CS410 Technology Review **Amanda Wang, afw3**

Introduction

Recommender systems are growing more commonplace in a variety of different contexts, such as song listening, online shopping, and content-streaming websites. They are so important to huge companies like Netflix, Amazon, Facebook, Spotify, and Google/YouTube, and we are becoming more accustomed to seeing content that seems perfectly curated for our interests and recent tastes. As a result, building an effective recommendation algorithm is both beneficial for us as consumers (to find content or items we didn't even know we were looking for) and for companies to keep their users happy and engaged. There has been a lot of research done into different recommendation algorithms, and this paper will discuss some of the major systems and techniques.

First, I will focus on the two major recommendation system types: content based filtering and collaborative filtering, and how they compare. Next, I will talk about a variety of different collaborative filtering techniques, specifically different memory-based and model-based models. Because of the issues of data sparsity and scalability, many modern collaborative filtering approaches use matrix factorization models based on SVDs, PCA, PMF, or NMF. I will explore these techniques and think about them in the context of our project. There are also many different similarity measures, evaluation metrics, and hybrid model structures that we can consider experimenting with.

Content-Based vs. Collaborative Filtering

This paper will start with a discussion about content-based vs collaborative filtering. Content based recommender systems typically recommend items to a user based on information from an item profile. For example, different pieces of information we might have about a book are the "format" (hardcover vs softcover vs ebook vs audio book), "length" (in terms of pages or number of minutes), "author", "publishing company", "publication year", and "genre." This approach avoids a cold start problem, since the moment a new book is published and available to the public, we already have all this information. Another benefit is that this is not a computationally heavy approach, since we only have a short item profile for each book. Furthermore [1], this approach results in a high coverage of different books. Even if a book is very obscure, if it seems to match the user's tastes strongly, it will be recommended to them.

The main downsides of content-based filtering are that it is sometimes quite difficult to build accurate, thorough item profiles (for example, it's hard to find a list of accurate genres for every book and come up with a list of "features" to describe every book), and the user is rarely recommended things outside their usual interests. On the other hand, collaborative filtering doesn't have either of these issues, but it is significantly more computationally heavy since we would need to build a matrix measuring similarities. There are also the issues of sparsity and

scalability, which will be discussed further in the next section, not to mention the cold start problem with new content.

Different Collaborative Filtering Techniques

Because collaborative filtering is typically the more effective and more well-studied approach, this paper will focus more on the two major types of collaborative filtering: memory based and model based filtering. Memory based algorithms [2], also known as neighborhood based algorithms, use statistical techniques to find users that are similar and create neighborhoods based on this similarity. Then it combines the tastes of the users in each neighborhood to create recommendations. This type of filtering is [3] easily implemented and allows new data to be added gradually, but it is also very dependent on human behavior in the form of ratings. It also has the issues of sparsity (most users have only rated around 10-500 books out of millions of books) or limited scalability due to the computational power needed to handle such a large similarity matrix (in our case, approximately a million by a million).

These issues are addressed by model-based collaborative filtering algorithms, which provide recommendations based on a model of user ratings. For these algorithms, [2] we are trying to compute the expected value of a user's rating for a new rating based on their historical ratings. The model building process itself uses different machine learning algorithms like the Bayesian network, clustering, and rule-based approaches. If we do decide to take a model-based approach, we might consider using a clustering algorithm, which clusters similar users into classes, calculates the probability of each user being in each class, and then using conditional probabilities to determine rating predictions.

Model-based approaches address the scalability issue by taking advantage of dimensionality reduction in the form of matrix factorization. Some common approaches [3] are Singular Value Decomposition, Principal Component Analysis, Probabilistic Matrix Factorization, and Non-Negative Matrix Factorization. Both SVD and PCA help decrease dimensionality of the data, with SVD producing a low rank matrix approximation and PCA resulting in a list of components that explain the most variance in the data. Meanwhile, PMF is a probabilistic linear model with Gaussian noise and NMF is like SVD with the added constraint that the data matrix is assumed to be non-negative. The first two approaches seem to be more widely used by researchers, so we may end up performing user-based collaborative filtering using SVDs to help decrease the complexity of our model.

Before we design and implement our model, there are still more things for us to consider, such as the different measures of similarity [2] (cosine similarity, Pearson correlation, and adjusted cosine similarity, which accounts for the user bias in ratings), as well as the evaluation metrics, including coverage (the percentage of items that are ever recommended), prediction accuracy, MAE, RMSE, precision, recall, and F1 score.

Hybrid Approaches

Finally, a reach goal for our project would be to develop a hybrid model that can combine content-based and collaborative filtering. There are several different approaches to combining the techniques, listed into the following classes according to Burke [4]: weighted, where the two systems are simply combined after independently making judgements; switching, where the system switches between different recommendation techniques based on the current situation; mixed, where the systems run independently and the user is shown all the results; feature combination, where collaborative filtering information is treated as another "feature" in the content based approach; cascade, where one technique is used first to find probable items to predict and the second is used to decide which of the probable items to actually recommend; feature augmentation, which uses the output recommendations of one system as the input to another; and meta-level, which uses the model generated from one approach as the input for the second model. If we get to it, we would probably consider the weighted, switching, or cascade approaches for combining our models because they seem more intuitive and easily explained.

Conclusion

Through this paper, I explored the differences between content based and collaborative filtering, and compared a variety of CF techniques, like memory-based and model-based techniques (like SVD, PCA, PMF, and NMF). I also talked about these techniques in the context of our final project, in addition to different similarity measures, evaluation metrics, and hybrid model structures we could use.

I plan on using this new knowledge to make more informed decisions when deciding on and evaluating models for our final project. For example, I would want to consider book coverage of our recommendation system since so many new books and authors have a difficult time getting started simply because they get lost among the more established authors and works. We could also try doing a memory based approach first for our collaborative filtering model since we are only working with 20,000 users, but if efficiency becomes a real problem for us, then we could shift to an SVD or PCA model-based approach instead. Finally, if we get the chance to implement a hybrid model, we will study the results of our two models first. If they seem to perform well under different circumstances, then we could use the switching hybrid approach. If not, then we can stick with a weighted or cascade approach.

Sources:

[1] Ana Belén Barragáns-Martínez, Enrique Costa-Montenegro, Juan C. Burguillo, Marta Rey-López, Fernando A. Mikic-Fonte, Ana Peleteiro, A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition, Information Sciences, Volume 180, Issue 22, 2010, Pages 4290-4311, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2010.07.024.

[2] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (WWW '01). Association for Computing Machinery, New York, NY, USA, 285–295. DOI:https://doi.org/10.1145/371920.372071

[3] Dheeraj Bokde, Sheetal Girase, Debajyoti Mukhopadhyay, Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey, Procedia Computer Science, Volume 49, 2015, Pages 136-146, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.04.237.

[4] Burke, Robin. (2002). Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction. 12. 10.1023/A:1021240730564.