PARL: Let Strangers Speak Out What You Like

Authors: Libing Wu , Cong Quan , Chenliang Li, Donghong Ji

Report by: August Soderberg and Matt Sehgal

Problem Definition

As with any problem in recommender systems, the heart of the problem at hand is how we can provide relevant recommendations to users. With many supervised learning based solutions we encounter the data sparsity problem; recommender systems are certainly no exception. Whether we are informing our recommendations based on reviews, ratings, or simple interactions, data sparsity is still a problem. In the case of reviews there is typically both a limited number of them and often they contain little useful information, in the case of ratings we still often have fewer than desired, and in the case of interactions they often fail to capture the feelings or desires of the user. The problem is: How can we provide relevant recommendations in light of sparse data?

Challenge Analysis

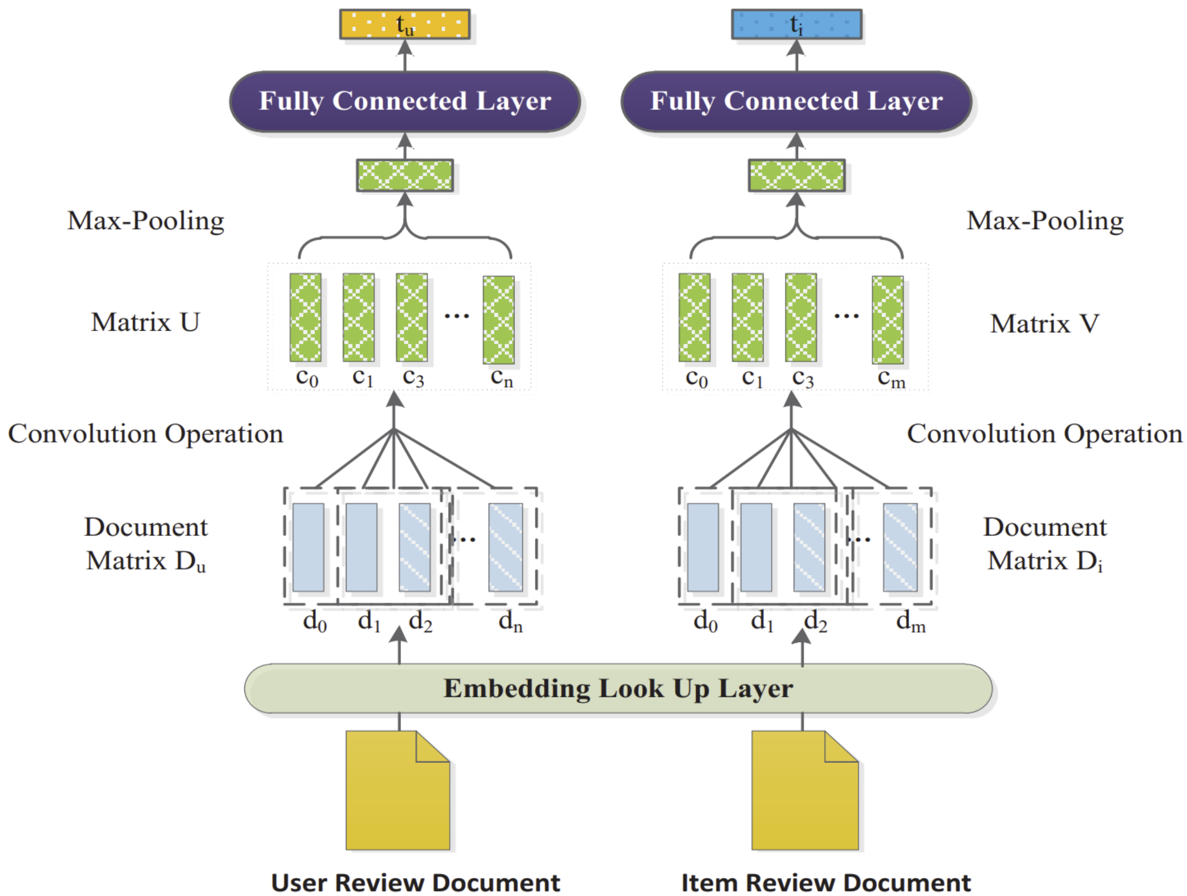
For the PARL paper, they are focusing on review-based recommender systems because of their impressive performance on even sparse datasets. The problem with review-based recommender systems is that most users do not leave any sort of reviews at all and if a review is left, it is typically short and doesn't contain much information on which to base future recommendations. The challenge here is how can we reduce the sparsity of the complex review data to ultimately create relevant recommendations using a review-based recommender model.

Philosophy

Data sparsity in rating-based collaborative filtering (CF) recommender systems is a well researched problem which can be solved using neighborhood-based methods. The methods will look at ratings made by the same user on other similar items to estimate the rating of that user on an item which they have not rated. The idea of the authors of PARL is to apply this same idea to users and their reviews of items. They can artificially increase the number of reviews in the dataset by attributing a review written by a similar user to the original user for a product which they have rated. This solution should be able to be plugged into any existing review-based recommender system model to help extract more informative features from the reviews to ultimately increase the relevance of recommendations.

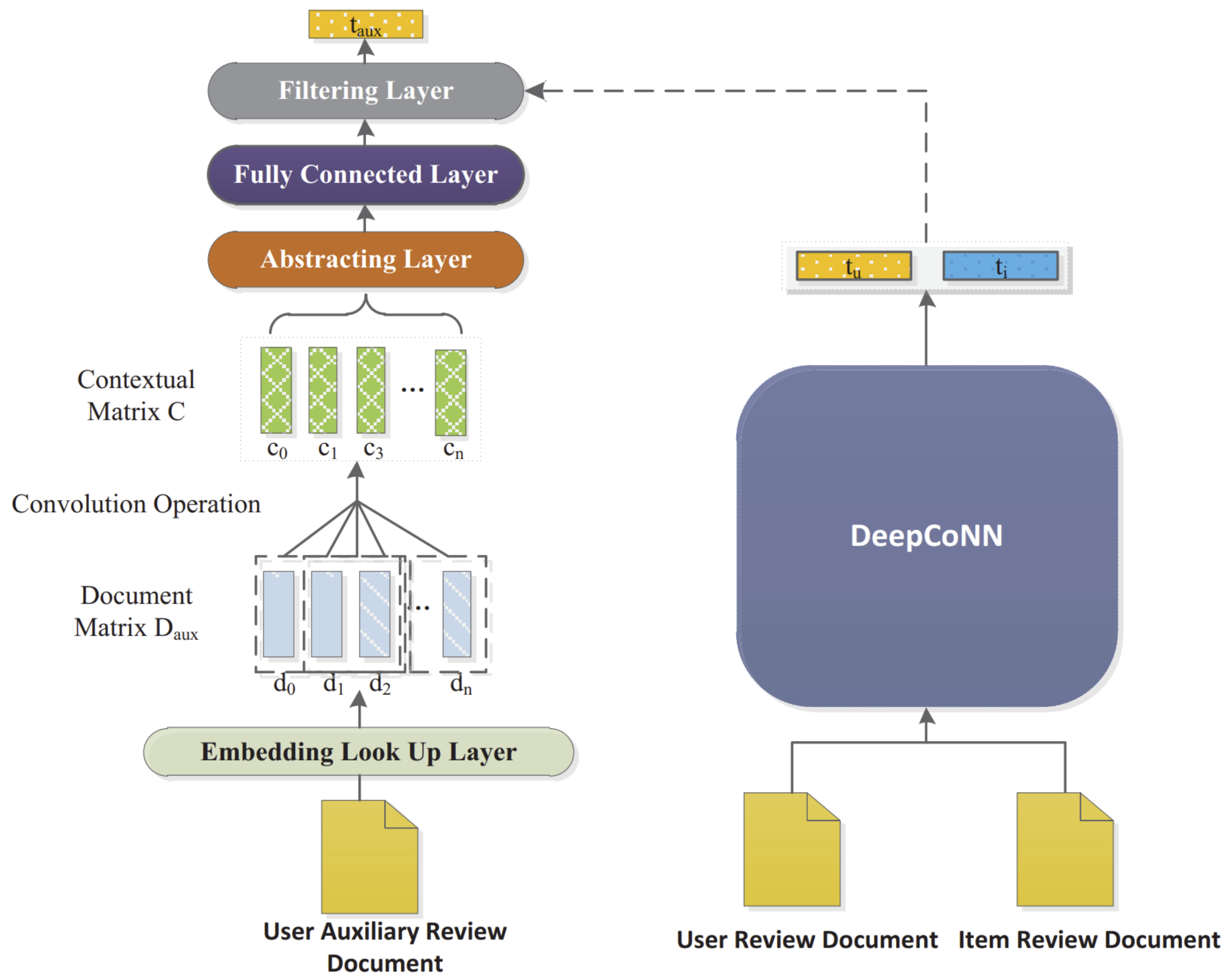
Solution

For this paper, the authors chose to use the state of the art DeepCoNN model for review-based recommendations.



This is a diagram of the DeepCoNN model which takes in a user review document and an item review document which ultimately will return two latent feature vectors, one for the item and one for the user which can be analyzed to estimate the rating of the user on the item.

PARL uses a very similar architecture integrated with DeepCoNN:



Here you can see that a third input is taken of the user auxiliary review document. In the proposed model, the user auxiliary review is created as follows. For a given user, for each item they have rated, a review will be selected from a random user who rated the item the same as the original user. If there does not exist anyone who gave the same rating, then a random review will be selected from ratings one greater than the original users rating. If such a review does not exist, then one will be randomly selected if its corresponding rating is one less than the given rating.

This proposed model working in conjunction will generate an auxiliary latent feature vector which will be combined in the filtering stage with the users latent feature vector and then finally can be compared with the item latent feature vector using a factorization machine to determine the estimated rating of the user on the item.

Experiments

The authors have selected 5 datasets on which to test their PARL + DeepCoNN model: Beer (from a beer rating website), Office Products, Digital Music, Video Games, and Tool Improvement with the last 4 datasets being 5-core datasets from Amazon. The authors compared their results to 10 other models and achieved lower MSE than all 10 other models (see table below).

Method	Beer	Office Products	Digital Music	Video Games	Tools Improvement
PMF	1.636†	1.091†	1.211†	1.669†	1.564†
SVD++	0.726†	0.771†	0.950†	1.183†	1.066†
CDL	0.678†	<u>0.754</u>†	0.882†	1.179†	1.033†
RBLT	<u>0.576</u>†	0.757†	<u>0.872</u>†	1.147†	<u>0.983</u>†
CMLE	0.607†	0.761†	0.883†	1.254†	1.023†
ConvMF	0.853†	0.960†	1.084†	1.449†	1.240†
DeepCoNN	0.617†	0.860†	1.060†	1.238†	1.063†
DeepCoNN-Aux	0.615†	0.860†	1.059†	1.236†	1.058†
TransNets	0.586†	0.760†	0.910†	1.196†	1.008†
D-attn	0.616†	0.824†	0.914†	<u>1.142</u>†	1.046†
PARL	**0.561**	**0.731**	**0.849**	**1.117**	**0.955**

Note that the best performing model is in bold and the second best performing model is underlined.

Out Improvement

- Philosophy

Our biggest problem with the implementation of the PARL model is that "like-minded" users are not truly "like-minded". If we are attempting to reflect a user's sentiments then we believe that we would be most successful using the sentiments of truly similar users. We are hoping that supplying reviews from similar users will improve the overall quality of the auxiliary reviews and thus the overall performance of the model.

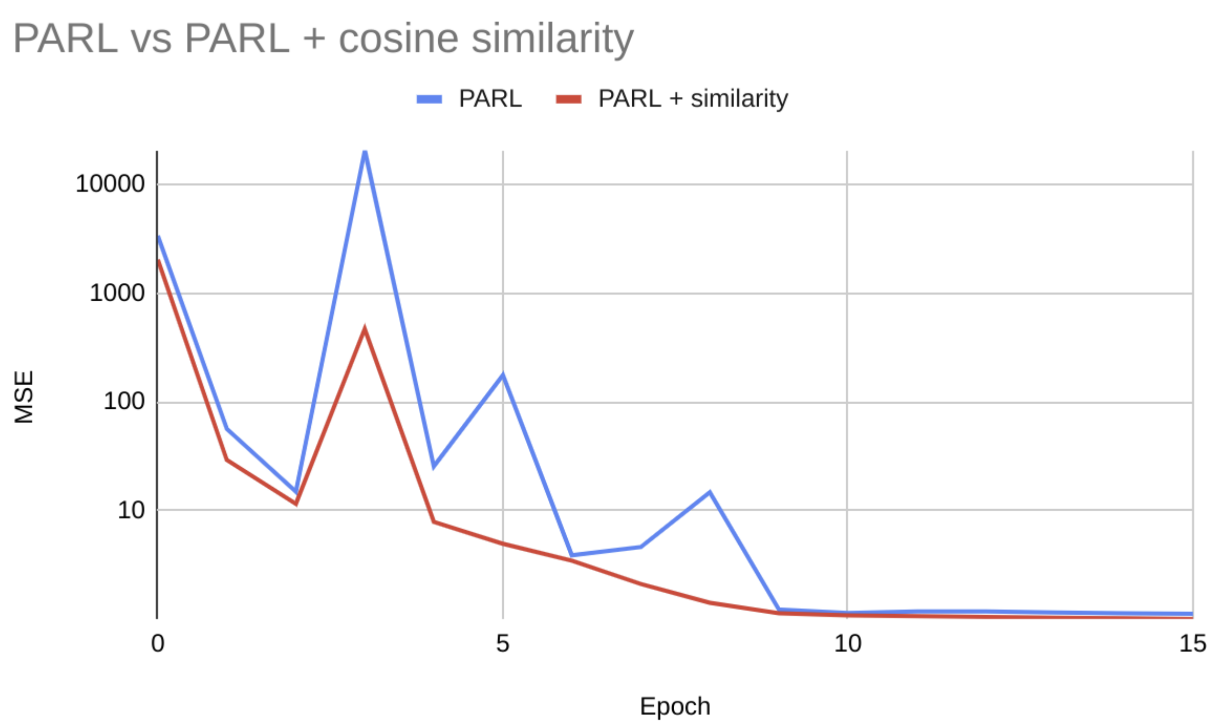
- Solution

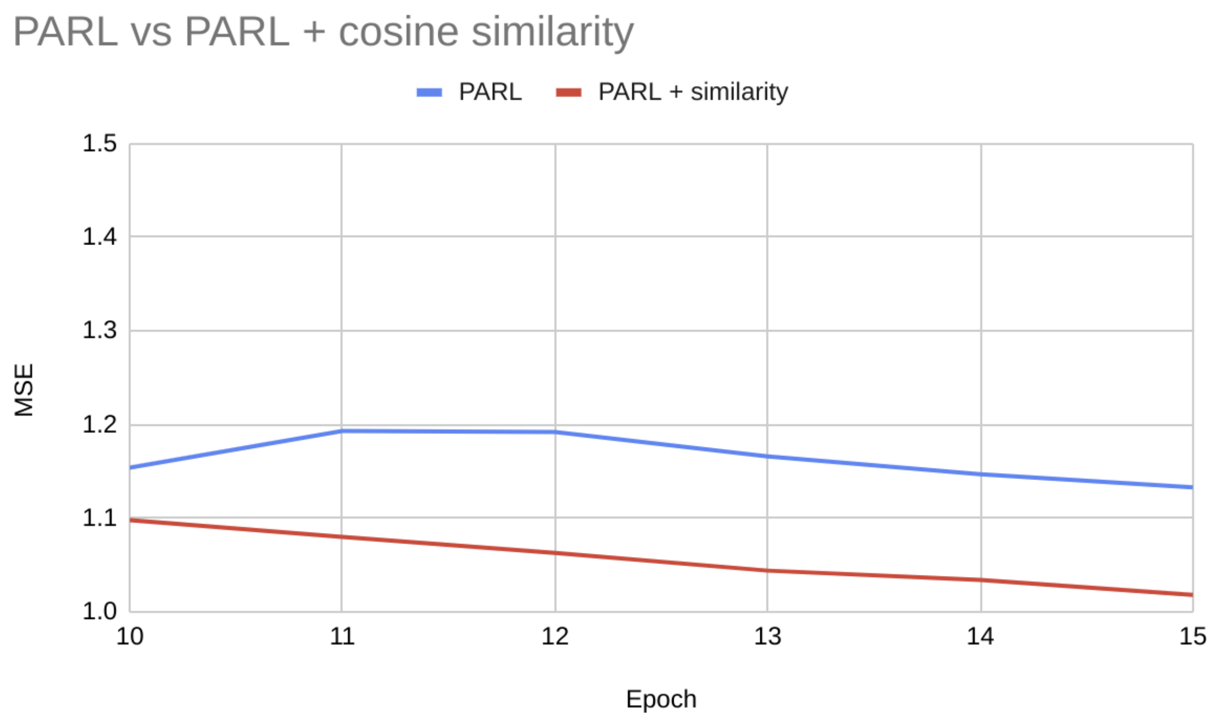
As we saw in the original PARL algorithm, the user auxiliary review document generation used a random selection for the auxiliary review. We are augmenting their algorithm in the following way: When we select the auxiliary review, we will look through the set of all users who gave the same rating to item *i*, the select the review of the user who is most similar to the original user based on cosine-similarity. We follow the same method of the original algorithm if no users gave the same rating to *i*: We will consider users who rated *i* 1 higher and then one lower than the original users rating.

- Experiments

To test our algorithm we have selected the Digital Music dataset from Amazon 5-core and used the original hyperparameters in the PARL repo because optimizing these values would be quite time consuming. We generated the 7 data files using our best approximation of the original PARL algorithm and then changed only the user auxiliary review document with our augmented pick_auxiliary_review algorithm. This is to ensure that we are making a 1 to 1 comparison to shed light on our improvement while ensuring there are no other variables that are changed between testing.

We trained each model for just 16 epochs because of the slow training time and compared the MSE on the test set for each epoch:





With our implementation in red you can see that for each epoch we outperformed the original model to varying degrees. At the end of the 16 epochs we can see that we have a MSE improvement of 0.115 which we believe to be significant because it is greater than the improvement PARL made over other algorithms to which it was compared.

While we have not improved the algorithm beyond the values provided in the paper, we are training for a much shorter period of time with different hyperparameters and slightly different dataset pre-processing. The improvement we have made is sound in logic and we have seen it outperforms the PARL algorithm when trained on identical data (except the user auxiliary review document) with identical model parameters. We believe this improvement to be significant.

Final Comments

The PARL model as proposed in the paper performs very well as seen in the above table, in fact it out performs 10 different highly capable models on all 5 datasets presented. Additionally it can be integrated with many other deep learning models which makes it very useful to the field of recommendation systems.

However, we feel that the datasets used in their research were not realistic to real world applications. Each rating in the dataset is accompanied by a review which makes for incredibly dense data utilized by a model claiming to fix the data sparsity problem. To use this dense data in their research feels artificial and it remains to be seen how it might perform on real world data. Finally we find it suboptimal to determine the auxiliary reviews based on a single same rating by another user. For this reason we have implemented a cosine similarity algorithm to better determine the auxiliary reviews and we find that our model performs better than the standard PARL model.