PARL: Let Strangers Speak Out What You Like

Libing Wu , Cong Quan , Chenliang Li, Donghong Ji

Presenters: August Soderberg and Matt Sehgal

Overview:

Problem Definition

Challenge Analysis

Philosophy

Solution

Experiments

Our Improvements:

- **Philosophy**
- **Solution**

Problem Definition

Problem Definition

Relevant recommendations

Data sparsity problem

Review, rating, interaction-based

Challenge Analysis

Challenge Analysis

Review-based

No reviews

Short reviews

Philosophy

Philosophy

Neighborhood-based method (CF)

Ratings by same user on other items

Apply to reviews

Philosophy

Neural network

Extract informative features

Users' auxiliary review document

Philosophy

Features - user-item pairs

Combined with different neural models

Rating prediction

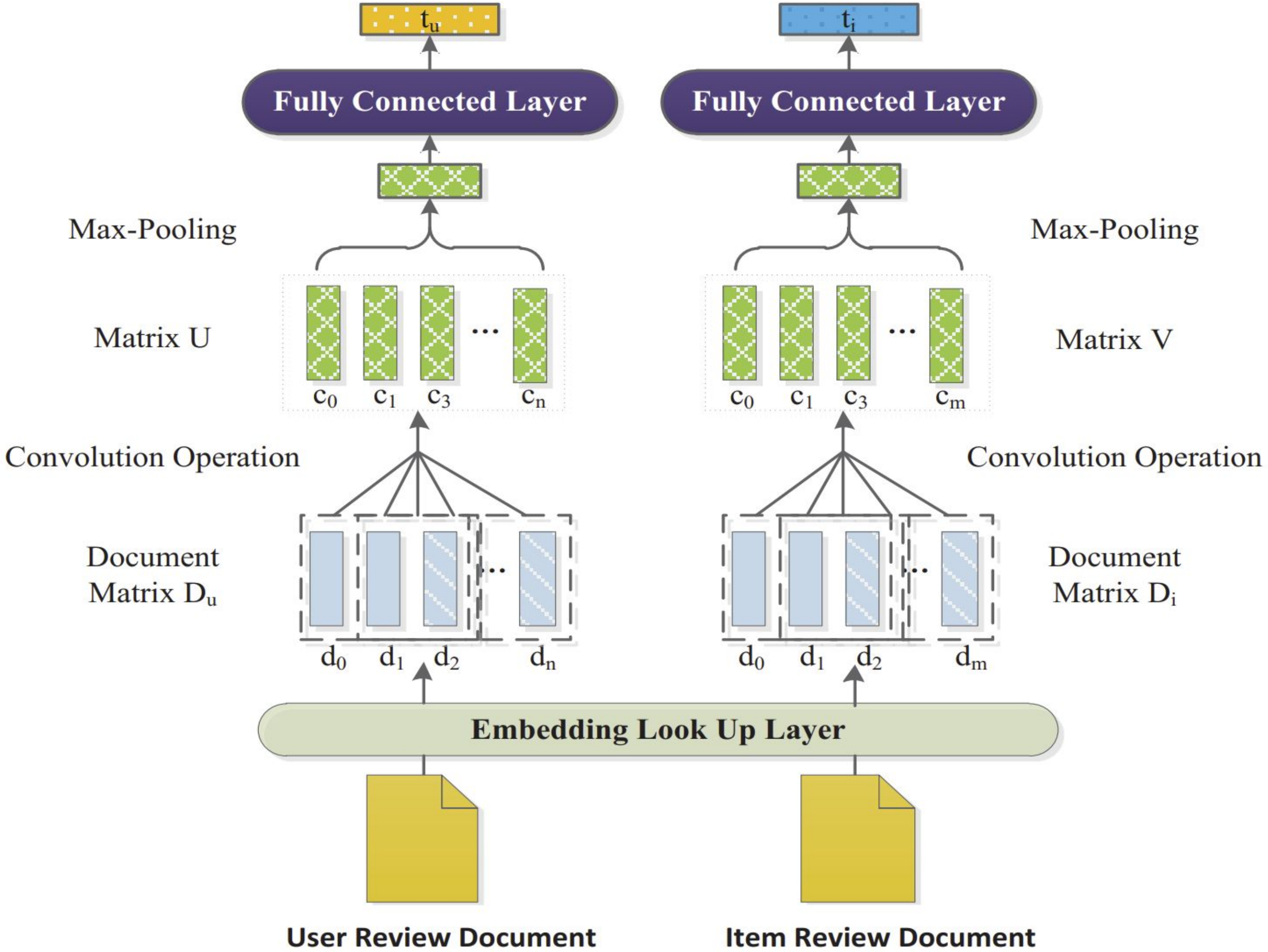
Solution

Solution

Extract informative features

Review-based neighborhood model

DeepCoNN



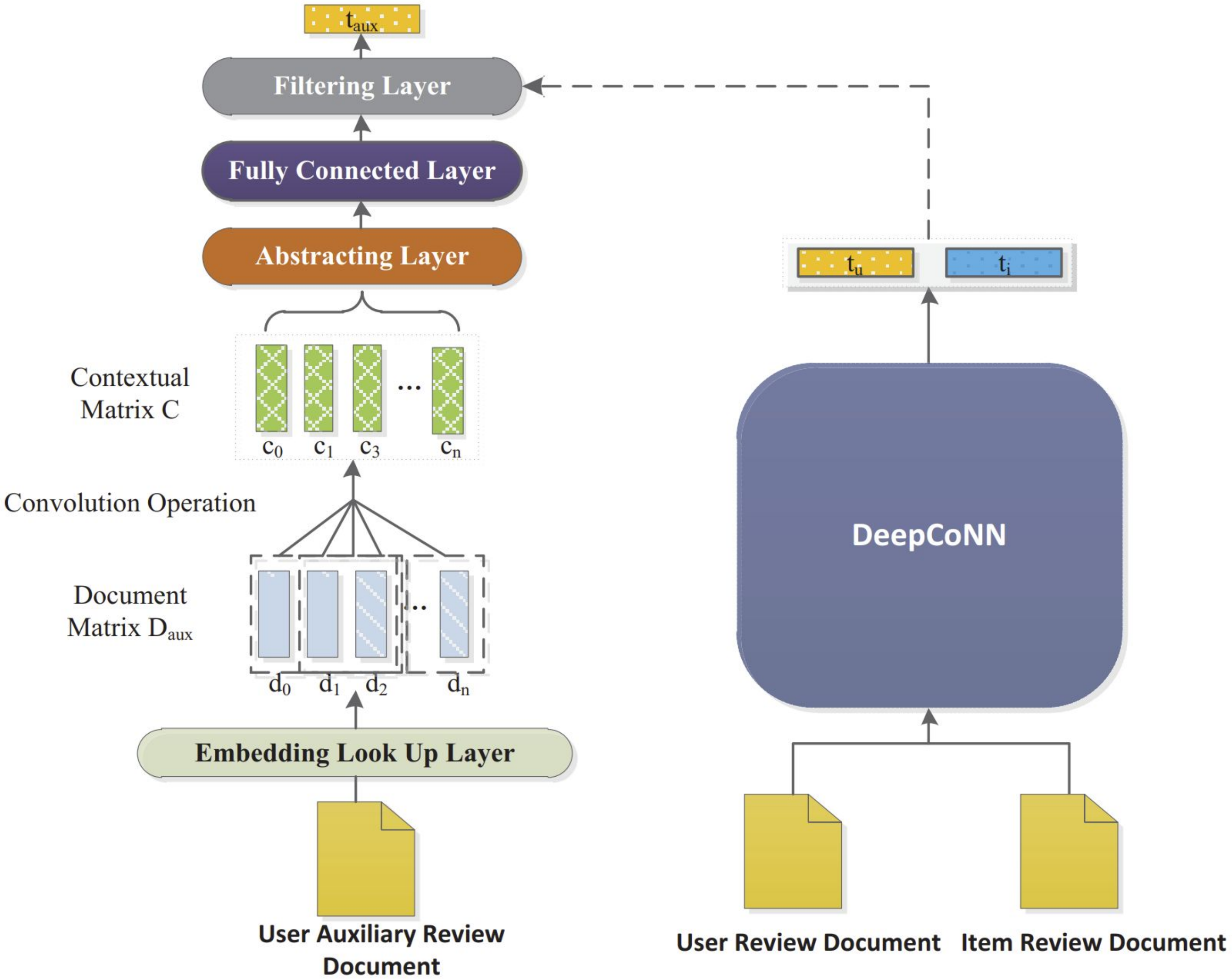
t_u
t_i
Fully Connected Layer
Fully Connected Layer
Max-Pooling
Max-Pooling
Matrix U
Matrix V
c_0 c_1 c_3 … c_n
c_0 c_1 c_3 … c_m
Convolution Operation
Convolution Operation
Document Matrix D_u
Document Matrix D_i
d_0 d_1 d_2 … d_n
d_0 d_1 d_2 … d_m
Embedding Look Up Layer
User Review Document
Item Review Document

Solution

Auxiliary Review Document

Algorithm 2: Pick_Auxiliary_Review

Input: item i, rating r
Output: picked $review$
$set = get\_users(i, r)$ # get users who give rate r to i
$u = random(set)$ # randomly select one like-minded user
return $review_{u,i}$


t_{aux}
Filtering Layer
Fully Connected Layer
Abstracting Layer
Contextual Matrix C
c_0 c_1 c_3 ... c_n
Convolution Operation
Document Matrix D_{aux}
d_0 d_1 d_2 ... d_n
Embedding Look Up Layer
User Auxiliary Review Document
t_u t_i
DeepCoNN
User Review Document
Item Review Document

Solution

Latent vectors t_u and t_aux

Item latent vector t_i

Factorization Machine

Experiments - MSE

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**

Our Philosophy

Our Philosophy

“Like-minded” users are not truly “like-minded”

User sentiments are better reflected by truly similar users

Supplying reviews from similar users will improve quality of auxiliary reviews

Our Solution

Our Solution

PARL design:

- plug-and-play architecture
- plugged into DeepCoNN in paper

Challenges:

- data preprocessing methods are vague
- no processed data provided
- only PARL architecture provided

Our solution experiment:

- Create dataset using methods in paper
- Evaluate our PARL against PARL with DeepCoNN

Our Solution

Dataset:

Amazon Digital Music Dataset (5-Core)

All users and items have 5 reviews each

Pre-processing methods from paper

Our Solution

Modified PARL:

Rank users by similarity

Get auxiliary review from most similar user

Algorithm 2: Pick_Auxiliary_Review

Input: item i, rating r
Output: picked $review$
$set = get\_users(i, r)$ # get users who give rate r to i
$u = random(set)$ # randomly select one like-minded user
return $review_{u,i}$

 # get all users ranked by similarity to user ***u***
set = get_similar_users(u)
 # select most similar user who gives rate ***r*** *to* ***i***
u = get_aux_user(set, i, r)

Algorithm 2: Pick_Auxiliary_Review **(modified)**

Input: item i, rating r
Output: picked $review$
set = get_similar_users(u) *# get all users ranked by similarity to user* ***u***
u = get_aux_user(set, i, r) *# select most similar user who gives rate* ***r*** *to* ***i***
return $review_{u,i}$

Our Solution

get_similar_users(u):

Cosine similarity

get_aux_user(set, i, r):

Most similar user with rating **r** for **i**

Our Comments

Pros:

Performs well

Integrates with other models

Our Comments

Cons:

Datasets contain reviews for each rating

Only considers users with single same rating