Generalized Cross-Multi Domains Recommendation System

Github Link - https://github.com/udit19120/IR_Project

Group 21

Aabhaas Batra (2019001) (aabhaas19001@iiitd.ac.in)

Eeshaan Ravi Tivari (2019465) (eeshaan19465@iiitd.ac.in)

Jahnvi Kumari(2019469) (jahnvi19469@iiitd.ac.in)

Rishi Singhal (2019194) (samyak19194@iiitd.ac.in)

Soham Das (2019477) (soham19477@iiitd.ac.in)

Udit Narang(2019120) (udit19120@iiitd.ac.in)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

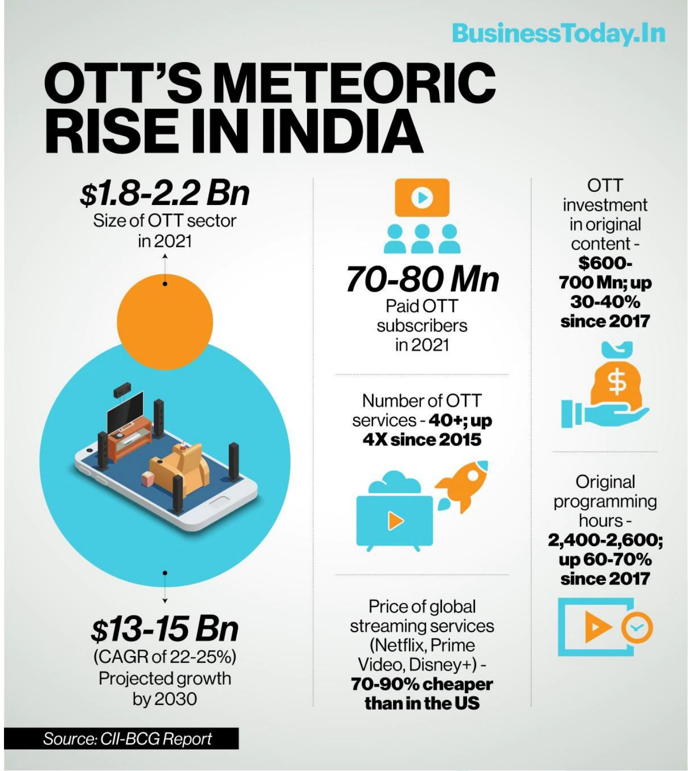
Introduction

- The goal of a recommender system is to provide personalized recommendations to users by predicting what they are most likely to prefer among a large set of items.
- These systems are used by social media, shopping and streaming applications like youtube, amazon, etc.

- With over 2.5 billion users of shopping websites and streaming services worldwide, there is a dire need to improve the cross domain recommendation(CDR) systems as time passes and more users onboard.

Continued …

- However, most of the existing work in this domain has been done with two domains only, i.e taking the user preferences data for two domains.
- And hence, there is a lot of potential in optimizing the performance of the existing recommendation systems if we take the user interaction with multiple domains into consideration.



Motivation

- Data sparsity in Recommendation systems is a common problem. This may arise due to a user interacting with an item (say, buying a dress) in a domain different than what she usually interacts in (say, buying books).
- Cross-domain representations can be used to transfer the knowledge of user-item interaction in interacted domains to a sparse domain.
- While work has been done on bi-domain models, generalizing them to k domains can help find the optimal number of domains to consider for representation learning.

Novelty

- Our work generalizes bi-domain representation learning models to k domains.
- We show that working with weighted edges in user-item interaction graphs gives more information about a user's likes and dislikes.
- We propose methods to compare the performance of our multi domain models with differing number of source domains. This aids in comparing our results with previous literature (predominantly on two domains).
- It will help better identify possible roadblocks for research in multi-domain models for recommender systems.

Dataset

The Amazon k-core review dataset is a collection of user reviews and ratings for products sold on Amazon containing :-

- ProductID,
- User ID,
- Review text,
- User rating,
- And other metadata.

K-Domains CDR

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Methodology

- Generalize the components of two domain CDR model to accommodate K domains
- Create a general loss function as follows

$$L = \sum_{i=1}^{k} (D_{KL}(q(Z_u^{X_i}|X_i) \parallel p(Z_u^{X_i}))) + D_{KL}(q(Z_u^S|X_1, \ldots$$

$$, X_k) \parallel p(Z_u^S)) + \sum_{i=1}^{k} (D_{KL}(q(Z_v^{X_i}|X_i) \parallel p(Z_v^{X_i}))) -$$

$$\sum_{i=1}^{k} E_{q(Z_u^{X_i}, Z_v^{X_i}|X_i) q(Z_u^S|X_1, \ldots, X_k)} (log p(A^{X_i}|Z_u^S, Z_u^{X_i}, Z_v^{X_i}))$$

$$+\beta \sum_{i=1}^{k} D_{KL}(q(Z_u^S|X_1, \ldots, X_k) \parallel q(\hat{Z}_u^S|X_i))$$

Methodology contd.

- For the K domain setup, we use NDCG and Hit Rate averaged over the K domains as our evaluation metric
- Encourages performance across all domains instead of just one target domain

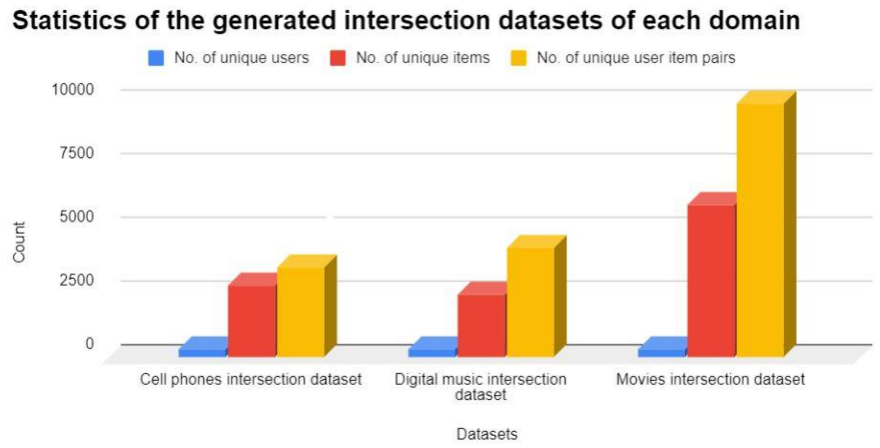
$$NDCG = \frac{DCG}{IDCG}$$

$$DCG = \sum_{i=1}^{R} \frac{Rel_i}{log(r_i + 2)}$$

$$Hit\text{−}Rate = \begin{cases} 1, & \text{if relevant item lies in top-k recommendation} \\ 0, & \text{otherwise} \end{cases}$$

Dataset Preprocessing

- Initially, we had datasets for three domains, i.e., cell phones, digital music, and movies.
- Our aim was first to find the common users who had given ratings for products in each domain and then create datasets for each domain that had the common users and the items in that domain they interacted with, respectively.
- To achieve our aim, we first removed the tuples of the products with less than ten ratings from each data set. Then we found the common users by finding the common UserIDs that existed in the datasets of all domains. After all the iterations, we found 681 common users from all the domains.
- Now to generate the intersection datasets, we, one by one, picked the dataset of each domain and found the intersection of it with the common users' data based on ProductID.



Experiments

- We compare the mean NDCG of the three domains from our model with the mean target NDCG of the results from the two domain models, trained on the six possible permutations of source and target domains from the three available domains. Likewise, we also compare the mean HIT values.

- There are many hyperparameters to be tuned, so we do a greedy search for the optimal settings, fixing one hyperparameter value at a time while keeping the others constant.

Results

Experiment Config	Domain 0 NDCG	Domain 0 Hit Rate	Domain 1 NDCG	Domain 1 Hit Rate	Domain 2 NDCG	Domain 2 Hit Rate	Mean NDCG	Mean Hit Rate
default	0.0627	0.1415	0.069	0.1511	0.047	0.1082	0.05956666667	0.1336
dropout 0	0.0758	0.1698	0.0447	0.1066	0.0917	0.1701	0.07073333333	0.1488333333
dropout 0, hdim=fdim=32	0.076	0.1415	0.0558	0.1155	0.0589	0.1134	0.06356666667	0.1234666667
dropout 0, hdim=fdim=64	0.0788	0.1509	0.0621	0.1288	0.061	0.134	0.0673	0.1379
dropout 0, hdim=fdim=256	0.0865	0.1509	0.0806	0.1822	0.1547	0.2938	0.1072666667	0.2089666667
dropout 0, hdim=fdim=512	0.0898	0.1698	0.089	0.1777	0.1287	0.2577	0.1025	0.2017333333
dropout 0, hdim=fdim=256, beta=0.1	0.0717	0.1509	0.0587	0.1377	0.1502	0.268	0.09353333333	0.1855333333
dropout 0, hdim=fdim=256, beta=0.5	0.0827	0.1698	0.06971	0.1644	0.1287	0.2835	0.09370333333	0.2059
dropout 0, hdim=fdim=256, beta=1.5	0.0938	0.1792	0.0811	0.1688	0.1534	0.2989	0.1094333333	0.2156333333
dropout 0, hdim=fdim=256, seed=0	0.0411	0.1132	0.0919	0.1777	0.1686	0.3247	0.1005333333	0.2052
dropout 0, hdim=fdim=256, n_epochs=100	**0.0877**	**0.1792**	**0.0877**	**0.1866**	**0.1695**	**0.2938**	**0.1149666667**	**0.2198666667**

Hyperparameter tuning using greedy search on 3 domain cross recommendation model

Comparison with baselines

Source Domain	Target Domain	Target Domain Hit Rate	Target Domain NDCG
Cellphones	Digital Music	0.1289	0.0598
Digital Music	Cellphones	0.1792	0.0778
Digital Music	Movies	0.2577	0.1235
Movies	Digital Music	0.1467	0.0642
Cellphones	Movies	0.2268	0.1093
Movies	Cellphones	0.1321	0.0623
	Mean values	0.1785666667	0.08281666667

Baseline results for 2 domain recommendations using Bi-TGCF (default settings)

Source Domain	Target Domain	Target Domain Hit Rate	Target Domain NDCG
Cellphones	Digital Music	0.1377	0.0584
Digital Music	Cellphones	0.1509	0.096
Digital Music	Movies	0.1391	0.0671
Movies	Digital Music	0.1111	0.0476
Cellphones	Movies	0.0927	0.0415
Movies	Cellphones	0.1415	0.0822
	Mean values	0.1288333333	0.06546666667

Baseline results for 2 domain recommendations using disenCDR (default settings)

Analysis

- From the experiments, we see that increasing the hidden and feature dimensions to 256 increases the performance across domains. The rest of the hyperparameters work the best at their default values.

- We see that in the 2 domain setting, BiTGCF performs slightly better than disenCDR for our dataset in terms of mean NDCG and HIT. In terms of both mean NDCG and HIT metrics, our model outperforms the results for the three available domains obtained using the existing two domain models.

- Our method generates embeddings for multiple domains from a single model and improves the predictive performance for each domain by extracting relevant information from all domains.

Sentiment-Based CDR

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Sentiment Based CDR

The previous experiments does not take into account the sentiment of the user-item interaction

User-item interaction represented using positive and negative labels

For k-cross domain:

Positive label: 1 (if the user has interacted with a particular item)

Negative label: 0 (if the user hasn’t interacted with the item)

For sentiment-based CDR:

Positive label: sentiment corresponding to the user-item interaction

Negative label: {0, 0.5, mean}

Sentiment based Loss

Loss function

$$
\begin{aligned}
L = D_{KL}(q(Z_u^X|X) \,||\, p(Z_u^X)) + D_{KL}(q(Z_v^X|X) \,||\, p(Z_v^X)) + \\
D_{KL}(q(Z_u^Y|Y) \,||\, p(Z_u^Y)) + D_{KL}(q(Z_v^Y|Y) \,||\, p(Z_v^Y)) + \\
D_{KL}(q(Z_u^S|X,Y) \,||\, p(Z_u^S)) + \beta D_{KL}(q(Z_u^S|X,Y) \,||\, q(\hat{Z}_u^S|X)) + \\
\beta D_{KL}(q(Z_u^S|X,Y) \,||\, q(\tilde{Z}_u^S|X)) + BCELoss(p_{\theta^X}, A^X) + \\
BCELoss(p_{\theta^Y}, A^Y)
\end{aligned}
$$

A^X and A^Y matrices which represents the user-item interaction for each user-item pair now contains sentiment for user-item interaction instead of binary values for Sentiment based CDR

Sentiment-Based Evaluation Metrics

$$NDCG = \frac{DCG}{IDCG}$$

$$DCG = \sum_{i=1}^{R} \frac{Rel_i}{log(r_i + 2)}$$

$$Hit\text{-}Rate = \begin{cases} 1, & \text{if relevant item lies in top-k recommendation} \\ 0, & \text{otherwise} \end{cases}$$

Here Rel_i represents the sentiment score of the user-item interaction, i.e., range of [0,1]

Dataset Preprocessing

- As part of the Sentiment-based CDR model we have consider 2 domains - Cellphones and Digital music of the 5-core Amazon Review Dataset.
- Each entry had a user-item pair with a corresponding review and customer rating.
- Thus we find three types of sentiments scores - 1(base case), VADER-sentiments and customer ratings.
- The VADER-based sentiments are found using VADER compound score of the customer review and is then scaled to 0 to 1. To apply the VADER model we apply basic text preprocessing techniques like punctuation removal, stopwords removal, stemming.
- Similarly the customer ratings are scaled to the range 0 to 1 from 0 to 5.

Experiments

We ran multiple experiments with following different types of weighing scheme for our model -

1. Scaled Sentiment scores
2. 1 based weights
3. Scaled Ratings

For all these we tried varying values of negative labels including 0, 0.5 and mean of all sentiment scores.

Hyper-parameters tuning: We further do hyper-parameter tuning of the model from above experiments which include changing the learning rate & decay, dropout and beta value.

Results

Results by keeping the default parameters of learning rate = 0.001, lr_decay = 0.03, beta_value = 0.9 and dropout = 0.3 and varying the sentiment wts and negative label values.

For Cellphones - digital music :-

Weights	Neg Label Values	Source Hit Rate	Source NDCG	Target Hit Rate	Target NDCG
		DisenCDR	DisenCDR	DisenCDR	DisenCDR
Sentiment - Scaled	0	0.0787	0.0347	0.2158	0.0971
	0.5	0.0820	0.0760	0.1216	0.1141
	Mean	0.0492	0.0482	0.1147	0.1103
Sentiment - 1	0	0.1148	0.0495	0.1298	0.0615
	0.5	0.1541	0.1381	0.1987	0.2111
	Mean	0.1178	0.1178	0.1389	0.1338
Rating Scaled	0	0.1209	0.0550	0.1254	0.0629
	0.5	**0.1634**	**0.1504**	**0.2474**	**0.2274**
	Mean	0.1051	0.1031	0.1076	0.1062

For digital music - Cellphones :-

Weights	Neg Label Values	Source Hit Rate	Source NDCG	Target Hit Rate	Target NDCG
		DisenCDR	DisenCDR	DisenCDR	DisenCDR
Sentiment - Scaled	0	0.1815	0.0821	0.1672	0.0732
	0.5	0.1233	0.1167	0.1443	0.1360
	Mean	0.1027	0.0989	0.1148	0.1124
Sentiment - 1	0	0.1280	0.0517	0.1450	0.0665
	0.5	0.2232	0.2024	0.1582	0.1319
	Mean	0.1230	0.1186	0.1631	0.1631
Rating Scaled	0	0.1254	0.0533	0.1046	0.0522
	0.5	**0.1271**	**0.1152**	**0.1601**	**0.1463**
	Mean	0.0829	0.0820	0.1083	0.1065

Hyper-parameter Tuning

A) Varying learning rate

Weights	Learning rate	Source Hit Rate	Source NDCG	Target Hit Rate	Target NDCG
DisenCDR (Rating - Scaled with neg label = 0.5)	0.0001 (decay=0.1)	0.0850	0.0785	0.1082	0.0991
	0.00001 (decay=0.1)	0.0882	0.0816	0.1065	0.0977
	0.0001 (decay=0.05)	**0.0752**	**0.0690**	**0.1203**	**0.1096**
	0.00001 (decay=0.05)	0.0882	0.0816	0.1065	0.0977

B) Varying beta

Weights	Beta rate	Source Hit Rate	Source NDCG	Target Hit Rate	Target NDCG
DisenCDR (Rating - Scaled with neg label = 0.5)	0.5	0.0852	0.0790	0.1182	0.112
	0.7	0.0852	0.0793	0.1164	0.1098
	1	**0.0787**	**0.0731**	**0.1216**	**0.1143**
	1.5	0.0854	0.0790	0.1182	0.1112

C) Varying dropout rate

Weights	Dropout rate	Source Hit Rate	Source NDCG	Target Hit Rate	Target NDCG
DisenCDR (Rating - Scaled with neg label = 0.5)	0.5	0.1176	0.1072	0.1976	0.1820
	0.2	0.1634	0.1503	0.2595	0.2387
	0.1	0.1601	0.1466	0.2629	0.2418
	0	**0.1961**	**0.1809**	**0.2698**	**0.2490**

Comparisons of our best model with Bi-TGCF and base DisenCDR model

For Cellphones - digital music :-

Weights	Source Hit Rate	Source NDCG	Target Hit Rate	Target NDCG
DisenCDR(Rating - Scaled)	**0.1961**	**0.1809**	**0.2698**	**0.2490**
DisenCDR(Base - Model)	0.1541	0.1381	0.1987	0.2111
Bi-TCGF	0.1353	0.0643	0.2388	0.1543

For digital music - Cellphones :-

Weights	Source Hit Rate	Source NDCG	Target Hit Rate	Target NDCG
DisenCDR(Rating - Scaled)	**0.2165**	**0.1990**	**0.1863**	**0.1695**
DisenCDR(Base - Model)	0.2232	0.2024	0.1582	0.1319
Bi-TCGF	0.1221	0.1119	0.1603	0.0758

Analysis

- We observe that using user ratings as sentiments are giving better results than sentiment one and VADER sentiment. Also, these results outperform our baselines as well.

- Also, the user ratings depict a true picture of the user experience in a user-item interaction. Using VADER sentiments is not better results than the user-ratings, which may be due to the fact that the amazon dataset contains reviews in foreign languages also like Spanish, French.

- Furthermore, using sentiment equal to one to all user-item interactions isn't correct as different users can have different experiences with a product.

- In conclusion, we can clearly say that making use of user sentiments is an essential aspect of a cross-domain recommendation as explained above and also visible in the final comparision tables in the previous slide.

Future Work

- In this work, we have proposed two mutually exclusive approaches, one using k-domains and the other using the sentiment of the user-item interaction. Future work can combine both approaches into one single approach that takes into account both multiple domains and sentiment.

- Furthermore, we would also like to take into account more sophisticated sentiment analysis models that work well for multiple languages.

- Also, for the k-domains formulation we would like to try out the model for higher values of k. For this, we would need to acquire and work on larger datasets such that the joint intersection between the domains has sufficient number of data points.

Contribution

- Aabhaas, Jahnvi, Soham - K-domains methodology setup and experiments.
- Eeshaan, Rishi, Udit - Sentiment based CDR methodology setup and experiments.
- Equal contribution in PPT, Video and Report making.

References

[1] Liu, Z., Zheng, L., Zhang, J., Han, J. and Philip, S.Y., 2019, December. JSCN: Joint spectral convolutional network for cross domain recommendation. In *2019 IEEE international conference on big data (big data)* (pp. 850-859). IEEE.

[2] Kuang, H., Xia, W., Ma, X. and Liu, X., 2021, March. Deep matrix factorization for cross-domain recommendation. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (Vol. 5, pp. 2171-2175). IEEE.

[3] Cao, J., Lin, X., Cong, X., Ya, J., Liu, T. and Wang, B., 2022, July. DisenCDR: Learning Disentangled Representations for Cross-Domain Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 267-277).

[4] Wang, Y., Yu, H., Wang, G. and Xie, Y., 2020. Cross-domain recommendation based on sentiment analysis and latent feature mapping. *Entropy*, *22*(4), p.473.

[5] Lu, Z., Zhong, E., Zhao, L., Xiang, E.W., Pan, W. and Yang, Q., 2013, May. Selective transfer learning for cross domain recommendation. In *Proceedings of the 2013 SIAM International Conference on Data Mining* (pp. 641-649). Society for Industrial and Applied Mathematics.

[6] Zhao, C., Li, C. and Fu, C., 2019, November. Cross-domain recommendation via preference propagation graphnet. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 2165-2168).

[7] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering.In proceedings of the 25th international conference on world wide web. 507–517

Thank you

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI