Content Boosted Collaborative Filtering with WARP Loss for Medical Diagnosis with Patient Data

Yu Fung David Wang
david.wang@emory.edu
Emory University

ABSTRACT

Recommender systems (RS) have a significant impact on products and applications regarding individuals and corporations. Research is performed on RS to improve predictive accuracy and runtime cost to identify user interests from an abundance of data. In most studies and applications, the recommendations given pertain to commercial products, social media, and other entertainment domains. However, RS for health-based recommendations, such as medical diagnosis, are vaguely studied due to its highly sparse nature with perturbed, private patient data, and the lack of patient data with rare treatments to cause a cold start problem. As such, there is a demand for Health Recommender Systems (HRS). Previous studies have shown the plausibleness of certain baseline algorithms, such as mechanisms of collaborative filtering (CF), including k-nearest neighbors (kNN), singular value decomposition (SVD), and co-clustering, to be effective on commercial data, but with weaknesses of data sparsity and cold start. Henceforth, these models would theoretically produce inaccurate recommendations on medical data. To alleviate such issues, a hybrid model of content boosting collaborative filtering (CBCF) with weighted approximate pair-wise (WARP) loss was utilized via the LightFM package. The basic idea of the model was to extract CF and CBF advantages and combine them by using CF for latent representation of user and item features as embeddings/latent vectors and CBF of representing the embeddings/latent vectors as linear combination. These two factors would solve the cold start problem and data sparsity respectively. Furthermore, the loss function of WARP would work to maximize the rank of positive recommender feedback. The model produced was tested on synthetic generated health data called FairGRecs. The results from CBCF with WARP were analyzed and compared with baseline CF algorithms, and the results were also compared with data from running the model with the common commercial dataset domains of movie recommendations and stack exchange. It was found that the hybrid algorithm performed better as a HRS compared to the baseline CF models. It was also found that the hybrid HRS of CBCF was an appropriate application towards FairGRecs when compared to CBCF implementations on commercial datasets. Furthermore, the CBCF model was noted to perform well with the sparsity issue, while faltering from the cold start issue.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Yu Fung David Wang. 2022. Content Boosted Collaborative Filtering with WARP Loss for Medical Diagnosis with Patient Data. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recommender systems (RS) is a crucial model for everyday tasks of sorting through massive amounts of data and identifying user interests to make information sorting and searching easier. It has alleviated communication between the World Wide Web with that of users to give relevant suggestions. Currently, there is a growing demand in developing recommender systems to manage information overload for personalized recommendations in many of the practical applications seen in social media and application products [5]. As such, a heavy influx of research is made by individuals and corporations to target customers and users to attract them to their products. In particular, the research deals with the field of data mining for its important analysis technique used in recommendation systems to predict user interests in many commercial transactions.

Through the persistence of these transactions, the capitalistic means of utilizing RS became increasingly prevalent, such as that of movie recommendations from Netflix or product recommendations from Amazon. Shifting away from that domain, there is the realization that clinical data, which is widely distinct from common RS domains, are required to make a range of recommendations; such recommendations may include patient-oriented decisions, medical diagnoses, pharmaceutical recommendations, and food preferences [11]. These recommendations can not only improve self-care of patients, but it will also better support medical decisions made by professionals. As of now, RS in healthcare has been explored and called Health Recommender Systems (HRS) in cases of food, drug, and healthcare professional recommendations [11]. However, one field that has yet to be explored thoroughly in HRS is that of disease or illness recommendation when given patient symptoms via medical diagnosis due to the certain challenges that may arise.

One of the many challenges is that of the diversity and sparsity of patient profiles. Both of these attributes cause issues in HRS where data diversity formulates the cold- start problem while data sparsity formulates a sparse matrix to make calculations more difficult. First, the cold-start problem refers to when items added to the system have none or little interactions with users. Henceforth, it makes recommendations difficult for these users due to the task of having to make a user profile without a history of similar user preferences. For example, in a medical setting, some patients may experience certain symptoms or diseases that lack the presence of recorded transactions and data, thus making it difficult for the HRS to distinguish a good diagnosis or treatment for the patient. Moving on, the data sparsity problem refers to when the number of

active users only rated a small portion of items. In a matrix setting, it can be imagined as the matrix having plenty of N/A values as opposed to real values. Sensibly, most real datasets suffer from this issue. Medical datasets, out of all real datasets, suffer more from this issue due to outdated technology in hospitals, user error in medical technology adaptation, and the fragmentation of electronic health records in health interoperability and information networks. Due to the prevalence of these issues, it is important to find a algorithmic way to improve the plausibility of HRS in the medical domain.

Another challenge with medical dataset is with patient privacy. Patient data is often covered with sensitive information with data on address, full name, age, diseases, etc., of individual patients. Feeding a dataset full of data that could be utilized as quasi-identifiers would be beneficial to achieve accurate recommendation results but the trade-off is not worth it as it can be potentially harmful for individuals against attackers. As such, this is another problem that needs to be addressed in HRS.

As a way to tackle the challenges in HRS domain, a brief introduction to the inner workings of RS through the basic yet effective method of collaborative filtering (CF) will be explained. CF methodology works with collaborative base algorithms to collect information from users to make automatic predictions. As such, the collected information is built on user/item feedback where the logic of the feedback can be narrowed down to user-based and item-based feedback. User-based CF has users' ratings on the same item compared and computed to predict the rating for a new user by a weighted average [8]. Item-based CF, on the other hand, reverses the view between users and items where it utilizes items that have been previously liked by the user to predict the rating the user would give a new item by a weighted average. In both cases, the weight measure is the similarity of the old users/items to the new user/item and the similarity method of calculation depends on the context of the problem.

However, CF models suffer from a cold-start problem [8]. This problem is caused by how the recommendations are only computed on user or item feedback and if there was a lack of feedback, the item cannot be recommended to users.

As a result, researchers developed content-based filtering (CBF) to fix the cold start problem by filtering recommendations through matching individual user interests with descriptions of items instead of basing the recommendation on feedback while collecting the information from individuals.

Even though CF may work in basic scenarios, previous studies have shown in user-based and item-based algorithms of CF, there are often skewed correlations and sparse ratings due to the data not generalizing a population [7]. For example, a skewed result would occur if a group of physicians agree on a controversial treatment as compared to a couple of physicians who agree upon a popular treatment for a patient. Furthermore, the implementation cost is high in naïve implementation due to requiring comparisons against all other treatment methods [7]. For physicians and hospitals that need to categorize and recommend data to the millions of data points and values, the processing time and memory consumption cost would be fatal.

Thus, a hybrid approach that utilizes both CF and CBF approaches can be utilized for a more effective result to balance out their perspective weaknesses and improve recommendation accuracy. A hybrid RS can be implemented in multiple ways including, averaging results of CF and CBF methods, adding CBF characteristics into a CF model, or constructing an entirely new model with both CF and CBF characteristics [10].

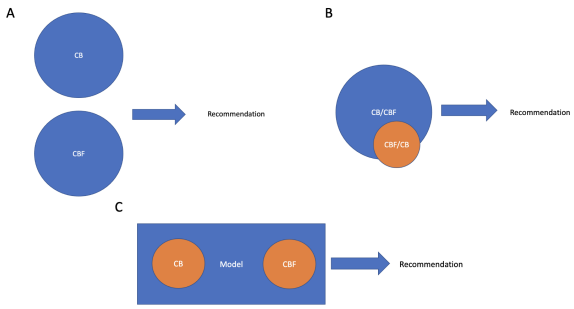

Figure 1: Hybrid approaches where A shows the integration of CF and CBF characteristics, B shows integration of CF/CBF into CBF/CF respectively, and C shows the incorporation of CF and CBF into a newly designed model

Given the idea to build a hybrid model with a new model, further research and emphasis is placed on methods to combine CF and CBF that proves to be most beneficial. Currently, one of the most valuable open-source recommender system frameworks is known as LightFM [3]. LightFM first takes in an interaction matrix to register user ratings or preferences over items for CF, and then it generalize new items from past item features and new users from past user features to incorporate CBF [3]. As a result, a hybrid model is created by adding characteristics of CBF into a purely CF model. As LightFM is the focus of the paper, more explanations and information will be provided in the preliminaries of the paper.

To iterate the formulation of this study, a hybrid RS built on content boosted collaborative filtering method will be considered on a generated synthetic medical dataset known as FairGRecs with 10 thousand users, 80 thousand items, and 1.5 million ratings [9]. The hybrid model will be tested to see how it performs against the issues of cold-start and data sparsity. The FairGRecs dataset will be utilized to combat the issue of patient privacy through a naïve implementation of a synthetic dataset. During experimentation, to note the effectiveness of the model, the hybrid model will be compared to baseline CF models with kNN, SVD, and co-clustering capabilities on the medical dataset. Then, to show the effectiveness of the model in the medical domain and under the privacy constraints, the hybrid model results on FairGRecs would be compared to a commonly utilized movie recommendation domain called MovieLens dataset with 100 thousand ratings from 1000 users on 1700 movies, and a Stackexchange dataset with 9000 users, 72 thousand questions, and 70 thousand answers [3].

The current paper will start with related work on hybrid learning RS including an exploration of HRS, a therapy recommendation system that uses the neighborhood-based collaborative filtering model, and a cloud-assisted drug recommender system that utilizes SVD to reduce matrix dimensions. Next, problem formulation and

preliminaries will be targeted with specific subsections. These subsections would be split amongst co-clustering CF, kNN CF, SVD CF, the CBCF methodology associated with LightFM, and the loss methodology of WARP to maximize positive feedback. Then, the FairGRecs synthetic health data would be explained, along with the Movielens and Stackexchange datasets. Furthermore, the resulting hybrid model will be compared based on the area under the curve (AUC) metric to baselines of the kNN CF, SVD CF, and co-clustering CF models. The results on the FairGRecs health dataset domain would also be compared to results seen in baseline commercial models of the MovieLens dataset and the Stackexchange dataset. Finally, closing thoughts would be given along with future directions of HRS.

2 RELATED WORKS

2.1 Recommender systems in the healthcare domain: state-of-the-art and research issues

In this survey, the researchers aimed to explore the accessibility and capability of HRS and expose certain challenges and limitations that arise. First, they start off by mentioning the current prevalent uses of HRS, including for food recommendation to provide users with healthier food choices to alter eating behaviors with cloud-based food recommender systems, drug recommendations through a user-based CF technique to suggest proper medications to diabetes patients with attributes of age, insulin, glucose, BMI, BP, and triceps thickness, and a healthcare professional recommendation through CBF techniques. Primarily, the authors noted HRS being applied in three use cases, including for a new patient, an existing patient with no interactions with primary care doctors, and an existing patient with prior interactions with primary doctors. Given the dynamicity and complexity of healthcare recommendations, it is often safe to assume each case has similar plausibility to occur. In other words, HRS should be able to apply recommendations to patients given patterns and treatments exhibited by the patient him or herself and by previous patients. As such, it will be worthwhile to experiment with LightFM so that CF can get the current user preferences and ratings while CBF can be used to generalize previous recommendations to get new user and item features. Furthermore, the paper mentions the challenge where HRS must be evaluated on the trust, robustness, and privacy it provides. Hence, by using synthetic datasets that are meticulously generated from generalized data, such as that of the FairGRecs algorithm, to provide a certain level of trust for this study. Nevertheless, there are still threats of attack on the FairGRecs algorithm that could be further perturbed for better robustness. [11]

2.2 Neighborhood-based Collaborative Filtering for Therapy Decision Support

In this study, HRS was specifically evaluated on therapy/healthcare professional recommendation. The proposed methodology was through neighborhood-based CF methods to exploit highly dimensional clinical data. Their method was akin to that of kNN CF methodology in which they integrated differing amounts of neighbors denoted as k. The CF algorithm first used information on therapy history such as previously applied therapies and associated therapy response and information on patient disease to make predictions on patients having similar therapy history and characteristics. The therapy outcome predictions are further predicted with w weighted sum of k nearest therapy cases to the case that is currently under investigation. It was noted in the study that the neighborhood-based CF provided reliable and personalized therapy recommendations. However, the algorithm was parameter heavy since it was highly dependent on k, where increasing k causes a performance decline from noise influence on prediction accuracy. As such, a way to ensure constant performance would be to only evaluate on algorithms that utilize epochs rather than a hard-set hyperparameter to improve performance overtime. [2]

2.3 Cloud-Assisted Drug Recommendation Service for Online Pharmacies

In this research, the authors designed a new HRS architecture called Cloud-Assisted Drug Recommendation (CADRE) for medicine recommendation. In CADRE, there are three important components including a drug database with information on various drugs, a modeling server which corresponds customer rating with the drugs, and a recommendation server that recommends drugs based on user keywords. One of their key concerns with their recommender server methodology is the sparsity of the interaction matrix produced by CF. This solution to the matrix sparsity is especially important due to the unpredictability of patient data. By utilizing SVD with CF, it improved the performance of CADRE by around 5 to 10 percent based on the number of drugs to recommend. However, despite CADRE being shown as an effective drug recommendation method according to the demand of customers, it does not consider the case of new customers that do not have certain patient demographic information needed to provide beneficial recommendations. As such, utilizing a hybrid method with CBF may positively impact CF models by looking for items that are similar to the ones assigned to him or her in the past. [13]

3 PROBLEM FORMULATION

RS have not been extensively researched in the domain of healthcare. Despite the existence of HRS, it is not heavily utilized due to the sparsity, cold start, and privacy issues of patient data. To be more specific, patient data suffer from unpredictability of user profiles/medical technology, differentiating cases between individuals, and lack of robustness against attackers. Standalone, popular RS methodologies of CF and CBF are insufficient in protection against the cases. Henceforth, a combination of both into a CBCF mechanism would prove vital to prevent the cases of data sparsity and cold start. Furthermore, synthetic datasets such as that of FairGRecs would provide a naïve solution to certain privacy attacks.

4 PRELIMINARIES

4.1 RS Algorithms

4.1.1 Co-Cluster Method. Consider low parameter approximations based on simultaneous clustering or co-clustering of users and items in the rating matrix. Each missing rating is approximated by the

average value in the corresponding co-cluster. Along with the co-cluster average, the co-cluster methodology also incorporates biases of individual users and items by including terms (user average – user cluster average) and items (item average – item cluster average) to get the matrix. [1]

$$A_c + (R_u - R_{uc}) + (R_i - R_{ic})$$

Figure 2: Co-cluster average + (average rating of user - average rating of user-cluster) + (average rating of item - average rating of item-cluster

This approximation matrix can be used to predict unknown ratings by finding optimal user and item clustering such that the approximation error of the matrix with respect to the known ratings are minimized [1].

4.1.2 k-Nearest Neighbors Method. To generate recommendations, standard kNN is utilized to get user or item similarity. For example, under the assumption of items, when an inference is made by kNN, it will calculate the distance between the target item and every other item in the database and rank the distances to return top k nearest neighbors for the most similar item features to recommend. This distance is most often computed via the Pearson correlation coefficient between the two items x and y. [6]

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

Figure 3: Pearson correlation coefficient (r) where $\overline{x}$ is the average rating user/item x gave to all items/users and $\overline{y}$ is the average rating user/item y gave to all items/users

4.1.3 Singular Value Decomposition Method. Singular value decomposition (SVD) is based on a matrix factorization method that takes a *mxn* matrix A with rank r and decomposes it to

$$A = USV^T$$

Where U and V are orthogonal dimensions and S is the singular matrix. The SVD method provides low-rank approximation of the original data matrix, hence they come in to be particularly useful in the context of RS by ensuring the number of diagonals of k is much less than the number of column vectors/rank to reduce the data dimensionality and prevent a sparse data matrix. By using the orthogonal dimensions and the singular matrix, the predicted rating can be found by a reduced matrix added to the mean of the appropriate row. [12]

4.1.4 LightFM. LightFM is a hybrid RS model that exists to surpass CF and CBF in that the model must be able to learn user and item representations from interaction data in CF and the model must be able to compute recommendations for new items and users in CBF. The first point was satisfied by CF implementation of latent representation of users and items as latent vectors and the second point was completed by CBF representing these latent vectors of users and items as linear combinations of their content features. For

$$pr_{ij} = row_i + U_k * \sqrt{S_k^T} * (i) * \sqrt{S_k} * V_k^T * (j)$$

Figure 4: The predicted rating for the i-th user on the j-th item where the second part of the equation gives the corresponding of the reduced matrix and the prediction is generated by adding the mean to the appropriate row

latent representations, it is a machine learning technique to infer variables that resemble one another more closely by their positioning in the latent space, where the latent space is an embedding of a set of items that encode features. For example, if "PlayStation" and "Xbox" are liked by the same users, their embedding would be close together. On the other hand, linear combinations of content features simply mean the construction of an item is by summing the features of the item. For example, the representation of a "online gaming system" is by summing representations of "online" with that of "gaming systems". In doing so, CF and CBF advantages are united to prevent cold-start by tagging similar embeddings and reduce highly sparse data through simple linear representation. Further mathematical implementation can be found in the Technical Design section. [3]

4.2 Loss Function

4.2.1 Weighted Approximate-Rank Pairwise. Weighted approximate-rank pairwise approach (WARP) gives the loss based on a predicted ranked list. As such, it is a loss function that optimizes classification metrics such as AUC for RS. WARP utilizes a negative sampling technique to only modify weights of negative items to optimize recommending negative items to a user that are deemed as more appropriate. Once the negative item is found, gradient updates are performed to model parameters, including users, positive items, and negative items to correct the model. The gradient updates are weighed using the estimated rank of the item to the user where these updates are amplified if the model failed to see an interaction between the item and user. Thus, the rank of the interaction between the item and user can be estimated by counting the negative items that were considered before an appropriate user to item interaction was found to fit the model. Further explanations of the technicalities of WARP can be found in the technical design. [4]

4.3 Datasets

4.3.1 FairGRecs. FairGRecs is a synthetic healthcare dataset that is used to evaluate and benchmark recommendation methods. As HRS requires access to information such as medical documents, rating datasets given by users to documents, and personal health information of users, it creates a hostile environment for attackers to create legal and ethical constraints. As such, EMRBots is utilized to generate documents of 10 thousand artificial patients. Even though none of these patients directly correspond with real patients, it creates a necessary medium to research the most optimal HRS method for medical diagnosis recommendations. After a working model is found for HRS, further implementations can be considered to perturb real patient data to be utilized in HRS. More explanation of the generative process can be found in technical design. [9]

4.3.2 MovieLens and StackExch. MovieLens is a rating data set collected by GroupLens Research where its main dataset is dynamically updated with new movies. However, for the sake of reporting preliminary model results, it will run on the static dataset. Movielens is utilized with RS for movie recommendations and has been commonly used for RS because of its explicit ratings. However, the dataset is very sparse as the number of zero entries far exceeds number of actual entries. Nevertheless, this exceedingly high sparsity was to be expected of a real-world dataset.

Stackexchange consists of users of a popular internet forum known as StackExchange where users answer questions. This dataset consists of all user-contributed content where the purpose of RS is to recommend new questions based on which users answered which questions. On the flip side, this dataset suffers from the cold-start problem where RS cannot draw inferences for the questions due to a lack of interactions between user and questions.

5 TECHNICAL DESIGN

5.1 FairGRec in Synthesizing HRS Data

The FairGRecs dataset utilizes 10 thousand chimeric patient profiles provided by EMRBots, which is an artificial large medical dataset that consists of patients' admission details, demographics, socioeconomic details, labs, medications, and more [9]. FairGRecs was used to generate the synthetic health data in a way that can be fed into a HRS. As such, the EMRBots data was manipulated in a way to combine personal health records, documents, and ratings [9]. The data was generated manually through a user-friendly API that utilized Java to specify file path, number of documents created per category, number of keywords each document will have, number of popular documents per category, percent of patients in different groups, the minimum and maximum ratings per different groups, and the distribution percentage of ratings to documents.

For the case of the documents in this study, 270 documents with 10 keywords per document and 70 documents that are popular per category were utilized. For the patient information, 50% of patients were set as occasional (gives 20 to 100 ratings), 30% were set as regular (gives 100 to 250 ratings), and 20% were set as dedicated (gives 250 to 500 ratings). Finally, the rating distribution was set to be roughly equal of 20 per ratings of 1 to 5.

These parameters resulted in generation of 3 different datasets: patients profile dataset, rating dataset, and documents dataset.

For the patient profile dataset, it was first important to note that EMRBots does not pull real data from existing Electronic Medical Records (EMR), thus there were no privacy concerns [9]. The document showed both population-level and patient-level characteristics. However, as this dataset showed only demographic data, it was unnecessary towards the implementation of HRS.

For the ratings dataset, it consists of user preferences towards sets of items, which in this case was of certain diseases or disease symptoms. First, it assumes that all patients are given a minimum non-zero number of ratings to work with the cold-start problem in HRS [9]. This is important because in cases in CF, if a user has not given any ratings, he or she would not be able to find any similar users and cannot be provided with recommendations. Second, the number of ratings per user was selected at random and in a numerical range depending on if they are occasional, regular, or dedicated. Finally, user ratings are assigned to different groups, where depending on the health problem, ratings will belong to the same health subtree, where the health problem would be designated by a document ID that points to the document dataset.

In the document dataset, randomly selected keywords were utilized as description text of nodes in each health subtree. Each document has a ID that can be pointed to by patient ratings.

From the generated datasets, FairGRecs was seen as a ranking problem to recommend the top items for a particular user or patient for a specific topic or disease. Henceforth, this dataset has the necessary user and item metadata to generalize new item features (diseases) and new user features (user id) for users and items based on the explicit ratings provided.

5.2 LightFM with WARP in HRS

In LightFM, users and items were described by their features, denoted by f_i or f_u. These features were known initially and represent the metadata of the items and users. The embeddings, which represented the item and user features, were estimated during the model training process with the WARP loss function and stochastic gradient descent (SGD) methods to minimize the loss function. Given that the focus was item features, WARP worked by sampling for negative items (wrongly associated item-user pairing) at random from the remaining items in the dataset. Then, predictions to the test dataset were computed for the negative and positive items where if the negative item's prediction probability exceeded that of the positive item, a gradient update by SGD was performed to update the rank of the positive item higher and the negative item lower. If this violation was not found, the negative items were continuously sampled until a violation was found. There are two caveats to this process. First, if a violating negative item was found at the first try, a large gradient update was made as lots of negative items were likely ranked higher than positive items. Second, if lots of negative item sampling was necessary to find a violating item, small updates were performed as model is likely nearly optimal. By doing so, the embeddings appropriately captured user preferences via user and item features.

These generated embeddings user were denoted as e_f^I and e_f^U for each item and user feature f and the features were also described by a scalar bias term to better fit the data. These generated embeddings are also known as latent vectors and make up the latent representation of collaborative filtering to represent items and features.

Furthermore, the linear combination of i and u were given by the sum of the features' embeddings that describe each user or item, where I is the set of items and U is the set of users. The bias term for item i and user u, denoted by b_f^I and b_f^U was also given by sum of features' biases to adjust and better fit the data.

$$p_i = \sum_{j \in f_i} e_j^I | b_i = \sum_{j \in f_i} b_j^I$$

$$q_u = \sum_{j \in f_u} e_j^U | b_u = \sum_{j \in f_u} b_j^U$$

Figure 5: Linear combination seen through summation formulas of feature embeddings/latent vectors, e, and summation of bias terms, b, for items i and users u

Given the linear combination and their biases, the prediction for the user and item was achieved by getting the dot product of user and item representations, adjusted for a better fit by user and item feature biases.

$$f(q_u \cdot p_i + b_u + b_i)$$

Figure 6: Prediction of user and item representations and adjustment of representations with feature biases

The result was then be used to get the probability of occurrence of the predicted rating to which the sigmoid function was used as $f(*)$ to predict the probability from a range of 0 to 1. Note that other functions could be also used in this step to substitute the sigmoid function.

$$f(x) = \frac{1}{1 + exp(-x)}$$

Figure 7: Sigmoid function used to predict data where sigmoid function is better at predicting binary data

With the predicted probability of ratings, a novel recommendation was made for users and/or items and the resulting recommendation based on the most likely rating was compared to that in the actual dataset. The comparison metric utilized was area under ROC curve (AUC) where it measured the proportion of user-item pairs in the proposed ratings that were identified correctly to be ranked higher than irrelevant items in the user-item pair based on the actual dataset.

$$AUC(R)_n = \frac{\sum_{r \in R} r \sum_{r' \in (1 \ldots n/R} r'}{|R|(n - |R|)}$$

Figure 8: AUC function that shows rank of relevant item (first summation) multiplied by rank of irrelevant item (second summation) over all possible pairs of relevant and irrelevant items

6 EXPERIMENTAL ANALYSIS

6.1 Comparison of Models

To reiterate, the CBCF model was tested against three baseline collaborative models that included co-clustering, kNN, and SVD methodologies.

The CBCF model was ran with 30 as the dimensionality of the feature latent embeddings, an alpha value of 1e-05 which was the L2 penalty on item and user features, a learning rate of 0.01 for AdaGrad learning schedule, 100 maximum number of negative samples for WARP fitting, and 30 epochs. By using a alpha value or L2 penalty, it reduces overfitting of the model, while a lower learning rate allows for more optimal training over a longer run time. A higher number of max negative samples leads to improved accuracy with the cost of run time as it would sample more negative triplets for users that are already well represented by the model. Finally, the epochs were optimally set at 30 to prevent overfitting and underfitting.

The co-clustering CF model was set up with default optimal parameters where there were 3 user clusters and 3 item clusters as the number of clusters to use was ambiguous. Epochs were not utilzed and instead, 30 coclusters were evaluated to find the optimal parameter to produce the best AUC.

The kNN CF model was ran with a basic model that took minimum k or number of neighbors of 1 and was tested based on the max k from 1 to 30. Epochs were not utilized and the task was set on finding the optimal number of neighbors k for the highest AUC metric from a range of 1 to 30.

The SVD CF model had 100 factors, regularization term of 0.02, a learning rate of 0.01 like CBCF, and 30 epochs. Other than default parameters, all other parameters were set to be comparable to the CBCF model and thus was placed to a side by side comparison with CBCF.

Experimentation was performed by running the models on the FairGRecs dataset. In the dataset, the document keywords were utilized as the item features for the users (user ID) and items (document ID) in the ratings dataset. Similarities were found between users and items given the item features for the models. Then, different implementations were furthered depending on the preliminaries and technical design of the different models. Finally, cross-validation was performed on the model to train data and evaluate the model on test data that made up 25% of all data. Optimally, the model should be able to recommend appropriate medical diagnoses for the test data and was tested based on the AUC metric.

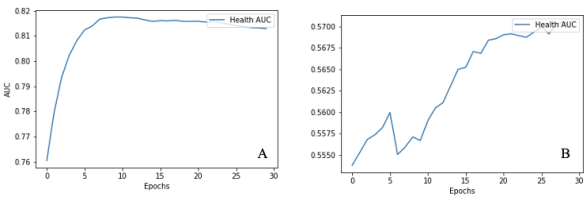

Figure 9: RS compared on FairGRecs where A is CBCF, B is SVD CF and these models were evaluating upon 30 epochs.

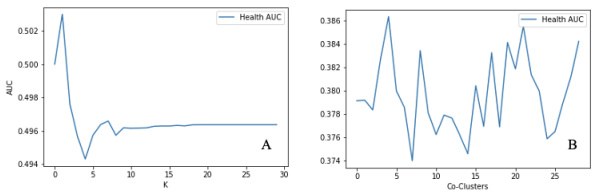

Figure 10: RS compared on FairGRecs where A is kNN, B is Co-Cluster and these models were evaluating upon there model parameters where kNN was ran upon 30 differing k while co-cluster was ran on 30 differing item and user co-clusters.

Table 1: Optimal AUC from CBCF and Baseline Algorithms

Algorithm	Best AUC
CBCF	0.81745
SVD	0.57112
kNN	0.50378
co-cluster	0.38659

After running the experiments on the FairGRecs dataset, the CBCF model performed the best by a margin with an AUC that starts of at 0.76 and has logarithmic growth where a sharp increase in AUC was observed for the first 10 epochs and the growth steadied to a halt for the remaining 20 epochs at around 0.820 where the optimal AUC was noted at 0.81745. Following the accuracy of CBCF, SVD CF model performed the second best with increasing AUC per epoch other than at epoch 5 where it dipped. However, the AUC stayed within the range of 0.55 to 0.57 throughout the 30 epochs so they were not huge divots. The optimal AUC noted for SVD CF model was 0.57112 Similarly, the kNN model and the co-cluster model only dipped slightly by the hundredths. Nevertheless, for the kNN CF model, which performed third best, an optimal k was observed at $k = 2$ to produce the optimal AUC of 0.50378. The co-cluster CF model, which performed the worst had no steady pattern with the number of co-clusters as the AUC wavered from 0.374 to 0.387. The optimal AUC produced by the co-cluster CF model was 0.38659.

From these results, two points can be concluded. First, CBCF performed the best with its hybrid nature in the health domain as compared to the other CF models. The AUC produced demonstrated that the top items produced by the CBCF model had more positive items that matched user preferences, followed by SVD, kNN, and co-cluster. Sensibly, the sparse medical data and cold-start characteristic of new patients in the test set made the implementation more difficult for the CF models as compared to the CBCF model. Second, when evaluating the AUC produced by the hybrid model, the AUC produced was decent at around 0.76 to 0.82 AUC. As such, this proves the plausibility to utilize the proposed CBCF model for mainstream HRS applications.

6.2 Comparison of Datasets

Following the comparison of models, it was noted that the CBCF model was able to perform substantially better than the baseline CF models. The AUC was good enough to be utilized in mainstream applications. As such, it would be interesting to compare the results of CBCF in the health domain with datasets in common RS domains that are studied extensively.

The first common RS datasets that was utilized was the MovieLens dataset that gives movie ratings for recommendations. This dataset utilizes a rating system similar to that of FairGRecs to which the user, items, and ratings were found in the ratings file with user ID and movie ID, and the item features were found in the movies data file with title and genre. This dataset reflects a real dataset in terms of data sparsity.

The second common RS dataset that was utilized was the StackExchange dataset that was taken off the Stackoverflow website. This dataset also had a votes dataset which incorporated the users, items, and ratings with user ID, posts, and votes. The item features were found in the answer's dataset. This dataset consists mainly with cold start issues where in many cases, there exists questions in areas without any previous expert user answers.

CBCF was set up in the same manner as the earlier tests with the same parameter settings for each of the data domains that were tested, and the cross-validation results were noted under the AUC metric for 30 epochs.

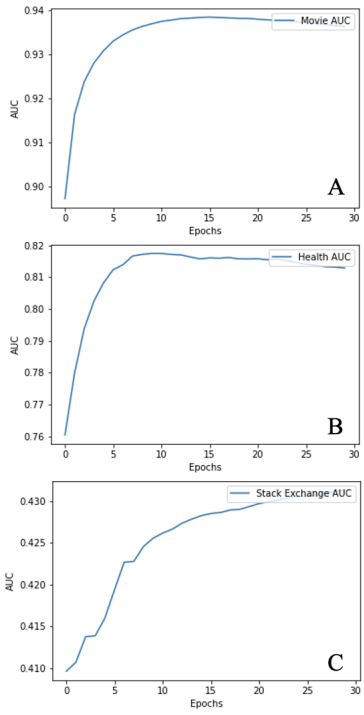

Figure 11: CBCF RS compared on datasets where A is MovieLens, B is FairGRecs, and C is StackExchange

Table 2: Optimal AUC from CBCF on Different Datasets

Dataset	Best AUC
MovieLens	0.93897
FairGRecs	0.81752
StackExchange	0.43388

The results showcase a difference in performance between all three datasets. All three datasets had logarithmic growth but at different AUCs. The MovieLens dataset performed the best with a optimal AUC of 0.93897, followed by the FairGRecs with a optimal AUC of 0.81752 and StackExchange with a optimal AUC of 0.43388 respectively. Intuitively, this result showcased the difference that CBCF has on matrix sparsity versus cold start problem and the effectiveness of the naïve privacy implementation of a synthetic dataset . The MovieLens dataset, being affected most by matrix sparsity, performed the best, followed by the FairGRecs dataset, which simulated both matrix sparsity and cold start problems, and finally followed by the StackExchange dataset, which was affected significantly by cold start. As such, it was notable that CBCF functioned better to solve the matrix sparsity issue compared to the cold start issue, especially comparing the big drop in AUC from MovieLens and FairGRecs to StackExchange. The reason why the CBCF model performed well in light of varying sparsity was due to the fact that it contained both matrix factorization when the data was dense and a pure CBF model for when the data was sparse. However, matrix factorization still suffered from cold start and the solution of representing latent vectors as linear combinations that could be estimated by new, cold-start users was too simple of a method. Hence, cold-start users were not able to receive appropriate recommendations as compared to other users. Nevertheless, for general datasets where cold start was not the main topic of concern, CBCF was applicable.

7 CONCLUSION

This research study aimed to explore the probability of RS in the health domain. Despite the existence of current HRS, it's capability to recommend medical diagnosis was not fully documented. The many reasons that may account for this issue includes the worry of patient privacy and the inability of RS to capture sparse and cold start problems. To solve the first problem, a synthetic dataset called FairGRecs was utilized and was appropriately fitted into a HRS. For the second issue, a CBCF model under the LightFM package was utilized and was properly implemented into the health domain as its performance highly exceeded those of conventional CF models. Further testing of the model on other commercial datasets proved that the model performed well in highly sparse matrices. However, despite performing significantly better than conventional CF models with the cold start problem, it was still not optimal as its performance faltered under datasets that suffered heavy cold start issues.

This study was the start to an exploration into the ability of the CBCF HRS for medical diagnosis. However, this model could still be tested and improved upon due to the current simplicity of implementation. First, an improvement could be made for the cold start problem as the current implementation of linear combination of latent vectors was too simple. One way would be to utilize deep learning architecture to enable to exploitation of nonlinear combination of latent vectors. Second, all current baseline model comparisons were to that of CF models. As such, further comparisons could be made with CBF models to conclude the advantages CBF/CF has over CF/CBF and the advantages CBCF has over both models or vice versa. Similar to the last point, the model parameters could be further tested to note the best optimizations of the CBCF model as a HRS. Finally, the current FairGRecs dataset took synthetic patient data from a bot to ensure that patient privacy was not violated. Utilizing this dataset showed the possibility of CBCF in HRS with its high AUC, however, it may not translate directly to that of real health datasets. It would be interesting to utilize perturbation methods on real patient dataset to be fed into the CBCF model.

REFERENCES

[1] Thomas George and S. Merugu. 2005. A scalable collaborative filtering framework based on co-clustering. *Fifth IEEE International Conference* (2005). https://doi.org/10.1109/ICDM.2005.14

[2] Felix Graber, Hagen Malberg, Sebastian Zaunseder, Sefanie Beckert, Denise Kuster, Jochen Schmitt, and Susanne Abraham. 2017. Neighborhood-based Collaborative Filtering for Therapy Decision Support. *Second International Workshop on Health Recommender Systems* (2017).

[3] Maciej Kula. 2015. Metadata Embeddings for User and Item Cold-start Recommendations. *Lyst* (2015). https://doi.org/10.48550/arXiv.1507.08439

[4] Defu Lian, Qi Liu, and Enhong Chen. 2020. Personalized Ranking with Importance Sampling. *International World Wide Web Conference* (2020). https://doi.org/10.1145/3366423.3380187

[5] Paritosh Nagarnik and A. Thomas. 2015. Survey on recommendation system methods. *2015 2nd International Conference on Electronics and Communnication Systems (ICECS)* (2015). https://doi.org/10.1109/ECS.2015.7124857

[6] Al Mamunur Rashid, Shyong K. Lam, Adam Lapitz, Georgia Karypis, and John Riedl. 2008. Towards a Scalable kNN CF Algorithm: Exploring Effective Applications of clustering. *Web Mining and Web Usage Analysis* (2008).

[7] Ben Schafer, Dan Frankowski, and Shilad Sen. 2007. Collaborative Filtering Recommender Systems. *The Adaptive Web* (2007).

[8] Meenakshi Sharma and Sandeep Mann. 2013. A Survey of Recommender Systems: Approaches and Limitations. *International Journal of Innovations in Engineering and Technology* (2013).

[9] Maria Stratigi, H. Kondylakis, and K. Stefanidis. 2018. The FairGRecs Dataset: A Dataset for Producing Health-related Recommendations. *SWH@ISWC* (2018).

[10] Poonam Thorat, R Goudar, and Sunita Barve. 2015. Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications* (2015).

[11] Thi Ngoc Trang Tran, Alexander Felfernig, Christoph Trattner, and Andres Holzinger. 2020. Recommender systems in the healthcare domain: state-of-the-art and research issues. *Journal of Intelligent Information Systems* (2020).

[12] Manolis Vozalis, Angelos Markos, and Konstantinos Margaritis. 2009. Evaluation of standard SVD-based techniques for Collaborative Filtering. *9th Hellenice Europen Research on Computer Mathematics and its Applications Conference* (2009).

[13] Yin Zhang, Daqiang Zhang, Mohammad Mehedi Hassan, Atif Alamri, and Limei Peng. 2014. CADRE: Cloud-Assisted Drug REcommendation Service for Online Pharmacies. *Mobile Networks and Applications* (2014).