much less intensive. So with that in mind our final user and item equations are:

$$x_u = (Y^T Y + Y^T (C^u - I) Y + \lambda I)^{-1} Y^T C^u p(u)$$

$$y_i = (X^T X + X^T (C^i - I) X + \lambda I)^{-1} X^T C^i p(i)$$

- X and Y: Our randomly initialized user and item matricies. These will get alternatingly updated.
- Cu and Ci: Our confidence values.
- lamda: Regularizer to reduce overfitting (we're using 0.1).
- p(u) and p(i): The binary preference for an item. One if we know the preference and zero if we don't.
- I (eye): The identity matrix. Square matrix with ones on the diagonal and zeros everywhere else.

By iterating between computing the two equations above we arrive at one matrix with user vectors and one with item vectors that we can then use to produce recommendations or find similarities.

4) Similar items: To calculate the similarity between items we compute the dot-product between our item vectors and it's transpose. So if we want artists similar to say Joy Division we take the dot product between all item vectors and the transpose of the Joy Division item vector. This will give us the similarity score:

$$score = V \cdot V_i^T$$

Fig. 12. Product

5) Making recommendations: To make recommendations for a given user we take a similar approach. Here we calculate the dot product between our user vector and the transpose of our item vectors. This gives us a recommendation score for our user and each item:

$$score = U_i \cdot V^T$$

Fig. 13. Recommendation

XI. Conclusion

Online buyers and internet users crave personalized experiences. Most users prefer to use recommendations suggested by a different website to save their time because they do not want to waste their precious time searching and getting lost in information. As this trend is evolving, more organizations are using different recommender systems to personalize their business deals.

Implementing a recommender system can be expensive, but surely, you will get the benefits of highly customized content. This creates a unique stickiness to the product offering creating an invisible pull in customers that benefits the organizations.

XII. References

1) https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe
2) https://medium.com/ai-society/a-concise-recommender-systems-tutorial-fa40d5a9c0fa
3) https://tryolabs.com/blog/introduction-to-recommender-systems/
4) https://realpython.com/build-recommendation-engine-collaborative-filtering/what-is-collaborative-filtering
5) https://www.analyticsvidhya.com/blog/2020/11/create-your-own-movie-movie-recommendation-system/
6) https://towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109
7) https://www.kdnuggets.com/2019/09/machine-learning-recommender-systems.html