Memory Network for Recommender Systems

The Latest Progresses

Jiarui Qin

hongzhu.qjr@alibaba-inc.com
Apex Data and Knowledge Management Lab
Shanghai Jiao Tong University

Sept.18 , 2018

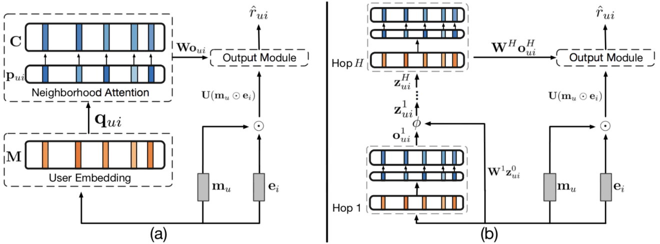
Outline

Motivation

- Memory network provides external spaces to store representation of knowledge to track long-term dependencies
- RNN-based models tend to compress all of a user's previous records into a fixed hidden representation, leading to low capacity to discriminate different historical records
- Since the state vector is encoded in a highly abstractive way, it is difficult to capture or recover fine-grained user preference from the interaction sequence. Moreover, it lacks of interpretability

Collaborative Memory Network: Structure

A unified hybrid model which capitalizes on the recent advances in memory networks and neural attention mechanisms for **collaborative filtering**. There are user memory M and C, item memory E storing embeddings. M and C can be regarded as internal and external user memorys. They allow storage of different aspects of the user preferences



T.Ebesu et al.Collaborative Memory Network for Recommendation Systems. SIGIR, 2018.

Collaborative Memory Network: Read&Write Mechanism

We form a user preference vector q_{ui} where each dimension q_{uiv} is the similarity of the target user u's 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 weights through softmax

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

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

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

All of the memory modules are updated through BP

Collaborative Memory Network: Read&Write Mechanism

This model is a hybrid model of latent factor model(MF) and neighbor-based model(KNN) which are commonly used in recommender systems

So, the output recommend score is composed of two parts:

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

Collaborative Memory Network: Multiple Hops

Extend the model to handle an arbitrary number of memory layers or hops to enhance the modeling capacity

$$z_{ui}^{h} = \phi(W^{h} z_{ui}^{h-1} + o_{ui}^{h} + b^{h}) \tag{5}$$

The initial layer state vector $z_{ui}^{0} = m_u + e_i$, and to calculate o_{ui}^{h} we need q_{ui}^{h},

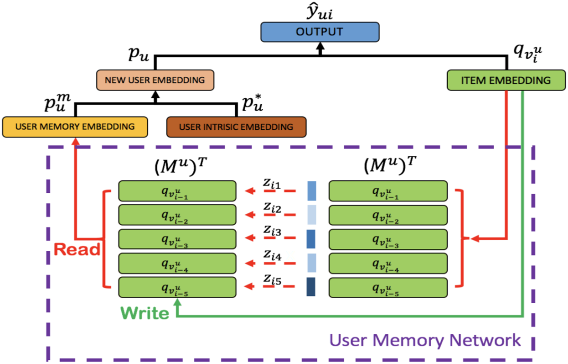
$$q_{uiv}^{h+1} = (z_{ui}^{h})^{T} m_v \tag{6}$$

The final score is

$$r_{ui} = v^{T} \phi(U(m_u \odot e_i) + W^{H} o_{ui}^{H} + b) \tag{7}$$

Outline

The memory module M^u stores the latest K items a user has interacted with. Use the history items to model the user's dynamic interests



X.Chen et al. Sequential Recommendation with User Memory Networks. WSDM, 2018.

Read operation is soft read through softmax attention mechanism,

$$w_{ik} = (q_{v_i^u})^T m_k^u, \;\; z_{ik} = \frac{\exp(\beta w_{ik})}{\sum_j \exp(\beta w_{ij})} \tag{8}$$

and the user's dynamic representation is

$$p_u^m = \sum_{k=1}^{K} z_{ik} m_k^u \tag{9}$$

The user representation p_u consists of two components:

$$p_u = MERGE(p_u^*, p_u^m) \tag{10}$$

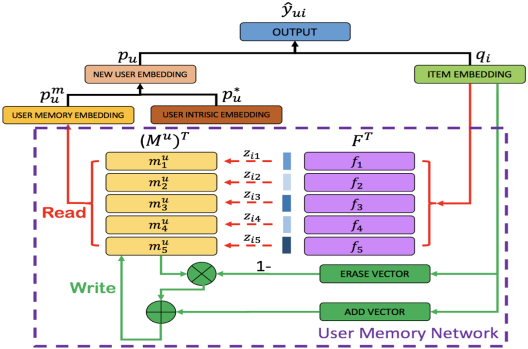
S.Rendle et al. Factorizing personalized markov chains for next-basket recommendation. WWW, 2010.

Write operation is simple: directly append the new item's embedding into M^u using FIFO mechanism

Outline

Instead of directly storing the latest K items that are clicked, the memory module stores items on feature level. There are K feature slots in the memory module and each item is regarded as a combination of the K features.

Structure of the memory module is K-V storage



Read Operation: Latent feature as key

$$w_{ik} = q_i^T f_k, z_{ik} = \frac{\exp(\beta w_{ik})}{\sum_j \exp(\beta w_{ij})} \tag{11}$$

$$p_u^m = \sum_{k=1}^{K} z_{ik} m_k^u \tag{12}$$

Write Operation: Erase first before new infomation is added

$$erase_i = \sigma(E^T q_i + b_e) \tag{13}$$

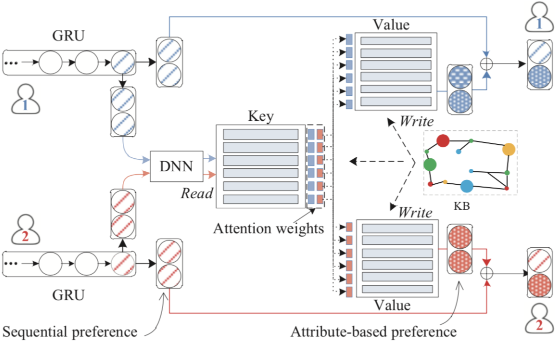
$$m_k^u \leftarrow m_k^u \odot (1 - z_{ik} \cdot erase_i) \tag{14}$$

$$add_i = \tanh(A^T q_i + b_a), \ m_k^u \leftarrow m_k^u + z_{ik} \cdot add_i \tag{15}$$

A.Graves et al. Neural turing machines. arXiv preprint, 2014.
J.Zhang et al. Dynamic Key-Value Memory Networks for Knowledge Tracing. WWW, 2017.

Outline

This model uses a knowledge base to do pre-training, and the keys in the K-V memory module is the **relation** in the knowledge graph



J.Huang et al. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. SIGIR, 2018.

To learn KB embeddings, the model uses a simple and efficient method: **TransE**

The knowledge base is consist of many triples: $< e_1, r, e_2 >$, and **TransE** just minimizes loss:

$$\sum_{\{<e_1, r, e_2>\}} ||e_1 + r - e_2|| \tag{16}$$

which provides a general and compact representation for entities and relations

Just use the **relations as keys** in the memory module

Read Operation:

$$m_t^u \leftarrow \sum_{a=1}^{A} w_{t,u,a} * v_a^u \tag{17}$$

where,

$$w_{t,u,a} = \frac{\exp(\gamma \tilde{h}_t^u \cdot k_a)}{\sum_{attr=1}^{A} \exp(\gamma \tilde{h}_t^u \cdot k_{attr})} \tag{18}$$

The model uses a GRU to model the user representation w.r.t time:

$$h_t^u = GRU(h_{t-1}^u, q_{i_t}; \Theta), \;\; \tilde{h}_t^u = MLP(h_t^u) \tag{19}$$

Write Operation: The update from item i relative to attribute a is computed as

$$e_a^i = e_i + r_a \tag{20}$$

eg. $e_{Avatar} + r_{directedby} = e_{JamesCameron}$
Still, soft write

$$z_a = sigmoid((v_a^u)^T \cdot e_a^i) \tag{21}$$

$$v_a^u \leftarrow (1 - z_a) \cdot v_a^u + z_a \cdot e_a^i \tag{22}$$

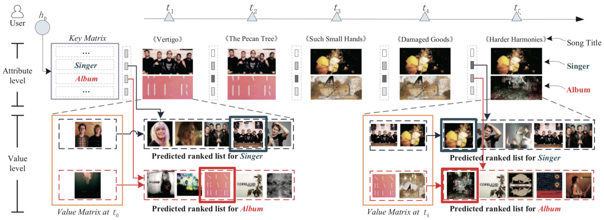
J.Weston et al. Memory Networks. ICLR, 2015.

The recommend score is

$$s_{u,i,t} = MLP(p_t^u)^T \cdot MLP(\tilde{q}_i) \tag{23}$$

where, $p_t^u = concat(h_t^u, m_t^u)$ and $\tilde{q}_i = concat(q_i, e_i)$

A user listened 5 songs from different albums and singers. The experiment shows that the model effectively learns the pattern of user interest



From attribute level, the model puts more attention on Album and then on Singer

From value level, the user-specific memory of Album and Singer will be close to the actual user interest values on different attribute

Outline

Read&Write Mechanisms

- Read Operation: use query embedding(from user side or item side) to calculate the attention weights, and weighted sum the contents in memory slots
- Write Operation: Three ways: BP, directly append, soft write

Outline

Comparison of Similar Models

Table: Summary of Related Models

Paper No.	Type	Mem Storage	Query	Read	Write	Similarity Calculation	Other Spotlight
#1	CMN	user&item	user emb to query user mem	attention weighted sum	BP	inner product	–
#2	CMN	item	item emb to query item mem	attention weighted sum	BP	inner product	–
#3	Feature-Level	history-item	GRU hidden state to query history mem	attention weighted sum	soft write	inner product	KB
#4	Item/Feature-Level	history-item	item emb to query history mem	attention weighted sum	append/soft write	inner product	–
#5	CMN	item	item emb to query item mem	attention weighted sum	BP	inner product	–
#6	Item-Level	history-item	item emb to query history mem	attention weighted sum	append	attention network	short+long term memory
#7	Feature-Level	history-item	user emb to query history mem	attention weighted sum	soft write	Euclidean distance	adversarial negative sampling

#1 T.Ebesu et al. Collaborative Memory Network for Recommendation Systems. SIGIR, 2018.
#2 L.Zheng et al. MARS: Memory Attention-Aware Recommender System. arXiv preprint, 2018.
#3 J.Huang et al. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. SIGIR, 2018.
#4 X.Chen et al. Sequential Recommendation with User Memory Networks. WSDM, 2018.
#5 D.Gligorijevic et al. Modeling Mobile User Actions for Purchase Recommendation using Deep Memory Networks. SIGIR, 2018.
#6 Q.Liu et al. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. SIGKDD, 2018.
#7 Q.Wang et al. Neural Memory Streaming Recommender Networks with Adversarial Training. SIGIR, 2018.

Limitation of Current Works

- The representation of item is too simple and lacks of information from users who have visited the item
- User-specific memory module lacks the ability to use collaborative information from similar users which is important in recommendation tasks
- Erase operation can't reflect the interest evolving phenomenon
- Collaborative memory network doesn't have effective history information storage