Review, revision, and enhancements of CB2CF: A Neural Multiview Content-to-Collaborative Filtering Model for Completely Cold Item Recommendations

Group 16- Ofer Inbar 033856402, Alex Tsvetkov 322155516

Link for the repo containing the code and other sources for this report [Recommendation-Systems/Project at master · tsalex1992/Recommendation-Systems (github.com)]

In this paper, we review the concept of the CB2CF model, which allows for improved performance of cold start item prediction by projecting their content representation towards the space that would represent those items in the collaborative filtering space. We then propose improvements for the CB2CF model and present an Auto Regression based Next Item (**ARNI** 😊) Predictor

CB2CF: A Neural Multiview Content-to-Collaborative Filtering Model for Completely Cold Item Recommendations

Paper overview

In their work, the writers combine two distinct approaches of recommender system algorithms to improve the results obtained on newly introduced items.

Recommendations made using collaborative filtering

CF algorithms designed to model and learn interactions between users and items [1–3], and are known to capture users' taste towards items, without requiring prior knowledge on the attributes of the items. It is common practice to observe user to item (either explicit or implicit) interactions, and model those interactions in a latent space, which embeds the users' and the item's representations separately. On prediction time, user representation is matched with the item's representations, producing a similarity score. The items that are recommended are those items that resemble the users profile the most.

Content-based recommendations

CB algorithms use features that are collected on items in advance to build a profile of users' taste and allow for item recommendations to be user specific. CB algorithms do not rely on similarity between users at all and can provide good results for newly added items (for as long as the new items carry the required features to form the content related profile). Overall, CF algorithms outperform the CB algorithms [5], but for the case of cold-starting items(items without prior user rating), CF algorithms fall short, as there are no user to item interactions to infer the newly added items' embeddings.

CF vs. CB

Objective

The cold start problem is the main motivation for the CB2CF model described in this paper- improving the accuracy of the recommendations made for newly added items, where only their content-based information is available. This is achieved by observing the relationship between the CB representation of items that already have a CF representation. This relationship is then learned and applied on newly added items, for which only the content-based information is available, producing a CF view of their CB representation.

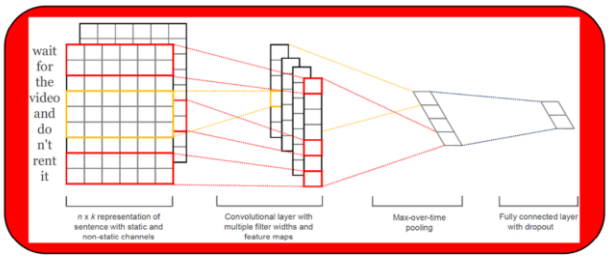
Method

Creating the CF representation of items

The algorithm chosen for the CF representation is BPR - [4] The Bayesian Personalized Ranking algorithm. The BPR produces representations of users and items using explicit and implicit interactions. Both the explicit (positive) interactions are being used, and implicit interactions are considered items that users did not interact with. BPR aims to maximize the difference between the scores it produces for positive and negative items. For this work, the researchers chose BPR, but have explained that the same representation could have been achieved by other CF methods. While a typical BPR application would be used to produce both users and item vectors to produce predictions, this implementation of the model Is designed to produce item mappings only, to be later used as target for the CB2CF model.

Creating the CB representation of the items

The CB representation is built from several data sources, each data source is treated differently. Two different approaches were taken for the handling of **textual sources**, such as plot descriptions and teasers.
With the **first approach- (CNN,** m_{w2v}**)** , texts are treated with a word2vec, where the first l words are given an m dimensional representation. Each of the vectored representations of the first l words of the textual sources is later used as input for training of the convolutional neural network (CNN), so it allows for improving the word2vec representations as these can be learned during propagation.



With the second approach (BOW), each word is clustered using k-means clustering algorithm that was executed on the entire vocabulary. Each word in the text is represented by a 1-hot cluster ID from one of the b k − means clusters. Eventually, the first l words of the text are represented as a histogram describing the frequency of each of the words using their b-clustered affiliation.

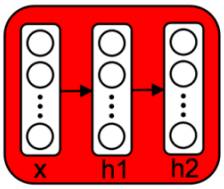
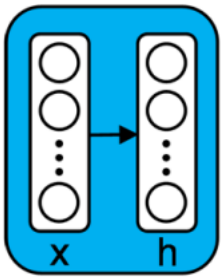


Figure 1 The Bow Component

Semi-structured sources, such as tags, categories, genres, actors and languages are represented as a 1 hot vector.



Numerical values are represented as such. The only numerical value used in this implementation is the item's release year.
The 3 different data types are brought together to be used as base for the multiview mapping between the CB and the CF representations. This task

is performed by the **combiner**, which concatenates the different data representations, to be used as an input of a neural network model that produces a vector that is of the same dimension of the CF representation of the item.

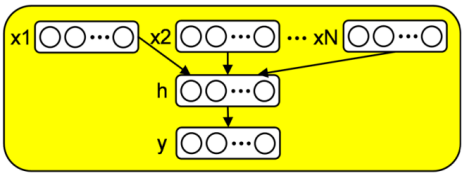
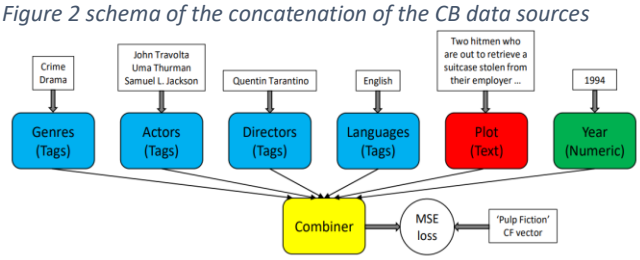


Figure 2 schema of the concatenation of the CB data sources



CB2CF experiments and evaluation

Two BPR models were constructed for two different purposes. The BPR1 model is trained on the entire training dataset, to be used as a base for evaluating the CB2CF model. As this model is exposed to items that are hot, we can later compare item vectors produced by the BPR1 model, to the item vectors produced by the CB2CF model for the cold items. The most important difference is that the CB2CF model will be trained on the mapping of the CB to a CF representation of a sub-set of items called BPR2. We then compare the similarity of the representation of the items made by the CB2CF model on items that were not present in BPR2 training data, with the representation made by the BPR1 model, that has these items' rating history.

Data Splitting

The ratings are split twice; once by users, to form a user train set to feed the BPR model that produces the CF item vectors, and the set of users P, which is kept for testing purposes. Second split is done at the item level, where train items are used for training the CB2CF and test items T used for testing it.

Evaluation measures

MSE (Mean Square Error)- Item vectors produced by the different models are compared using MSE. This is the same objective of the CB2CF model- to reconstruct the CF item vector for items the CB2CF model has only seen content features for:

$MSE = \frac{1}{|\tau|}\sum_{i\in\tau}||v_i - \hat{v_i}||^2$, where τ is the set of all items as described above, $\hat{v_i}$ is the vector predicted by the CB2CF model, and v_i is the value produced by the CF model.

Top-K Mean Accuracy. Top-K is determined as 1 if for a given rating the prediction of the CB2CF includes the next item rated by the user in the top K recommendations, else 0. The Top-K Mean Accuracy is achieved by averaging all Top-k values, for all next item pairs in τ.

MPR- Mean percentile rank. Percentile rank is calculated as the rank of the next item rated in the result of the prediction made by CB2CF, normalized by the size of the item domain M. For the case of a perfect match, an item is predicted as index 0, and PR = $\frac{0}{M}$. At the worst case, the correct item would be last to be recommended, and $PR = \frac{M-1}{M}$. MPR is calculated as the mean PR for all items in τ.

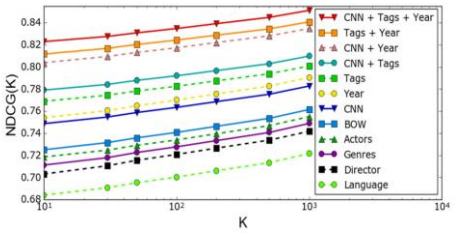
Experiment 1 – CF Space Reconstruction (Ablation Study)

In this experiment, the researchers aim to evaluate the contribution of each of the different data sources used in the CB representation to the accuracy of the reconstruction of the item representation space produced by the CB2CF when checked for similarity for the representation produced by the BPR1 model.

The main conclusion from this experiment is that while all data is helpful, the CNN for the Word2VEC outperformed the BOW approach. It was decided to drop the BOW approach in favor of the CNN. The writers explain that the main benefit of the CNN for the word2vec is that this representation can learned as part of the entire CB2CF model.

System	MSE / MPR
Language	23.1 / 40.8
Director	22.2 / 34.3
Actors	21.6 / 25.5
Genres	21.3 / 21.4
BOW	21.2 / 19.2
CNN	20.3 / 17.2
Year	19.8 / 15.4
Tags	19.2 / 12.4
CNN + Tags	18.6 / 11.2
CNN + Year	17.4 / 7.6
Tags + Year	17.1 / 6.7
CNN + Tags + Year (CB2CF)	**16.5 / 5.4**

While evaluating the contribution of the different data types to the NDCG, we can see that semi-structured data (genres, languages) + Year produces better results than text alone, and that CNN is stronger than BOW. In addition, the marginal contribution of texts to the NDCG of the Tags + Year is ~ 0.015, and that year alone is stronger than all other feature sets.



Experiment 2- Cold Starting Item Recommendation

In this experiment, the performance of cold item recommendations produced by the CB2CF model are compared to the performance of a plain CB model and the BPR1 model that was trained over the entire rating data set (all users and items). For this experiment, the CB2CF model is trained using embeddings produced by the BPR2 model, which include only train items (omitting the "cold" ones). The evaluation is performed on 2 different data sets:

Train Test – The set of consecutive ratings i, j , when either $i \; or \; j \; items \; belong \; to \; set \; T$, made by users from P(test users).

Test Only- ratings made by any user, while both consecutive items $i, j \in T$, the items test set.

Quantitative evaluation

Both the Top-20 mean and the MPR Results demonstrate that the CB2CF model outperforms the CB model on both data sets, giving better results for both cold only items and mixed items. The CB2CF model falls short of the performance of the BPR1 model, which is expected in the sense that CF outperform CB models, and that CB2CF was trained to match the embeddings of the BPR2 model, which was trained on a subset of the items given to the BPR1 model.

Model / Set	Train Test	Test Only
*BPR1 (**oracle**)*	*0.556*	*0.551*
CB2CF	**0.482**	**0.475**
CB	0.372	0.369

*Figure 3 **Top-20** mean accuracy values obtained on the "Test Only" set*

Model / Set	Train Test	Test Only
*BPR1 (**oracle**)*	*15.04*	*15.13*
CB2CF	**17.93**	**19.51**
CB	27.18	27.23

Figure 4 MPR values obtained on the "Test Only" set

Qualitive evaluation

The evaluation was performed by feeding the different models with an item from the test set. Recall that BPR1 model was given all items to train all, where the CB2CF models have not seen those items. The results shown in the table below, demonstrate the top 5 nearest items returned from each model. Similarity is determined by using cosine similarity.

Model\ Query	Shrek (2001)	The Hangover (2009)	Gladiator (2000)
BPR1 (oracle)	Monsters Inc., Shrek 2, Finding Nemo, Ice Age	Superbad, Role Models, I Love You Man, Knocked Up	The Patriot, The Last Samurai, Saving Private Ryan, Enemy at the Gate
CB2CF using CNN + Tags + Year	Shrek 2, Stuart little 2, Monsters Inc., Toy Story 2	The Hangover 2, Grown Ups, Role Models, Due Date	The 13th warrior, Story of Joan of arc, The Musketeer, The Last Castle
CB2CF using CNN only	Shrek The Third, Shrek Forever After, Shrek 2, Finding Nemo	21 jump street, The Hangover 3, The Hangover 2, Grown Ups	The 13th warrior, King Arthur, 300, Troy

Figure 5 Top-4 Recommendations produced by different models for test movies

Suggested revisions for the CB2CF model

We propose two improvements. The first improvement aspires to enhance the CB2CF item vectors representation, and the other addition suggests a method of predicting the next item based on both the CB2CF embeddings and the sequence of recent rating activity performed by the user.

Improvements towards the CB2CF embeddings

The original CB2CF model used a few data sources. Textual, categorical, and numerical (year of release). We propose changes to the handling of the representation of the release date and the textual sources.

Proposed changes to the time dimension

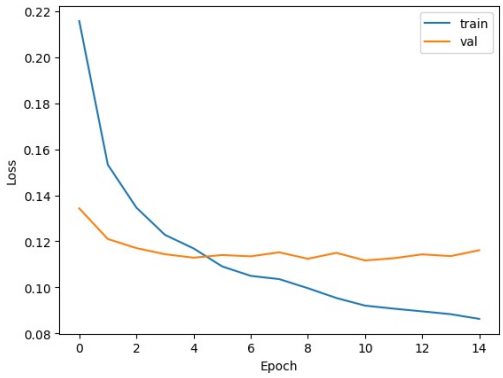
In the original work, the release date was converted into a yearly representation. The year feature alone has proven to be more accurate in terms of NDCG than the entire CNN presentation of the textual sources combined, and stronger than all components of the categorical features (Actors, Genres, Director, Language) in the ablation study done in experiment 1. We have converted the time dimension to be a continuous time representation, later to be embedded by a vector of 3 * (item embedding dim) for two reasons. The first is to enable finer granularity for the representation of the time dimension, which helps to capture the exact release date, and the second is to enable capturing complex user-time interactions which potentially require high dimensionality. Both are crucial for the usage of the ARNI model, proposed later in this paper.

Proposed changes for the text embeddings

Sentence transformers [6] are an extension of transformers [7], designed to generate vector embeddings for sentences or paragraphs of text. The key innovation of sentence transformers is their ability to encode the semantic meanings of pieces of text into a single vector representation. We have chosen to replace the word2vec based CNN representation of the textual sources used in the CB2CF model, as sentence encoders have shown to outperform word2vec [8] in capturing textual information. The model used for that task is all-MiniLM-L6-v2. The embedding dimension was set at 40 to match the size of the original work.

Training the improved CB2CF

We ran 15 epochs with a learning rate of 0.001, and a batch size 32.



Evaluation of the suggested improvements for the CB2CF embeddings

We have repeated experiment 1, with the time dimension converted to Unix epoch time, and the textual source represented by the sentence transformer. Results shown in the Figure 6, give an average increase of 0.07 in the NDCG, when compared to original work's best result set (CNN + Year+ Tags) across all available Ks.

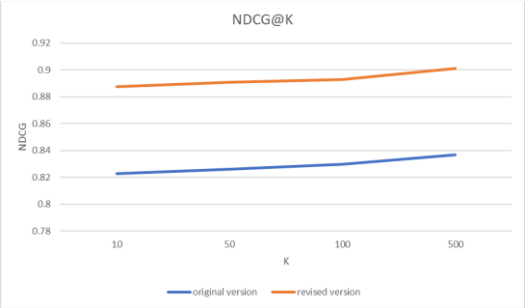


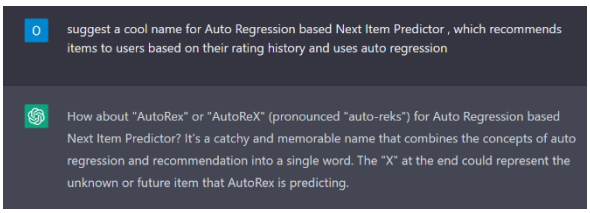
Figure 7 NDCG @K results - revised Vs. original

When comparing MSEs for the full set of data sources (CNN + Year + Tags) Vs. revised model (Sentence transformer + timestamp+ Tags), we get an improved MSE (X 100) of 12.7, Vs. the MSE given on the original work of 16.5 demonstrating a drop of 23% in MSE for prediction of the oracle embeddings on the test items.

	MSE
original version	16.5
revised version	12.7

Figure 8 MSE results - revised Vs. original

Auto-regressive next item (ARNI) predictions



Decoder-based transformer language models are designed to predict the next word, based on context [9], and have attention mechanisms that allow for understanding trends and seasonality. We demonstrate a framework that extends CB2CF and other recommendation systems, by treating the sequence of user's item history as incomplete sentences, while the next item to be recommended is the next word. This framework is also both context and content aware, as it is fed with the embeddings from the CB2CF model. Our model aims to estimate the following probability function: $P(X_{i+1}|X_{1:i}, U_i, t_{i+1}, t_{1:i})$
Which is the prediction of the next item, give the sequence of i previous items, i_{th} user, current rating time and previous i rating times.

Data used

We use the user embeddings of the BPR2 model, the item embeddings from the CB2CF (trained on the BPR2 items), the ratings from the Movie Lens 20M (same ratings used for the BPR model) converted to sequences. Each sequence of user ratings was trimmed to be maximum of 53 most recent ratings. The last 3 ratings of each user were saved as test data, and

the rest of the ratings were treated as train data. To simulate the cold start scenarios, we have removed all test items from the sequences.

Model Input

The decoder transformer gets an input which consists of a SOS token and a sequence of movie ids. Later on they are converted to the pre-trained embedding from the CB2CF model, while the first token is converted into the user embedding from BPR2. To these embeddings we add the traditional learned positional embeddings. Later on we also add time embeddings of the rating time(Unix epoch) – while for the SOS token the prediction time embedding is added. For sequences shorter than 50, we have placed tokens of "End of Sequence".

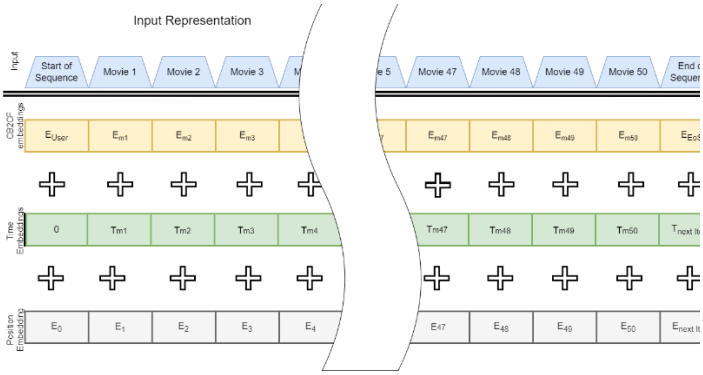
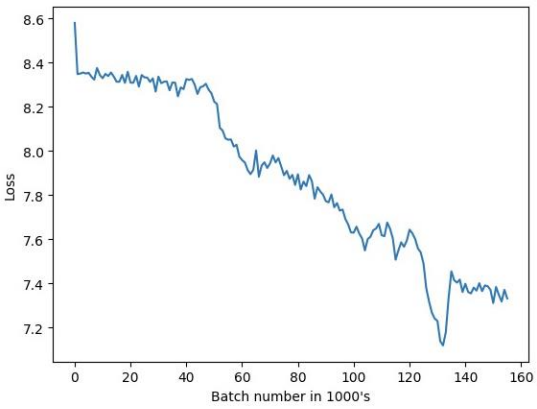


Figure 9 input representation schema of the sentence transformer

Training

We built an iterator to produce the data required for the training, as the entire set was too big to fit into memory. Each user's last 50 ratings were sorted by their time, and then the learning started predicting the next item according to the items it has seen thus far. The prediction produces a vector of probabilities for all items, and a Cross entropy loss is calculated between the target next item and the recommended item. Each batch consisted of 32 user sequences. Out of the 20M interactions, 20% were dropped as these were made on test movies. The remaining sequences were then trimmed to 50 interactions at most, leaving us with 6.4M predictions made during the training phase.



Quantitative Results

For the quantitative results, we chose the MPR metric. Due to lack of resources (time), we randomly selected 5000 users, and measured their performance. For these users, we have obtained an MPR score of 19.85, which is very close to the MPR achieved by CB2CF (19.51) over the test data.

Qualitive Results

Case 1:

Let's start with a good example. Gladiator was successfully selected by ARNIV as first choice, after it was given the last 50 movies the user has seen, and a requested date of 22/12/2000. A quick check discovers that gladiator was released on May 1st 2000, while the 2 of the last 4 ratings made by the user were for movies released on June 2000. Seems like this user was catching up on summer block buster around Christmas holiday.

Last rated: [['Good Will Hunting'], ['Titanic'], ['Suicide Kings'], ['Deep Impact'], ['The Land Girls'], ['Smoke Signals'], ['Armageddon'], ['Saving Private Ryan'], ['Practical Magic'], ['Elizabeth'], ['Shakespeare in Love'], ['20 Dates'], ['Night Shift'], ['Analyze This'], ['Edtv'], ['The Matrix'], ['Election'], ['Entrapment'], ['Star Wars: Episode I - The Phantom Menace'], ['Dracula'], ['American Pie'], ['Arlington Road'], ['Ghostbusters'], ['The Sixth Sense'], ['The Thomas Crown Affair'], ['Stigmata'], ['American Beauty'], ['The Buddy Holly Story'], ['Double Jeopardy'], ['Total Recall'], ['Fight Club'], ['The Longest Day'], ['Toy Story 2'], ['The Green Mile'], ['The Cider House Rules'], ['Snow Falling on Cedars'], ['Pacific Heights'], ['Down to You'], ['Forever Young'], ['Boiler Room'], ['Mission to Mars'], ['Erin Brockovich'], ['Midnight Express'], ['Bell, Book and Candle'], ['The Patriot'], ['Air America'], ['Horror of Dracula'], ['Shanghai Noon']]
ARNI recommendations: ['Gladiator'], ['Mission: Impossible II'], ['å•§è™Žè—•é¾™'], ['X-Men'], ['Cast Away'], ['What Lies Beneath'], ['Pitch Black'], ['Sleepy Hollow'], ['The Gift'], ['Memento']

Case 2

User has rated the movie "Chocolat" on 05/07/2011. ARNI has recommended a diverse set of movies in terms of genres, but did not predict "Chocolat", perhaps because it was released on 2000-12-14, 7 months before.

Last rated: ['The Shawshank Redemption'], ['Robin Hood: Men in Tights'], ['Dangerous Beauty'], ['Never Been Kissed'], ['Where the Heart Is'], ['The Lord of the Rings: The Fellowship of the Ring'], ['The Importance of Being Earnest'], ['The Lord of the Rings: The Two Towers'], ['The Lord of the Rings: The Return of the King'], ['ãƒ•ã,¦ãƒ«ã•®å‹•ã••åŸŽ'], ['Dan in Real Life'], ['Penelope'], ['Harry Potter and the Deathly Hallows: Part 1'], ["The King's Speech"], ['Unknown'], ['Just Go with It'], ['Sucker Punch'], ['Water for Elephants'], ['X-Men: First Class'], ['Stardust']
ARNI recommendations: ['How to Train Your Dragon'], ['Despicable Me'], ['Tangled'], ['Up'], ['The Adjustment Bureau'], ['Gnomeo & Juliet'], ['Rango'], ['Paul'], ['Horton Hears a Who!'], ['Rio']

References

[1] J. Bennett and S. Lanning, "The Netflix Prize," in Proceedings of the KDD Cup Workshop 2007, 2007, pp. 3–6.

[2] R. M. Bell and Y. Koren, "Lessons from the Netflix Prize Challenge," SIGKDD Explor. Newsl., vol. 9, no. 2, pp. 75–79, Dec. 2007.

[3] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer, "The Yahoo ! Music Dataset and KDD-Cup ' 11," KDD Cup, pp. 8–18, 2012.

[4] Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L., 2009, June. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence (pp. 452-461). AUAI Press

[5] Slaney, M., 2011. Web-scale multimedia analysis: Does content matter?. IEEE MultiMedia, 18(2), pp.12-15.

[6] Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).

[7] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[8] Amin, Akhter Al, et al. "Using BERT Embeddings to Model Word Importance in Conversational Transcripts for Deaf and Hard of Hearing Users." arXiv preprint arXiv:2206.12368 (2022).

[9] Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.