Two Tower Model with Self-Supervision Learning

Friday, March 15, 2024 3:50 PM

Two Tower Model Issues:

① A few popular items takes most clicks

② Large number of items get a few clicks.

③ The representation vector of popular items is learned well

However, the vector of $\underbrace{\text{unpopular items}}_{\text{Take large fraction}}$ is poorly learned.

How to solve? data argumentation

Two Tower Model Training:

Batch Examples:

users	items
#1	#1
#2	#2
⋮	⋮
#N	#N

Positive Samples: user ID

(1, 1), (2, 2), ... (N, N)

item ID

Negative Samples:

(1, 2), (1, 3) ..... (1, N)

(2, 1), (2, 3) ..... (2, N)

⋮

(N, 1) (N, 2) ..... (N, N-1)

# of positive samples: N # of negative: N(N-1)

objective: make $\cos(a_i, b_+)$ large; make $\cos(a_i, b_-)$ small

Two Tower Model with Self-Supervision Learning

Friday, March 15, 2024 3:50 PM

loss function:

positive sample

negative sample

observation: y_i : 1, 0, 0, ... 0

prediction: P_i : $P_{i,1}$ $P_{i,2}$ $P_{i,3}$ ..... — $P_{i,N}$

Softmax Activation

$\cos(a_i, b_1)$ $\cos(a_i, b_2)$ $\cos(a_i, b_3)$ .... $\cos(a_i, b_N)$

positive samples

negative samples

Cross Entropy Loss (y_i, P_i)

$$L = -\log P_{i,i} = -\log \frac{\exp(\cos(a_i, b_j))}{\sum_{j=1}^{n} \exp(\cos(a_i, b_j))}$$

Probability of item j being selected:

$$P_j \propto (\#\text{ of click})$$

Estimated like from user i to item j: $\cos(a_i, b_j)$

Corrected like: $\cos(a_i, b_j) - \log P_j$

Two Tower Model with Self-Supervision Learning

Friday, March 15, 2024 3:50 PM

Training Process:

① Randomly select N users and their interacted items

$(a_1, b_1), (a_2, b_2), \cdots (a_N, b_N) \longrightarrow$ one batch

② Loss function:

$$L_{main}[i] = -\log \frac{\exp[\cos(a_i, b_i) - \log p_i]}{\sum_{j=1}^{n} \exp[\cos(a_i, b_j) - \log p_j]}$$

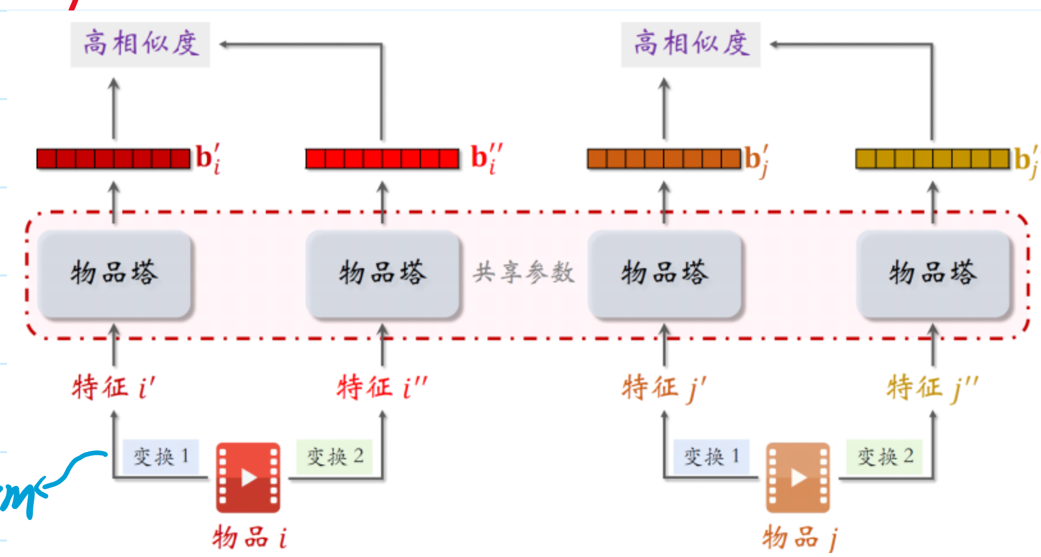
for user i

③ gradient descent:

$$\frac{1}{N} \sum_{i=1}^{N} L_{main}[i]$$

Self-supervision:

Transform



(Picture from Shusen Wang on Youtube/Bilibili)

① make $\cos(b_i', b_i'')$ and $\cos(b_j', b_j'')$ large

② make $\cos(b_i', b_j')$ and $\cos(b_i', b_j'')$ small

same item: high similarity; different item: low similarity

Two Tower Model with Self-Supervision Learning

Friday, March 15, 2024 3:50 PM

Transform #1: Random mask

① select some categorical features and mask them

e.g. item #1: genre = (photo, beauty, travel)

→ mask: genre = ("missing")

Transform #2: Drop out.

① randomly drop part of features

e.g. item # genre = (photo, beauty, travel)

→ drop: genre = (photo)

Transform #3: Complementary:

e.g. items: (ID, genre, hashtag, city)

two groups $\begin{cases} \text{(ID, hashtag)} \\ \text{(genre, city)} \end{cases}$

each item has two representation:

$\left.\begin{array}{l} \text{(ID, "missing", hashtag, "missing")} \rightarrow \text{item tower} \\ \text{("missing", genre, "missing", city)} \rightarrow \text{item tower} \end{array}\right.$

vector representation similar

Two Tower Model with Self-Supervision Learning

Friday, March 15, 2024 3:50 PM

Transform #4: mask correlated features.

① gender: $U = \{male, female, neutral\}$

② genre: $V = \{cosmetics, digit, football, tech\}$

③ $u = male$ and $v = digit$ probability is large

④ $u = male$ and $v = cosmetics$ probability is small.

mutual information:

$$MI(U,V) = \sum_{u \in U} \sum_{v \in V} p(u,v) \cdot \log \frac{p(u,v)}{p(u) \cdot p(v)}$$

k features → $k \times k$ matrix for MI

① random select one feature as "seed feature"

② get most relevant $k/2$ features

③ mask "seed" and $k/2$ features

Pros: works best

Cons: implementation is difficult; hard to maintain

Overall: NOT efficie

Two Tower Model with Self-Supervision Learning

Friday, March 15, 2024 3:50 PM

Model Training: for self supervision

1. uniformly select m items (NOT based on # of clicks)
2. perform two types of transformations
   item tower generates two groups of vectors
   $(b_1', b_2', \cdots b_m')$ $(b_1'', b_2'', \cdots b_n'')$
3. loss function for item i:

$$L_{self}[i] = -\log \frac{\exp(\cos(b_i', b_i''))}{\sum_{j=1}^{m} \exp(\cos(\underbrace{b_i', b_j''}_{\text{i and j similarity}}))}$$

4. objective function:

$$\frac{1}{m} \sum_{i=1}^{m} L_{self}[i]$$

Self supervision + Two Tower Model:

1. random sample n (user, item) from clicked items
2. uniform sample m items from all items
3. gradient descent:

$$\underbrace{\frac{1}{n} \sum_{i=1}^{n} L_{main}[i]}_{\text{two tower}} + \alpha \underbrace{\frac{1}{m} \sum_{j=1}^{m} L_{self}[j]}_{\text{self supervision}}$$

hyper parameter