PCIC 2021: Causal Inference and Recommendation

Jinwei Luo[1], Zinan Lin[1], and Weike Pan[1,2]

luojinwei2016@email.szu.edu.cn, lzn87591@gmail.com, panweike@szu.edu.cn

[1]College of Computer Science and Software Engineering
Shenzhen University, Shenzhen, China

[2]Team Advisor

Statistics (1/3)

Table: Statistics of the datasets used in the competition. Note that P/N denotes the ratio between the numbers of positive feedback and negative feedback, and unlike the Big-Tag set, tags that do not appear in Choice-Tag set can be considered negative.

Dataset	|User|	|Item/Tag|	|Record|	P/N
Big-Tag	1000	1719	14133	37.96 %
Choice-Tag	999	1720	5802	10.64 %
Rating	1000	1000	19903	-

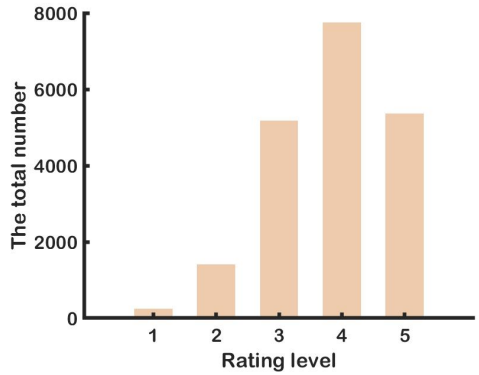


Figure: The distribution of rating records in Rating set.

Statistics (2/3)

We find that there is some noise between rating set and both tag sets. For example, the user's rating is high but his feedback on all tags are negative, or the user's rating is low but his feedback on all tags are positive.

- The number of noise feedback between rating set and big-tag set is 24.
- The number of noise feedback between rating set and choice-tag set is 4.

Statistics (3/3)

We find that there is an intersection between different subsets, i.e., the same (user, item) pair exists. In particular, the intersection with the validation set needs to be removed first during evaluation to avoid the impact on parameter search.

Table: Statistics of the size of the intersection between different subsets.

Object	Size
Big-Tag and Choice-Tag	514
Big-Tag and the validation data	223
Choice-Tag and validation data	90

Causality Diagrams

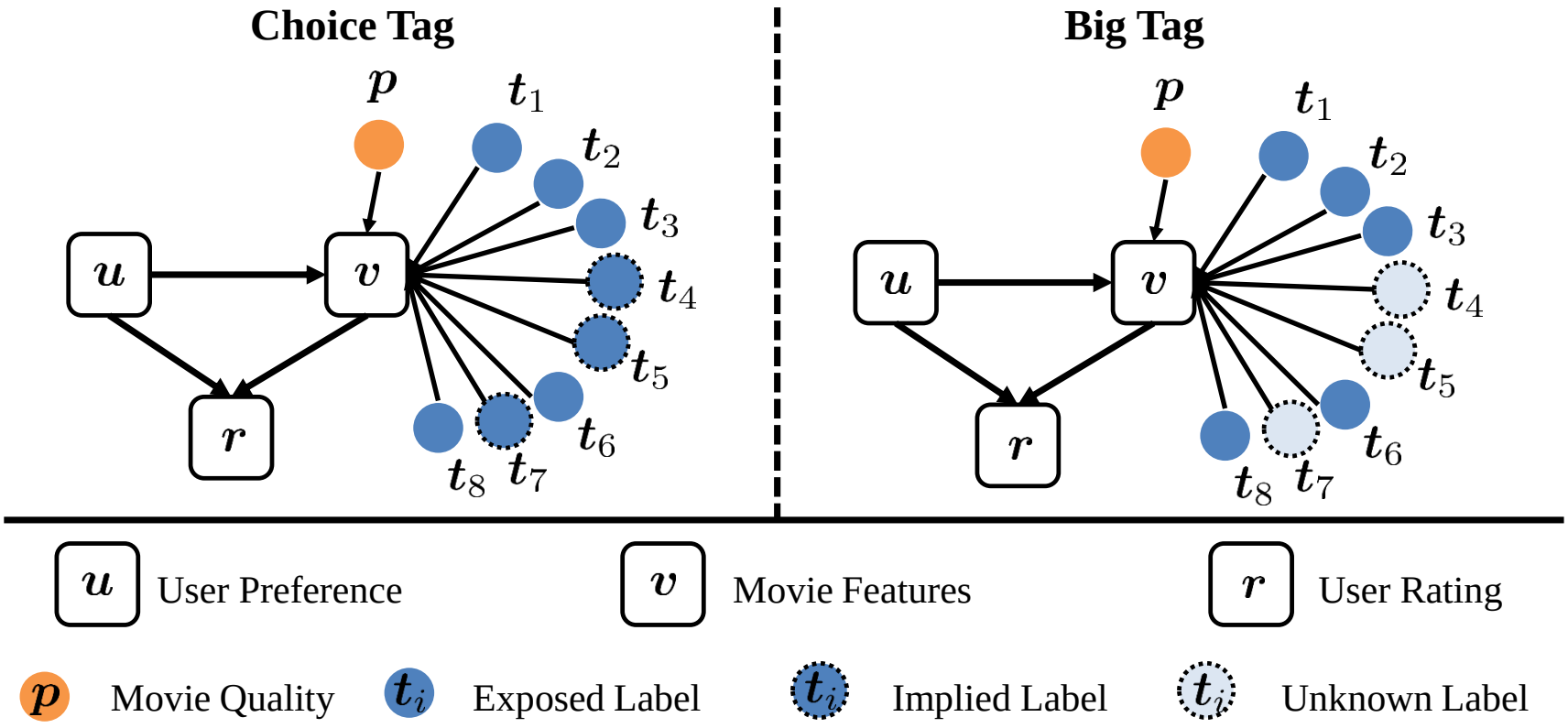
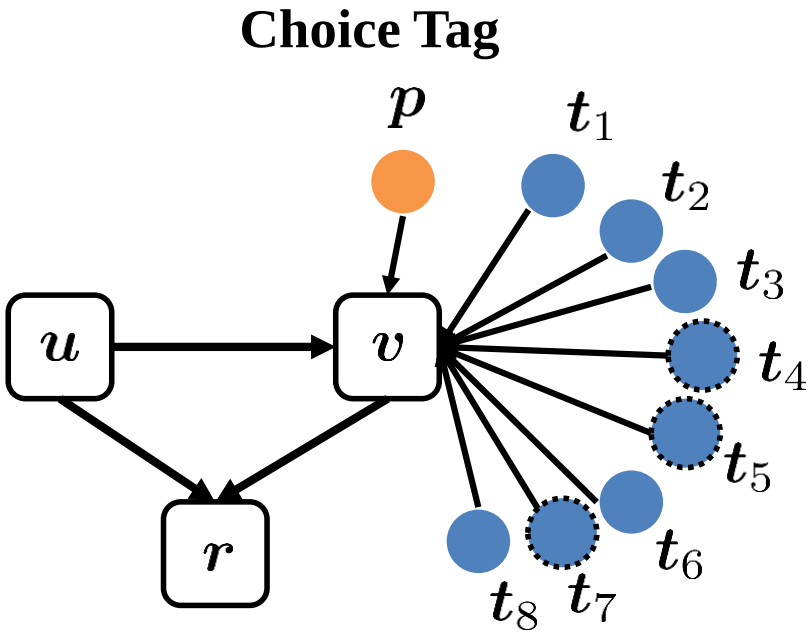


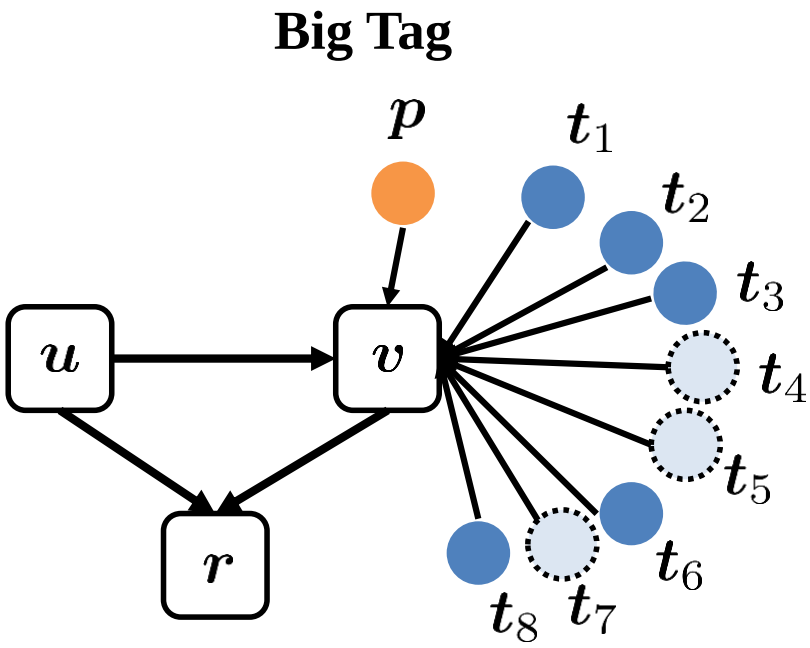
Figure: Causality diagrams for ratings on different subsets.

Intuitive Ideas (1/2)



- The rating is represented by the inner product sum of the movie features and the user preference. The key question is how to obtain movie features:
  - The movie features can be decomposed into the quality of the movie itself and the attributes (tags) of the movie.
  - From the user's perspective, the influence of different attributes of the movie on the final rating can be positive or negative.
- Easy to solve.

Intuitive Ideas (2/2)



- Different from the setting of choice-tag, the unknown(missing) label of tags can be negative or unobserved positive.
- Difficult to solve.

Notations(1/2)

n	user number
m	movie number
z	tag number
$u \in \{1, 2, \ldots, n\}$	user ID
$i \in \{1, 2, \ldots, m\}$	movie ID
$t \in \{1, 2, \ldots, z\}$	tag ID
r_{ui}	observed rating of user u on movie i
y_{ui}	one class feedback of user u on tag t
$\mathbb{G}$, e.g., $\mathbb{G} = \{1, 2, 3, 4, 5\}$	grade score set (or rating range)
$\mathbf{R} \in \{\mathbb{G} \cup ?\}^{n \times m}$	rating matrix
$\mathcal{R} = \{(u, i, r_{ui})\}$	observed rating records (training data)
$\mathcal{Y} = \{(u, t, y_{ut})\}$	one-class feedback of tags (training data)

Notations(2/2)

$b_u \in \mathbb{R}$	user bias
$b_i \in \mathbb{R}$	movie bias
$b_t \in \mathbb{R}$	tag bias
$U_{u\cdot} \in \mathbb{R}^{1\times d}$	user-specific latent feature vector
$V_{i\cdot} \in \mathbb{R}^{1\times d}$	movie-specific latent feature vector
$T_{t\cdot} \in \mathbb{R}^{1\times d}$	tag-specific latent feature vector
$d \in \mathbb{R}$	number of latent dimensions
α, β	tradeoff parameters of different tasks
$\hat{r}_{\text{ui}}$	predicted rating of user u on item i
$\hat{y}_{\text{ut}}$	predicted preference of user u on tag t

Weighted and Pairwise Joint Matrix Factorization (1/6)

$$\mathcal{L}_{WPJ-MF} = \mathcal{L}_{Rating} + \alpha\mathcal{L}_{Tag} + \beta\mathcal{L}_{Pair} + \lambda\|\theta\|, \quad (1)$$

- We propose a method named Weighted and Pairwise Joint Matrix Factorization (WPJ-MF) to fully leverage the knowledge in the two forms of data, in which a multi-task learning strategy is designed to focus more on the modeling of causality of data.
- The first term is the dominant part of the loss function, which is the task of rating prediction.
- The second term and the third term force the model to predict the preference of user on tags from both pointwise and pairwise perspective.
- The last term is the regularization term.

Weighted and Pairwise Joint Matrix Factorization (2/6)

$$\mathcal{L}_{Rating} = \min_{\Theta} \sum_{r_{ui} \in \mathcal{R}} (r_{ui} - \hat{r}_{ui})^2 , \tag{2}$$

$$\hat{r}_{ui} = U_u \cdot V_i^T + b_u + b_i + \frac{\sum_{t \in \mathcal{N}(i)} b_t}{|\mathcal{N}(i)|}, \tag{3}$$

$$V_i = \frac{\sum_{t \in \mathcal{N}(i)} T_t}{|\mathcal{N}(i)|}, \tag{4}$$

where $\mathcal{N}(i)$ is the tag set of movie i.

- It can be seen from Eq.(3) that we use the mean vector of tag-specific latent feature vectors of the tags belong to movie i as the movie-specific latent feature vector V_i.
- Notice that the movie-specific bias b_i is important to model the casual effect of movie quality on rating.

Weighted and Pairwise Joint Matrix Factorization (3/6)

$$\mathcal{L}_{Rating} = \min_{\Theta} \sum_{r_{ui} \in \mathcal{R}} (r_{ui} - \hat{r}_{ui})^2 , \tag{5}$$

$$\hat{r}_{ui} = U_u \cdot V_i^T + b_u + b_i + \frac{\sum_{t \in \mathcal{N}(i)} b_t}{|\mathcal{N}(i)|}, \tag{6}$$

$$V_i = \frac{\sum_{t \in \mathcal{N}(i)} T_t}{|\mathcal{N}(i)|}, \tag{7}$$

where $\mathcal{N}(i)$ is the tag set of movie i.

- In rating task, we use all the tags of the movie to simply represent the movie features, which can be regarded as not considering the effect of the path from $u \rightarrow v \rightarrow r$.

Weighted and Pairwise Joint Matrix Factorization (4/6)

$$\mathcal{L}_{Tag} = \min_{\Theta} \sum_{y_{ut} \in \mathcal{Y}} (y_{ut} - \hat{y}_{ut})^2 , \tag{8}$$

$$\hat{y}_{ut} = U_u \cdot T_t^T + b_u + b_t, \tag{9}$$

where $\sigma(z) = 1/(1 + e^{-z})$ is a sigmoid function.

- In tag task, we force the model directly fit the user's preferences for different tags, which indicates the effect of the path from $u \to v \to r$.

Weighted and Pairwise Joint Matrix Factorization (5/6)

$$\mathcal{L}_{Pair} = \min_{\Theta} \sum_{y_{ut} \in \mathcal{Y}} -\ln \sigma \left(\hat{y}_{ut} - \hat{y}_{ut'} \right), \tag{10}$$

$$\hat{y}_{ut} = U_u \cdot T_t^T + b_u + b_t, \tag{11}$$

where $\sigma(z) = 1/(1+e^{-z})$ is a sigmoid function and t' is a sampled unobserved tag to apply the pairwise training.

- In pairwise task, considering the missing mechanism of tags on big-tag, we use the pairwise training method to further model the path effect of $u \to v \to r$ on the big-tag.

Weighted and Pairwise Joint Matrix Factorization (6/6)

$$\mathcal{L}_{WPJ-MF} = \mathcal{L}_{Rating} + \alpha\mathcal{L}_{Tag} + \beta\mathcal{L}_{Direct} + \lambda\|\theta\|, \tag{12}$$

- By modeling the causal effects on ratings on different subsets, we believe that WPJ-MF can reasonably capture user preferences on tags.
- Therefore, we consider an improved version of WPJ-MF, that is, WPJ-MF is used as a imputation model to calculate predicted labels $\hat{y}_{ut}$ for some sampled (u, t), and then use the idea of direct methods to introduce predicted labels for retraining.
- The calculation of $\mathcal{L}_{Direct}$ is similar to that of $\mathcal{L}_{Tag}$, and we remove the term $\mathcal{L}_{Pair}$ because the unknown label may have been replaced by the predicted label at this time.

Implementation Details (1/2)

- We run our experiments on a GPU computer cluster with Linux system. The cluster contains a total of four 14-core CPUs, 256GB RAM, and eight Nvidia Tesla P100 GPUs.
- We implement our methods on PyTorch 1.0.1 with the Adam optimizer.
- In order to find the optimal hyper-parameters, we use a hyper-parameter search library Optuna instead of grid search by checking the AUC performance on the validation data in the experiment.

Implementation Details (2/2)

- The ranges of the values for the hyper-parameters to be tuned is as follows. Notice that we adopt an early stopping strategy with the patience set to 5 times for the methods.

Name	Range	Functionality
rank	$\{4, 8, ..., 60, 64\}$	Embedded dimension
λ	$\{1e^{-5}, 1e^{-4} \cdots 1e^{-1}, 1\}$	Regularization
bs	$\{128, 256, 512, 1024, 2048\}$	Batch size
lr	$\{0.001, 0.005, 0.01, 0.05, 0.1\}$	Learning rate
iter	$\{100\}$	Iteration number
α	$\{0.01, ..., 0.99\}$	Weighting for $\mathcal{L}_{Tag}$
β	$\{0.01, ..., 0.99\}$	Weighting for $\mathcal{L}_{Pair}$

Results

- WPJ-MF w/o $\mathcal{L}_{Pair}$: the reduced version of WPJ-MF without term $\mathcal{L}_{Pair}$.
- WPJ-MF + direct method: the method in which we use WPJ-MF to impute soft labels of the phase 1 test set, and use WPJ-MF w/o $\mathcal{L}_{Pair}$ to retrain.
- Note that we do not use any ensemble method.

Method	Phase 1	Phase 2
WPJ-MF	0.8040	0.7862
WPJ-MF w/o $\mathcal{L}_{Pair}$	0.7972	0.7898
WPJ-MF + direct method	-	0.7908

In Future

Considering the small scale of the data set, in the experiment we only used the matrix factorization framework. A more complex and advanced framework may bring better gains. We can explore the combination of the following methods:

- Autodebias [Chen et al., 2021]
- AT-MF [Saito, 2020]
- Rel-MF [Saito et al., 2020]
- KDC-Rec [Liu et al., 2020]
- Feature Calibration [Islam et al., 2021]

In addition, we only considered the missing mechanism of tags in the competition due to the time relationship. The modeling of the missing mechanism of rating also needs to be further considered in future work.

Thank You!

- We thank the sponsors of the competition, i.e., Huawei Noah's Ark Laboratory and Peking University.
- We also thank Shenzhen University and our laboratory for their resource support.

Chen, J., Dong, H., Qiu, Y., He, X., Xin, X., Chen, L., Lin, G., and Yang, K. (2021).

Autodebias: Learning to debias for recommendation.

arXiv preprint arXiv:2105.04170.

Islam, R., Keya, K. N., Zeng, Z., Pan, S., and Foulds, J. (2021).

Debiasing career recommendations with neural fair collaborative filtering.

In *Proceedings of the Web Conference 2021*, pages 3779–3790.

Liu, D., Cheng, P., Dong, Z., He, X., Pan, W., and Ming, Z. (2020).

A general knowledge distillation framework for counterfactual recommendation via uniform data.

In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 831–840.

Saito, Y. (2020).

Asymmetric tri-training for debiasing missing-not-at-random explicit feedback.

In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 309–318.

Saito, Y., Yaginuma, S., Nishino, Y., Sakata, H., and Nakata, K. (2020).

Unbiased recommender learning from missing-not-at-random implicit feedback.

In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 501–509.