CONTEXT-AWARE RECOMMENDER SYSTEMS USING DEEP LEARNING

BY SHRUTHI MOHAN

21011101125

AI&DS-B

INTRODUCTION

- A context-aware recommender system can make recommendations to users by considering contextual information such as time and place, not only the scores assigned to items by users.
- Based on existing deep learning models, we combine a neural network and autoencoder to extract characteristics and predict scores in the process of restoring input data.
- The autoencoder here acts a preprocessor for the neural network. By encoding the input data into a more compact form, they can simplify the training process and improve the convergence of deep neural networks.

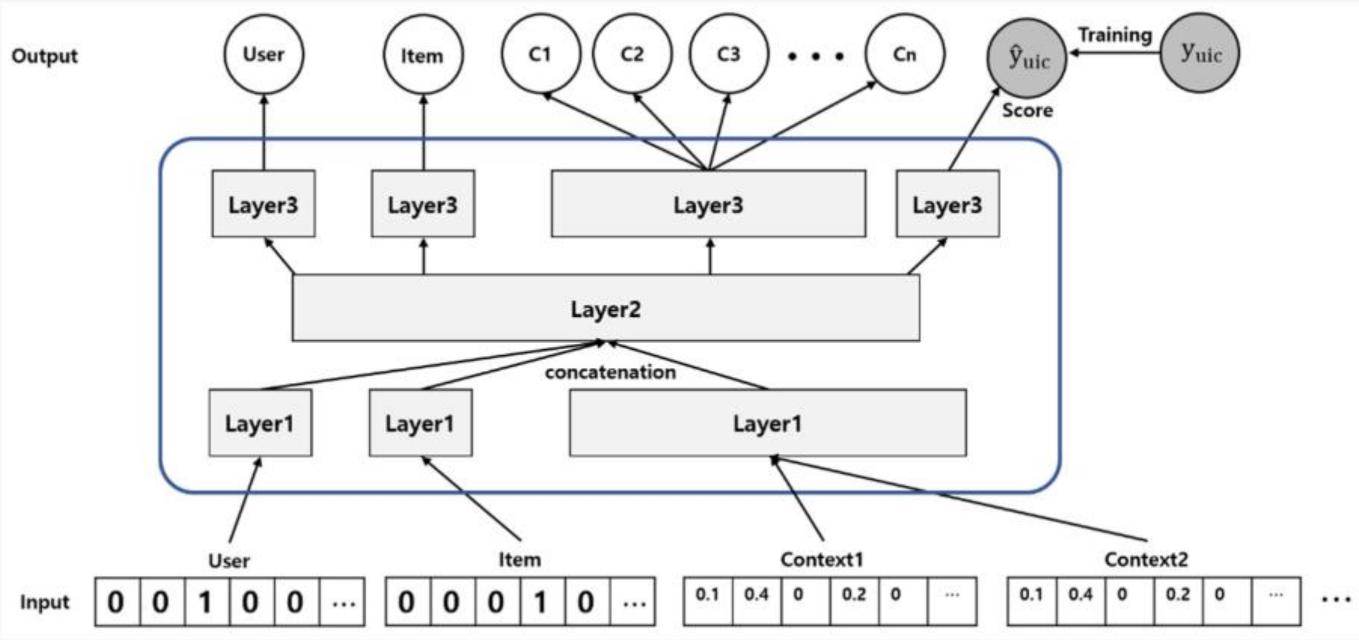
DATASETS:

- Restaurant dataset : https://github.com/irecsys/CARSKit/blob/master/context-aware_data_sets/Restaurant_TijuanaRestaurant.zip
- Movies dataset : https://github.com/irecsys/CARSKit/blob/master/context-aware_data_sets/Movie_DePaulMovie.zip
- Music dataset : https://github.com/irecsys/CARSKit/blob/master/context-aware_data_sets/Music_InCarMusic.zip
- Instacart Market Analysis: https://www.kaggle.com/c/instacart-market-basket-analysis

BASE MODEL

- The proposed model is formed by connecting multiple AEs and NN.
- The input and output data for the autoencoder include user vectors, item vectors and context vectors.
- The neural network is trained by using score as targets from the hidden layer in the middle.
- R2 Regularization
- Adaptive Moment Estimation (Adam) for optimization

Figure 2. Neural Context-aware Recommender Systems.



Algorithm 1 Algorithm of Proposed Model

Input: User vector *U*, Item vector *I*, Context vector C_n, Learning rate *N*, number of iterations *k*

Output: parameter w_{AE}, w_{NN}, b_{AE}, b_{NN}, Rating vector *Y*

Randomly initialize parameters w_{AE}, w_{NN}, b_{AE}, b_{NN}

for $i = 1$ to *k* **do**

for $u \in U$ **do**

Update w_{AE}, b_{AE} using L_{AE}'

Update w_{NN}, b_{NN} using L_{NN}'

end for

end for

BASE MODEL- RESULTS

- For the restaurant dataset, the NN yields similar results to those obtained from the proposed method because it has less contextual dimensions.
- For the movie and music datasets, which have relatively **higher contextual dimensions**, the proposed method yields much better performance than the user KNN.
- A supplementary model for the problem of data sparsity. Effective for nonlinear data by using deep learning

	Movie	Music	Restaurant
#of users	123	42	50
#of items	79	139	41
#of ratings	5043	4013	896
Dimension	3	8	1
Rating Scale	1–5	0–5	1–5
Data sparsity	94.5	99.9	93.4

The proposed system- autoencoders and neural network works better than a normal autoencoder system because:

- Neural networks help to learn the non-linear relationships and model better.
- Neural networks can handle multi-modal data integration by processing each modality separately and then combining the learned representations.
- Transfer Learning can be used as feature extractors and combined with autoencoders to leverage existing knowledge and improve the quality of content-based recommendations.

Table 2. Performance comparison result in Precision.

Precision@10	Music	Restaurant	Movie
UserKNN	0.010082	0.062889	0.063732
SVD++	0.045452	0.062174	0.059719
CAMF_CI	0.024234	0.050423	0.058384
ItemSplitting-BiasedMF	0.045365	0.06249	0.054859
CSLIM_CI	0.003996	0.012763	0.004598
PMF	0.048151	0.068362	0.06807
FM	0.010663	0.047136	0.027956
NN based model	0.050387	0.059913	0.076811
Proposed model	**0.056760**	**0.072748**	**0.102263**

POSSIBLE REFINEMENTS

Cons of the Base Model:

- Needs high contextual dimensions for good accuracy
- Needs to reflect reliability- unaware of what context affects the user's preference
- It struggles to capture short-term preference changes

1) FOR HIGH CONTEXTUAL DIMENSIONS

- Transfer learning can be incorporated using pre-trained datasets.
- Reinforcement Learning: reward functions that explicitly consider contextual information can be created. This encourages the model to learn from contextual cues and make better recommendations.

2) FOR RELIABILITY

- Conversational Systems can be incorporated to help users understand the rationale behind the recommendations and make them more trustworthy.
- We can use already built systems like Google LaMDA (2023) but the UI needs to be user-friendly and shouldn't make them feel like they are attending a survey.

3) FOR LONG-TERM PREFERENCE CHANGES

Transfer learning can be incorporated using pre-trained datasets.

- Reinforcement Learning: reward functions that explicitly consider contextual information can be created. This encourages the model to learn from contextual cues and make better recommendations.

REFERENCES

- Deep Learning-Based Context-Aware Recommender System Considering Contextual Features(2021): https://www.mdpi.com/2076-3417/12/1/45
- Deep Learning-Based Context-Aware Recommender System Considering Change in Preference(2023): https://www.mdpi.com/2079-9292/12/10/2337
- Leveraging Large Language Models in Conversational Recommender Systems: https://arxiv.org/abs/2305.07961