Memory Network for Recommander System

The Latest Progresses

Jiarui Qin

Apex Data and Knowledge Management Lab
Shanghai Jiao Tong University

Jul. 20, 2018

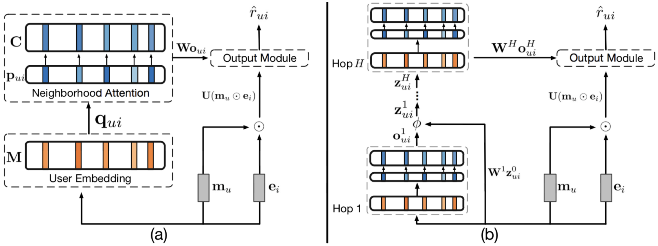
Outline

Motivation

- To preserve the history information of user's interaction with the recommander system.
- Dynamic user or item representation which is more suitable for the scenario of recommanding task. Users tend to have very diverse interests and it is always changing.
- Then why not LSTM? All history information is combined together which is not flexible enough and lacks of interpretability.

Outline

Collaborative Memory Network for Recommendation Systems



We have a user embedding matrix M, an item embedding matrix E.

User Embedding

We form a user preference vector q_{ui} where each dimension q_{uiv} is the similarity of the target user us level of agreement with user v in the neighborhood given item i

$$q_{uiv} = m_u^T m_v + e_i^T m_v \quad \forall v \in N(i) \tag{1}$$

Then we can calculate neighborhood attention

$$p_{uiv} = \frac{\exp(q_{uiv})}{\sum_{k \in N(i)} \exp(q_{uik})} \tag{2}$$

Next we construct the final neighborhood representation by interpolating the external neighborhood memory with the attention weights

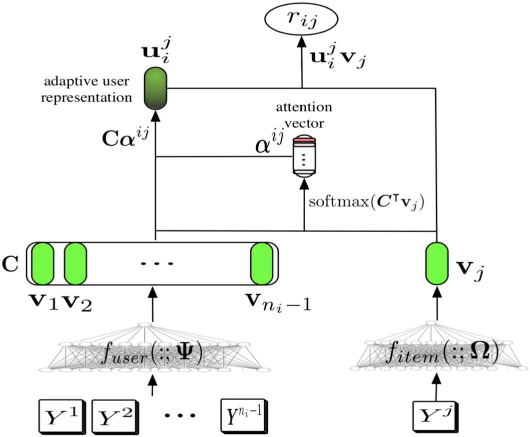
$$o_{ui} = \sum_{v \in N(i)} p_{uiv} c_v \tag{3}$$

And the final output(score) is

$$r_{ui} = v^T \phi(U(m_u \odot e_i) + W o_{ui} + b) \tag{4}$$

Outline

MARS: Memory Attention-Aware Recommender System

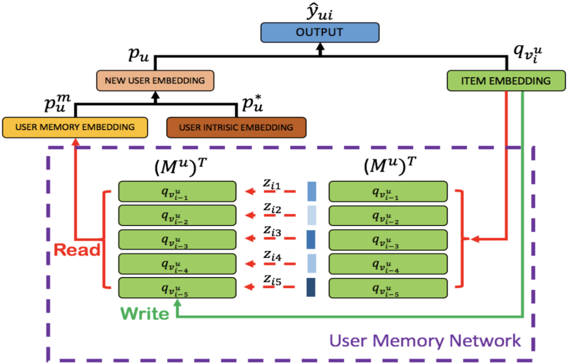


Item Embedding

Item-Level

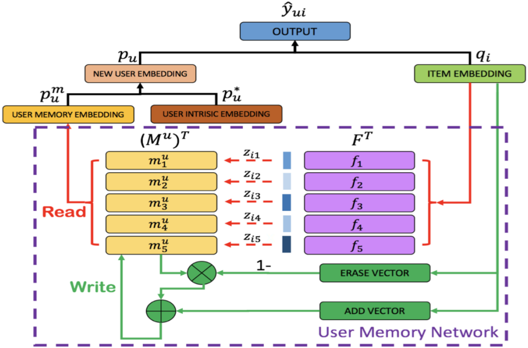
Y^n is the words that describe the item. Learn the embedding of the items v_i through CNN. Use attention mechanism to get a dynamic representation of a user. This model directly store the embedding of items in the memory module C, and the user embeddings are come from item embeddings in the memory module.

Sequential Recommendation with User Memory Networks



This model also stores item embeddings in memory module. It stores the most recent K items in the sequence of the user's interaction with the recommander system.
The user embedding p_u consists of two components: independent embedding p_u^* and the embedding got from memory module by attention machanism p_u^m.

Sequential Recommendation with User Memory Networks

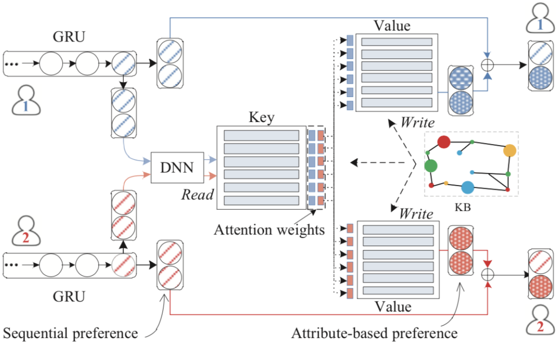


Item Embedding

Feature-Level

Previous item-level memory module has limits: only stores recent K items, not fine-grained enough. It is better if we can turn the items that the user likes to features that the user likes.
So, we get a global latent feature table F. We use these feature vectors as a "key"(feature name) to the memory module. And we use the same softmax attention machanism to read the "values"(feature value) from the memory module.

Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks



Item Embedding
Knowledge Based Enhanced

In the previous model, the features table lacks of interpretability. So, the new module makes the keys of the memory module explanable by adding a knowledge based pretrainning.
It uses a knowledge graph which contains $< e_1, r, e_2 >$ triplets, eg.$< Avator, directedby, James\ Carmelon >$. It pretrains the embeddings of e_i and r_i, and uses r_i as keys of memory module.

What to Use as Query to Memory Network

- Use user embedding vector as query: the first and the last model mentioned above.
- Use item embedding vector as query: the other models mentioned above.

Outline

Read

All the models use attention mechanism to read from memory module. Say the keys of memory are k_i, the query vector q. And we got similarity representation $s_i = q^T k_i$, then use softmax to generate weights:

$$w_i = \frac{\exp(s_i)}{\sum_j exp(w_j)} \tag{5}$$

and read value:

$$v^m = \sum_{i=1}^{N} w_i * v_i \tag{6}$$

Outline

Write

- No explicit write operation. The memory module is updated in the process of back propagation. Eg.the first and second model mentioned above.
- Explicit write operation. Write to the memory module when the user interacts with an item. The detail of the design is denpendent on specific settings of each different model. Eg.the last three models mentioned above.

Training and Evalution

- Loss function: pairwise loss of BPR, cross entropy loss.
- Evaluation metrics: Hit Ratio, NDCG, etc.