Sequential Recommendation for Modeling Preference Dynamics and Feature Interaction

Qi Zhang and Longbing Cao

Advanced Analytics Institute, University of Technology Sydney

Background and Problem

1) User and item interactions are embodied through user/item features and their relations and a user's general preference at a point of time is embodied through his/her preferred item features (e.g., animated items fit children while artists may prefer musical movies).

2) A user's preference may change over time, driven by multiple aspects of dynamics, e.g., the introduction of new products and contextual change (such as successive releases of new movies in a year), which drive user's preference change over time.

Accordingly, both user/item feature relations and user's preference dynamics should be accounted in modeling user's preference.

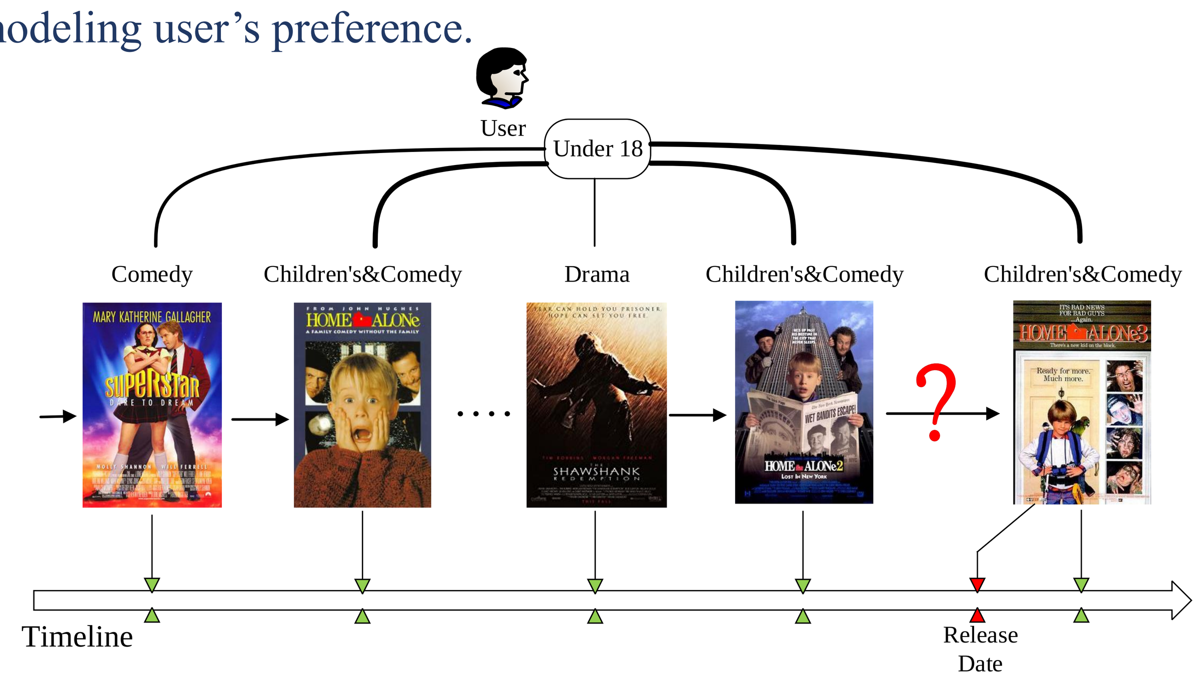

Fig. 1. An example to illustrate how our model works. A user under 18 years old watched a series of movies (labeled movie genre) in order. Green arrows denote the time points when a movie was watched, and the lines between user/item features denote the relations where thicker lines mean stronger relations. The film Home Alone 3 may be highly recommended to the user at a time point after its release because: (1) the user prefers Children's and comedy movies, (2) a sequential pattern exists from Home Alone 1 to Home Alone 3, and (3) current time point is right after the release date of the Home Alone 3.

Model

We elaborate the design of CORN which incorporates a CNN to learn user's preference dynamics and a latent factor model to capture user's general preference. Shown in Figure 2, CORN consists of four components: Embedding Layers, Convolutional Layers, Latent Interaction Layers, and Output Layers.

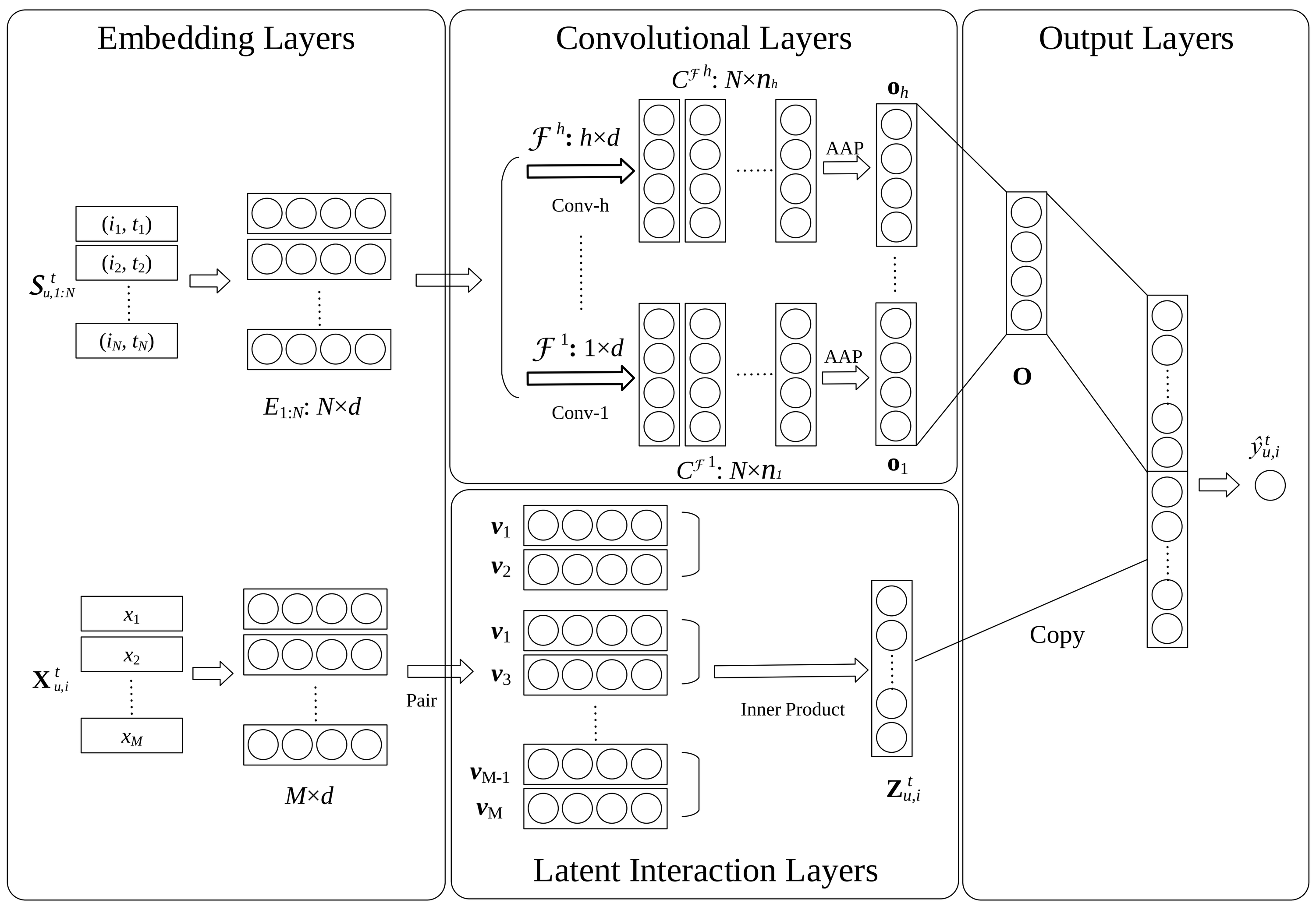

Fig. 2. The CORN architecture for context-aware modeling of user's action sequence and user/item feature relations.

Modeling Preference Dynamics

Leveraging the advantages of CNN in capturing local features and relations, our method applies a CNN with multi-height filters to discover multifold sequential patterns of user actions on items. Before introducing this CNN, we first propose the representations of relative temporal context.

1) Relative temporal context representations

To make models sensitive to a temporal context, we introduce a relative time embedding method which embeds the time of the current action and the time interval between the current time and the previous action's time in the same low-dimensional space. Here, the time interval can be interpreted as the time period of user absence from the system and supplements temporal relations between successive actions, which is formulated as below:

$$\Delta t_i = \lfloor log(t_i - t_{i-1} + 1) \rfloor$$

Note that the converted integers Δt_i have an upper bound and are enumerable, since the timestamp t_i is bounded by the max timestamp in each application.

Then given any clipped N-action sequence $S_{u,1:N} = \{(i_1, t_1), \cdots, (i_N, t_N)\}$ of user u, the sequence is represented by the following embedding matrix (for concision, we omit the subscript u and time step in the following):

$$E_{1:N} = \{e_1, e_2, \cdots, e_N\}, \qquad s.t.\ e_j = v_j * (\tau_j + \Delta\tau_j + 1),$$

where $v_i, \tau_i, \Delta\tau_i$ are embeddings vectors for the item i_j, the timestamp t_j and the time interval Δt_j respectively.

2) Multi-filter convolutional neural network

By assuming that a user's choice at time t is influenced by his/her recent actions, we generate a high-level representation of the previous action sequence to model user's preference dynamics.

Considering the existence of multifold sequential patterns, we leverage the CNN with multiple filters of different heights which are formulated below:

$$F = F^1 \cup F^2 \cup \cdots \cup F^H,\ s.t.\ F^h = \{f^k \in \mathbb{R}^{h \times d} \mid k \in [1, n_h], h \in [1, H]\},$$

where H is the maximum filter height, h is the height of a filter and n_h denotes the number of filters with a height h. Intuitively, each filter f^k treats the embedding vector of each action as a whole and covers h actions each step when sliding over the embedding matrix. Accordingly, given $E_{1:N}$, the i-th convolution feature by wide convolution is given by:

$$c_i^k = \phi_a\left(f^k \cdot E_{i:i-h+1} + b_c\right), s.t., f^k \in F, 1 \le i \le N,$$

operator $\cdot$ computes the inner-product, b_c is a bias term, and a is a activation function. For all filters in F, the convolutional output is $C^F = \left[C^{F^1} C^{F^2} \cdots C^{F^H}\right]$ and we have:

$$C^{F^h} = \begin{bmatrix} c_i^{k_1} & \cdots & c_i^{k_j} \\ \vdots & \ddots & \vdots \\ c_N^{k_1} & \cdots & c_N^{k_j} \end{bmatrix}, s.t., f^{k_1}, f^{k_2}, \cdots, f^{k_j} \in F^h, j = n_h.$$

3) Attentive average pooling

Instead of adopting multilayer convolutional networks, we introduce an Attentive Average Pooling (AAP as shown in Figure 2) over each stacked convolutional output to capture the large-span feature combinations.

Empirical Results

- Three public datasets: MovieLen-1M, last.fm and Tafeng.
- Baselines: POP, FM, NFM, RRN and Caser
- Metrics: Mean average precision (MAP), HR@10 (Hit rate)

1) Ablation test

- larger embedding size is beneficial to improve performance;
- feature relations and preference dynamics can work collaboratively and complementary;
- CORN effectively captures and integrates the two aspects for the prediction.

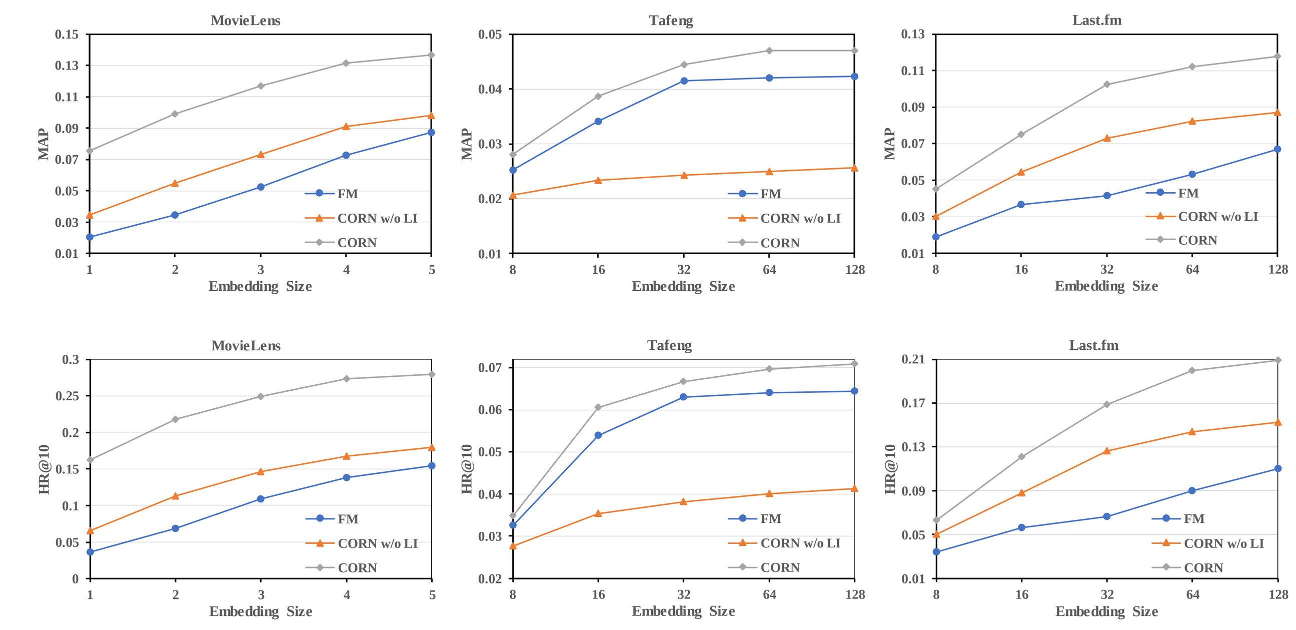

Fig. 3. Ablation test: HR@10 and MAP of FM, CORN w/o LI and CORN w.r.t. embedding sizes.

2) Performance Comparison

- the results indicate that CORN achieves the state-of-the-art performance.
- the result verifies the effectiveness of integrating feature relations and preference dynamics.
- explicit feature relations reflect an inherent driving force of user's preference on a certain item and contributes to predicting user's choices.

Table 1. CORN recommendation performance comparison with baselines.

Dataset	Metric	POP	FM	NFM	RRN	Caser	CORN-MP	CORN-IF	CORN-NT	CORN	Imp.
MovieLens	MAP	0.0201	0.0727	0.1028	0.1138	0.1231*	0.1269	0.1119	0.126	**0.1315**	+6.8%
	HR@10	0.0363	0.1379	0.1889	0.2402	0.2627*	0.2695	0.2397	0.2664	**0.2738**	+4.2%
Tafeng	MAP	0.0284	0.0421	0.0435*	0.0334	0.0326	0.0484	0.0437	0.0463	**0.0471**	+8.3%
	HR@10	0.0427	0.0629	0.0647*	0.0461	0.0472	0.0694	0.0651	0.0687	**0.0697**	+7.7%
Last.fm	MAP	0.0044	0.0533	0.0672	0.0952	0.0989*	0.1052	0.0991	0.1058	**0.1109**	+12%
	HR@10	0.0051	0.0899	0.1139	0.1782	0.1818*	0.1948	0.185	0.1951	**0.1997**	+9.8%

3) Interpretability

- first order interactions play no effects to the prediction of users' choices.
- inter-relations between users, items and context are more productive to the prediction

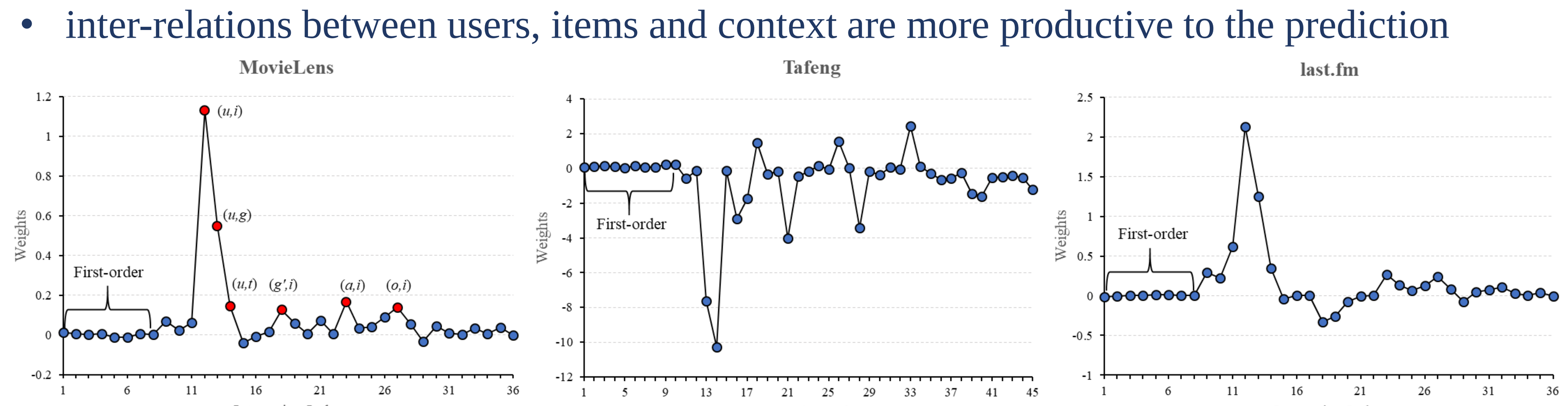

Fig. 4. Visualization of weight for MovieLens, Tafeng and last.fm.

Contact

For more information, please contact the author via email: qi.zhang-13@student.uts.edu.au or refer to 'Context-aware Convolutional Recommendation Network for Modeling Preference Dynamics and Explicit Feature Relations'.