Collaborative Filtering Advantages & Disadvantages

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.

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 i_0 not seen in training, if the system has a few interactions with users, then the system can easily compute an embedding v_{i_0} for this item without having to retrain the whole model. The system simply has to solve the following equation or the weighted version:

$$\min_{v_{i_0} \in \mathbb{R}^d} \|A_{i_0} - U v_{i_0}\|$$

The preceding equation corresponds to one iteration in WALS: the user embeddings are kept fixed, and the system solves for the embedding of item i_0. 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 $\bar{A}$, where:

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

3lock (1, 1) is typically left empty. If you apply matrix factorization to $\bar{A}$, then the system learns embeddin
ıatures, in addition to user and item embeddings.

Previous
← Matrix Factorization (/machine-learning/recommendation/collaborative/matrix)

Next
Movie Recommendation System Exercise → (/machine-learning/recommendation/labs/movie-rec-programming-exercise)