Q1. Find out the frequent datasets with the help of PCY Algorithm and given Data?

Transaction

T1 ----> {1,3,4}

T2 ----> {2,4,5}

T3 ----> {3,4,6}

T4----> {4,5,7}

T5 ----> {1,4,5}

T6 ----> {2,4,6}

T7 ----> {1,3,5}

T8 ----> {2,4,7}

T9 ----> {3,5,6}

T10----->{2,6.7}

T11----->{3,6,7}

T12------>{4,6,7}

use Hash Function as (i \* j)mod 10.

Min support=3

Q2. What's the difference between Content based and collaborative filtering recommendations ?

| Content Based Filtering | Collaborative Filtering |
| --- | --- |
| Content-based filtering, makes recommendations based on user preferences for product features. | Collaborative filtering mimics user-to-user recommendations. |
| Content-based filtering can recommend a new item, but needs more data of user preference in order to incorporate the best match. | collaborative filtering needs large dataset with active users who rated a product before in order to make accurate predictions. |
| To calculate how good a movie is to a user, we use cosine similarity | For binary rating (like or not), we can use a similarity measure like Jaccard Similarity to compute item similarity. |
| The point of content-based is that we have to know the content of both user and item. Usually we construct user-profile and item-profile using the content of shared attribute space. | Collaborative algorithm uses “User Behaviour” for recommending items. They exploit behaviour of other users and items in terms of transaction history, ratings, selection and purchase information. Other users behaviour and preferences over the items are used to recommend items to the new users. In this case, features of the items are not known. |
| In content based filtering you use properties of the objects and link similar ones and show them | collaborative filtering you usually use data of what was in any way linked together by an outside sorting entity (e.g. bought together by an online shopper) and show them in an ordered list. |
| Eg you show all the books that have have same author, same publisher, same genre and the most similar number of pages as book A first, then you slowly make it less strict... | Eg you analyse what books have been read by people that have read book A and the ones with highest count are on the top of the list. |

Q3. List various advantages and disadvantages of Collaborative filtering ? Which similarity is helpful user -user or Item -Item in Collaborative filtering recommendation ? Explain with the example ?

Advantages

* No domain knowledge necessary

We don't need domain knowledge because the embeddings are automatically learned.

* Serendipity

The model can help users discover new interests. In isolation, the ML system may not know the user is interested in a given item, but the model might still recommend it because similar users are interested in that item.

* Great starting point

To some extent, the system needs only the feedback matrix to train a matrix factorization model. In particular, the system doesn't need contextual features. In practice, this can be used as one of multiple candidate generators.

* **Not required to understand item content**: The content of the items does not necessarily tell the whole story, such as movie type/genre, and so on.
* **No item cold-start problem**: Even when no information on an item is available, we still can predict the item rating without waiting for a user to purchase it.
* **Captures the change in user interests over time**: Focusing solely on content does not provide any flexibility on the user's perspective and their preferences.
* **Captures inherent subtle characteristics**: This is very true for latent factor models. If most users buy two unrelated ...

Disadvantages

* Cannot handle fresh items

The prediction of the model for a given (user, item) pair is the dot product of the corresponding embeddings. So, if an item is not seen during training, the system can't create an embedding for it and can't query the model with this item. This issue is often called the cold-start problem. However, the following techniques can address the cold-start problem to some extent:

* Projection in WALS. Given a new item not seen in training, if the system has a few interactions with users, then the system can easily compute an embedding for this item without having to retrain the whole model. The system simply has to solve the following equation or the weighted version:

The preceding equation corresponds to one iteration in WALS: the user embeddings are kept fixed, and the system solves for the embedding of item . The same can be done for a new user.

* Heuristics to generate embeddings of fresh items. If the system does not have interactions, the system can approximate its embedding by averaging the embeddings of items from the same category, from the same uploader (in YouTube), and so on.
* Hard to include side features for query/item

Side features are any features beyond the query or item ID. For movie recommendations, the side features might include country or age. Including available side features improves the quality of the model. Although it may not be easy to include side features in WALS, a generalization of WALS makes this possible.

To generalize WALS, augment the input matrix with features by defining a block matrix , where:

* Block (0, 0) is the original feedback matrix .
* Block (0, 1) is a multi-hot encoding of the user features.
* Block (1, 0) is a multi-hot encoding of the item features.

Problems with collaborative filtering

• Scale

– Netflix (2007): 5M users, 50K movies, 1.4B ratings

• Sparse data

– I have rated only one book at Amazon!

• Cold-Start

– New users and items do not have history

• Popularity bias

– Everyone reads “Harry Potter”

• Hacking

– Someone reads “Harry Potter” reads “Karma Sutra

Problems that commonly occur with collaborative filtering include (but are not limited to) the following:

* The early rater problem occurs when a new user is introduced to the system and has yet to rate a selection of items significantly large enough for the service to start suggesting similar items. A simple solution to this would be to have the user rate content they may have consumed on another site or platform in order to let the algorithm have a sufficiently large amount of data to start recommending products with an acceptable probability of success. It is of course harder to determine what for example the closest neighbour is if there are a vast amount of users who rated content similarly to this user, but rated other content severely different from one another[10].
* The sparsity problem occurs when there are too little information to work with in order to provide the user base with decent approximations as to which products they would likely prefer. The sparsity problem occurs if the data is too spread, and while users might have rated a fair amount of movies, too few have rated the same movie to make accurate predictions[10].
* Cold start occurs when one or several users or products is added to the system, and not enough data is recorded to provide optimal recommendations. This can affect an entire recommendation algorithm if the service is newly established because it can not provide 7 acceptable recommendations to any one user as of yet. This is a subject that has been researched for a long time and is still a problem in many recommendation systems[11].
* A gray sheep is a user which has no obvious closest neighbour and seems to rate content in an unpredictable pattern unlike that of any other user. A gray sheep is a problem as it can be hard to estimate what the user might like or dislike when there are no similar consumption or rating patterns[10]. It takes a certain amount of time and amount of ratings for this new element to be introduced into the system in a way such that it works similarly to the rest of the objects within it. It is simply the process that leads up until the time when element has been correctly established within the environment.
* A shilling attack is when a person or group of people create multiple accounts in order to promote certain content and take away user’s interests in other, in an attempt to promote their own products and hurt their competitors, and is an attempt to manipulate users into buying, watching or subscribing to a certain type of content based on a hidden agenda[14].