Learning Disentangled Representations for Recommendation

Jianxin Ma[1,2*], Chang Zhou[1*] (co-first), Peng Cui[2], Hongxia Yang[1], Wenwu Zhu[2]
[1]Alibaba Group [2]Tsinghua University

NeurIPS | 2019

0 TL;DR

We learn *disentangled representations* purely from *user behavior data* such as clicks in a recommender system.

- We disentangle both (1) items' representations, which are *parameters*, and (2) users' representations, which are the outputs of an encoder, via *prototype-based clustering* and β-*VAE*.

1 (Background) Disentangled Representation Learning

It aims to learn **factorized** representations that uncover and disentangle the latent causal factors hidden in the observed data (Bengio et al., 2013).

Existing work mostly focuses on image data, while we focus on the user behavior data collected in a recommender system.

2 (Method) Macro-Micro Disentangled VAE

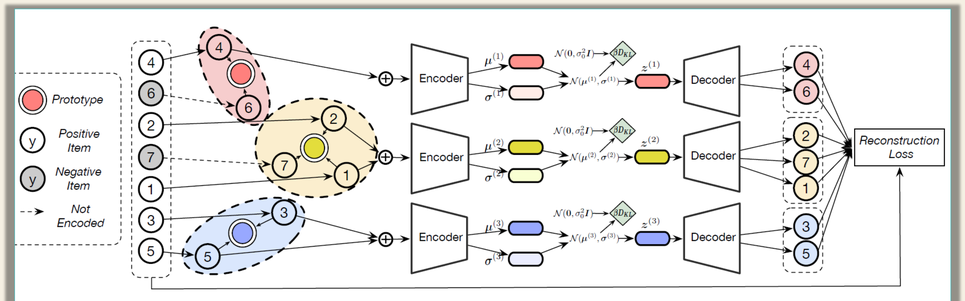


Figure 1: Our framework. Macro disentanglement is achieved by learning a set of prototypes, based on which the user intention related with each item is inferred, and then capturing the preference of a user about the different intentions separately. Micro disentanglement is achieved by magnifying the KL divergence, from which a term that penalizes total correlation can be separated, with a factor of β.

- **Macro disentanglement:** Separate a user's **different intentions**, e.g., to buy a shirt or a phone.
- **Micro disentanglement:** Separate a user's preference regarding **different aspects** when executing a specific intention, e.g., the preferred price, color, or size when buying a shirt.

$$p_{\boldsymbol{\theta}}(\mathbf{x}_u) = \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{C})}\left[\int p_{\boldsymbol{\theta}}\left(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C}\right) p_{\boldsymbol{\theta}}(\mathbf{z}_u)\, d\mathbf{z}_u\right],$$

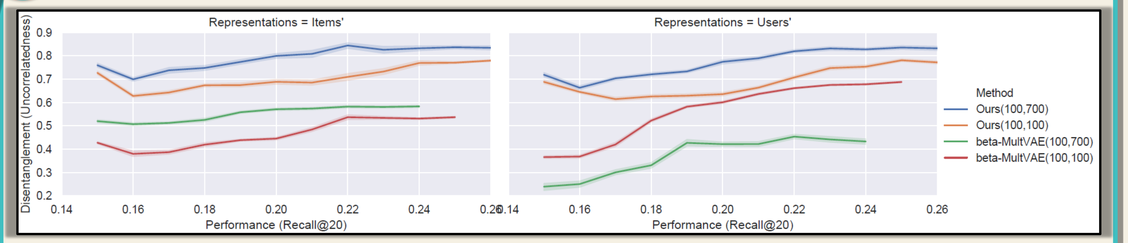
$$p_{\boldsymbol{\theta}}\left(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C}\right) = \prod_{x_{u,i} \in \mathbf{x}_u} p_{\boldsymbol{\theta}}(x_{u,i} \mid \mathbf{z}_u, \mathbf{C}).$$

(1) $\boldsymbol{x}_u$ is a list of items clicked by user u. (2) $\boldsymbol{z}_u$ is the user's representation, which is first factorized into K components at the macro level, and further factorized into d dimensions at the micro level. (3) $\boldsymbol{C} = \{\boldsymbol{c}_i\}_i$ are one-hot indicators that tells which intention (out of the K options) is typically related with each item i. $p(\boldsymbol{C})$ is parameterized as a prototype-based network.

$$\mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{C})}\left[\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}_u|\mathbf{x}_u,\mathbf{C})}[\ln p_{\boldsymbol{\theta}}(\mathbf{x}_u \mid \mathbf{z}_u, \mathbf{C})] - \beta \cdot D_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\mathbf{z}_u \mid \mathbf{x}_u, \mathbf{C}) \| p_{\boldsymbol{\theta}}(\mathbf{z}_u))\right]$$

(See the paper for details.)

3 (Analysis) Disentanglement vs. Performance



Our approach outperforms the baselines in terms of both disentanglement and performance. Moreover, it appears that:

1. **Disentanglement is generally associated with better performance.**
2. **Macro disentanglement benefits micro disentanglement**, since the blue curve (i.e., when $K = 7$) is higher than the orange one (i.e., when $K = 1$).

4 (Results) Top-N Recommendation

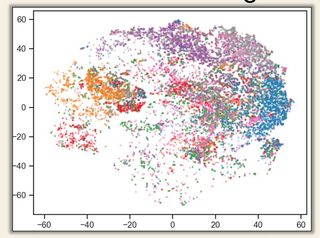
Table 1: Collaborative filtering. All methods are constrained to have around $2Md$ parameters, where M is the number of items and d is the dimension of each item representation. We set $d = 100$.

Dataset	Method	Metrics: NDCG@100	Recall@20	Recall@50
AliShop-7C	MultDAE	0.23923 (±0.00380)	0.15242 (±0.00305)	0.24892 (±0.00391)
	β-MultVAE	0.23875 (±0.00379)	0.15040 (±0.00302)	0.24589 (±0.00387)
	Ours	**0.29148** (±0.00380)	**0.18616** (±0.00317)	**0.30256** (±0.00397)
ML-100k	MultDAE	0.24487 (±0.02738)	0.23794 (±0.03605)	0.32279 (±0.04070)
	β-MultVAE	0.27484 (±0.02883)	0.24838 (±0.03294)	0.35270 (±0.03927)
	Ours	**0.28895** (±0.02739)	**0.30951** (±0.03808)	**0.41309** (±0.04503)
ML-1M	MultDAE	0.40453 (±0.00799)	0.34382 (±0.00961)	0.46781 (±0.01032)
	β-MultVAE	0.40555 (±0.00809)	0.33960 (±0.00919)	0.45825 (±0.01039)
	Ours	**0.42740** (±0.00789)	**0.36046** (±0.00947)	**0.49039** (±0.01029)
ML-20M	MultDAE	0.41900 (±0.00209)	0.39169 (±0.00271)	**0.53054** (±0.00285)
	β-MultVAE	0.41113 (±0.00212)	0.38263 (±0.00273)	0.51975 (±0.00289)
	Ours	**0.42496** (±0.00212)	**0.39649** (±0.00271)	0.52901 (±0.00284)
Netflix	MultDAE	0.37450 (±0.00095)	0.33982 (±0.00123)	0.43247 (±0.00126)
	β-MultVAE	0.36291 (±0.00094)	0.32792 (±0.00122)	0.41960 (±0.00125)
	Ours	**0.37987** (±0.00096)	**0.34587** (±0.00124)	**0.43478** (±0.00125)

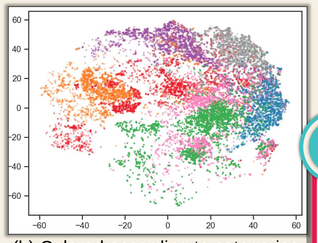
5 (Visualize.) Macro Disentanglement

To see how *interpretable* the macro factors are, we color the items according to their associated macro factors.

- The macro factors resemble the categories.

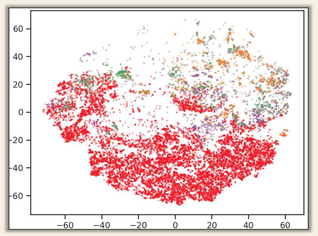


(a) Colored according to macro factors.



(b) Colored according to categories.

We observe the said interpretability with ***cosine***, but not with dot product.



(c) Mode collapse happens when we use dot product instead of cosine similarity.

6 (Visualize.) Micro Disentanglement & User-Controllable Recommendation

A user may now (1) **vary a single dimension** such as color, while (2) **keeping the other dimensions constant** when browsing the recommendation list.

Varying this single dimension results in different **bag sizes**.

Another dimension is about **bag colors**.

(These images are not generated, but retrieved according to the altered query vectors.)

Note: Not all dimensions are human-understandable. Well-trained models can only be identified with the help of a few labels. We encourage future efforts to explore (semi-)supervised methods (Locatello et al., 2019a; 2019b).