Learning and Transferring IDs Representation in E-commerce

Kui Zhao, Yuechuan Li, Zhaoqian Shuai, Cheng Yang

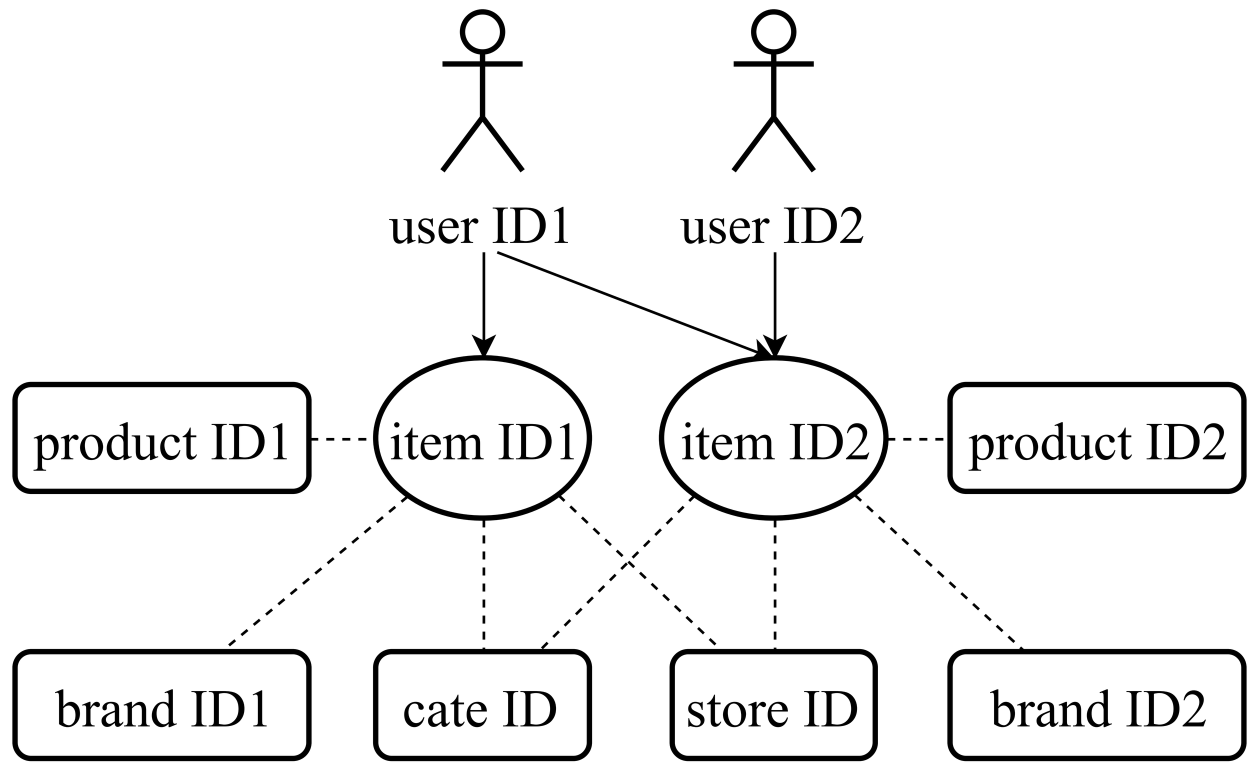
Alibaba Machine Intelligence Technology Lab

Contact Information:

{*zhaokui.zk, yuechuan.lyc*
wanqian.szq, charis.yangc}
@alibaba-inc.com

Summary

We proposed an embedding-based framework to learn and transfer representations of all IDs in E-commerce.



It was deployed and evaluated in several scenarios of Hema App:

- *Item relationships*: measuring items similarity.
- *New items*: transferring from seen items to unseen items.
- *New users*: transferring across different domains.
- *Sales forecast*: transferring across different tasks.

Learning IDs Representation

Jointly embedding attribute IDs

By exploring structural connections between the item ID and its attribute IDs, their representations can be learned jointly.

Firstly, the co-occurrence of item IDs also implicates the co-occurrence of corresponding attribute IDs:

$$
\begin{aligned}
&p\left(\text{IDs}(\text{item}_j)|\text{IDs}(\text{item}_i)\right)\\
=&\sigma\left(\sum_{k=1}^{K}(w_{jk}\mathbf{e}'_{jk})^{\mathrm{T}}(w_{ik}\mathbf{e}_{ik})\right)\prod_{s=1}^{S}\sigma\left(-\sum_{k=1}^{K}(w_{sk}\mathbf{e}'_{sk})^{\mathrm{T}}(w_{ik}\mathbf{e}_{ik})\right),
\end{aligned}
\tag{1}
$$

where w_{ik} is the weight of $\text{id}_k(\text{item}_i)$.

Secondly, structural connections mean constrains:

$$
p(\text{item}_i|\text{IDs}(\text{item}_i)) = \sigma\left(\sum_{k=2}^{K} w_{ik}\mathbf{e}_{i1}^{T}\mathbf{M}_k\mathbf{e}_{ik}\right). \tag{2}
$$

Then we maximize the following average log probability:

$$
\begin{aligned}
\mathcal{J} =& \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{-C\le j\le C}^{1\le n+j\le N, j\neq 0}\log p(\text{IDs}(\text{item}_{n+j})|\text{IDs}(\text{item}_n))\right.\\
&\left.+\alpha\log p(\text{item}_n|\text{IDs}(\text{item}_n)) - \beta\sum_{k=1}^{K}||\mathbf{M}_k||_2\right).
\end{aligned}
\tag{3}
$$

Embedding user IDs

Users' preferences are reflected by the sequences of interactive item IDs. Thus the representation of user IDs can be obtained by aggregating embedding vectors of interactive item IDs. As embedding vectors of user IDs should be updated frequently to reflect the latest preference, *Average* is chosen instead of RNN etc.

Deploying IDs Representation

Measuring items similarity

The item-item similarity can be measured by cosine similarity between vectors. These relationships are extensively used in many recommendation tasks, such as "People also like" and CTR prediction.

Transferring from seen items to unseen items

New items cause cold-start problems, and we construct approximate embedding vectors to relieve that. According to Eq.(2), maximizing Eq.(3) leads to $p(\text{item}_i|\text{IDs}(\text{item}_i)) \to 1 \Rightarrow \mathbf{e}_{i1} \approx \sum_{k=2}^{K} w_{ik}\mathbf{e}_{ik}^{T}\mathbf{M}_k^{T}$.

Transferring across different domains

For emerging platforms like Hema App, a high proportion of users are new customers. To overcome this cold-start problem, we transfer the preference of users on long existing platforms (e.g. Taobao), onto the emerging platform. As items in the source domain that are similar to items in the target domain should have higher weights, *Average* can be extended to *Weighted-average*.

Transferring across different tasks

Sales forecast can help to manage the workforce, such as guiding us to pre-order appropriate number of delivery staff. By taking embedding vectors of store IDs as a part of input, it helps to depict different stores efficiently and improve the accuracy of forecast.

Results

Measuring items similarity

The baseline is item-based CF, and the performance is measured by click-recall@top-N on candidate set: $\frac{\sum_{u_i}\#\{\text{hits in top-N}\}_{u_i}}{\sum_{u_i}\#\{\text{total clicks}\}_{u_i}} \times 100\%$.

top-N	30	40	50	60	70	80	90	100	1000
CF	8.46	10.39	12.12	13.65	15.05	16.32	17.49	18.57	42.33
ITEM2VEC	13.29	15.58	17.48	19.04	19.81	20.55	21.23	21.88	43.34

Table 1: The click-recall@top-N for different methods (higher is better)

In online system, by integrating new similarity scores into original scores, the final recall is increased by **24.0%**.

Transferring from seen items to unseen items

Our method is compared to baselines where candidate sets consist of random or hot items.

top-N	30	40	50	60	70	80	90	100	1000
RANDOM	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.08	0.70
HOT	3.19	3.79	4.35	4.83	5.42	6.00	6.53	7.13	27.75
NEW2VEC	12.27	14.02	15.33	16.37	16.47	16.56	16.66	16.75	21.89

Table 2: The click-recall@top-N for different methods (higher is better)

Transferring across different domains

An A/B test was conducted, and the performance is measured by Pay-Per-Impression (PPM). The baseline is hot-items list. Compared to baseline, naive Average increases PPM by **71.4%** and Weighted-average increases PPM by **141.8%**.

Transferring across different tasks

The performance is measured by Relative Mean Absolute Error (RMAE): $\text{RMAE} = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{\sum_{i=1}^{N} y_i} \times 100\%$.

dataset	day 1	day 2	day 3
HISTORY	43.23	40.75	34.26
HISTORY with ONE-HOT	42.57	39.00	34.57
HISTORY with VEC	40.95	33.75	33.02

Table 3: The RMAE scores for different methods (lower is better)