Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding

Jiaxi Tang, Ke Wang
School of Computing Science, Simon Fraser University

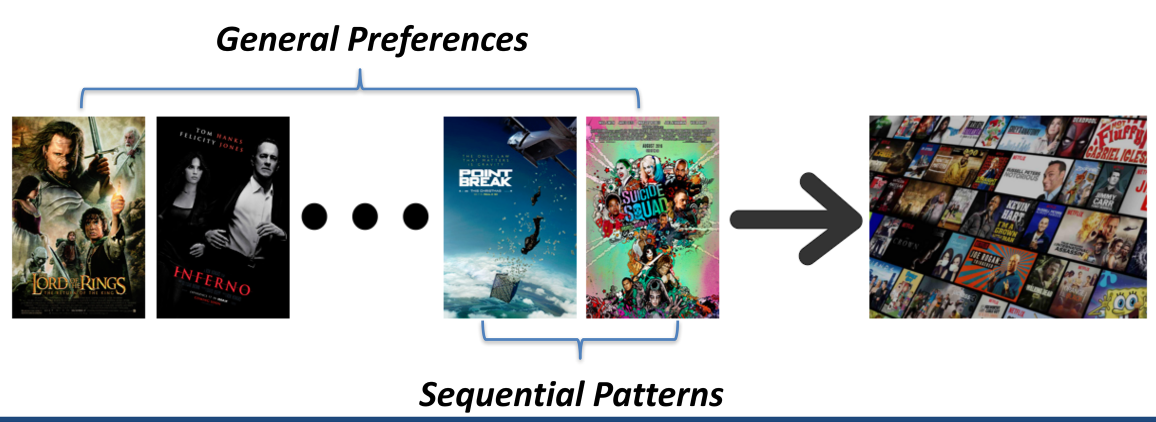
Abstract

- We study *sequential recommendation* problem as the order of interaction implies that sequential patterns play an important role on user's next action.
- Under such setting, sequential patterns should be carefully modeled, in both point-level and union-level.
- We propose a *Convolutional Sequence Embedding Recommendation Model* (Caser) to model the above two types of sequential patterns.
- The experiments on public data sets demonstrated that Caser consistently outperforms state-of-the-art sequential recommendation methods

Sequential Recommendation

Given a user's sequences Su , recommend a list of items that maximize her/his future needs, by considering both general preferences and sequential patterns.

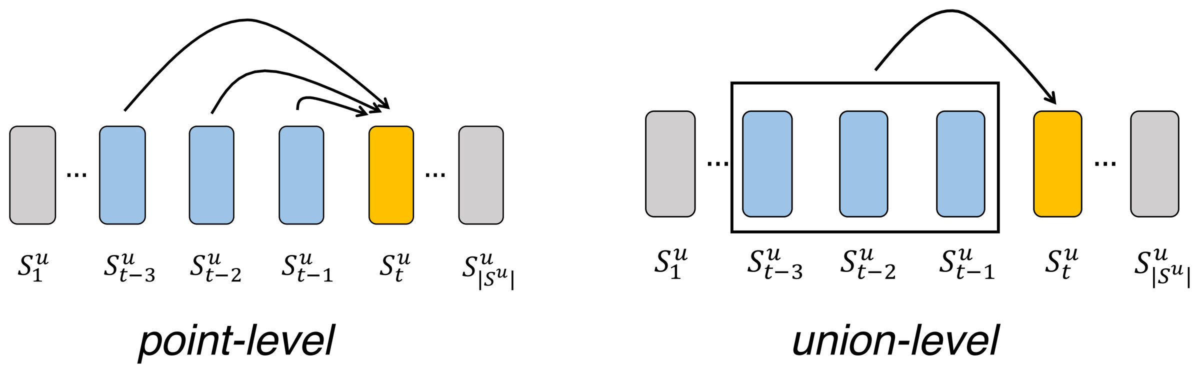
– **General Preferences:** represent user's long term and static behaviors and are unlikely to change in a short period of time.
– **Sequential Patterns:** represent user's short term and dynamic behaviors and come from a close proximity of time.



Related Works and Motivations

- *Existing works model sequential pattern in point-level, fail to model sequential pattern in union-level.*

– **point-level:** each of the previous actions influences the target action individually, instead of collectively
– **union-level:** several previous actions jointly influence the target action.

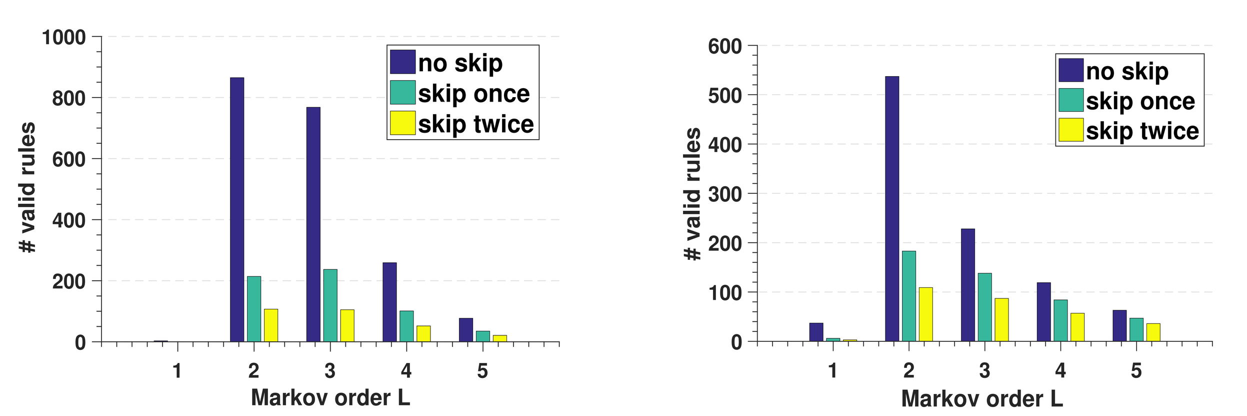


- *Find the existence of union-level sequential pattern.*

When we mine sequential association rules of the form

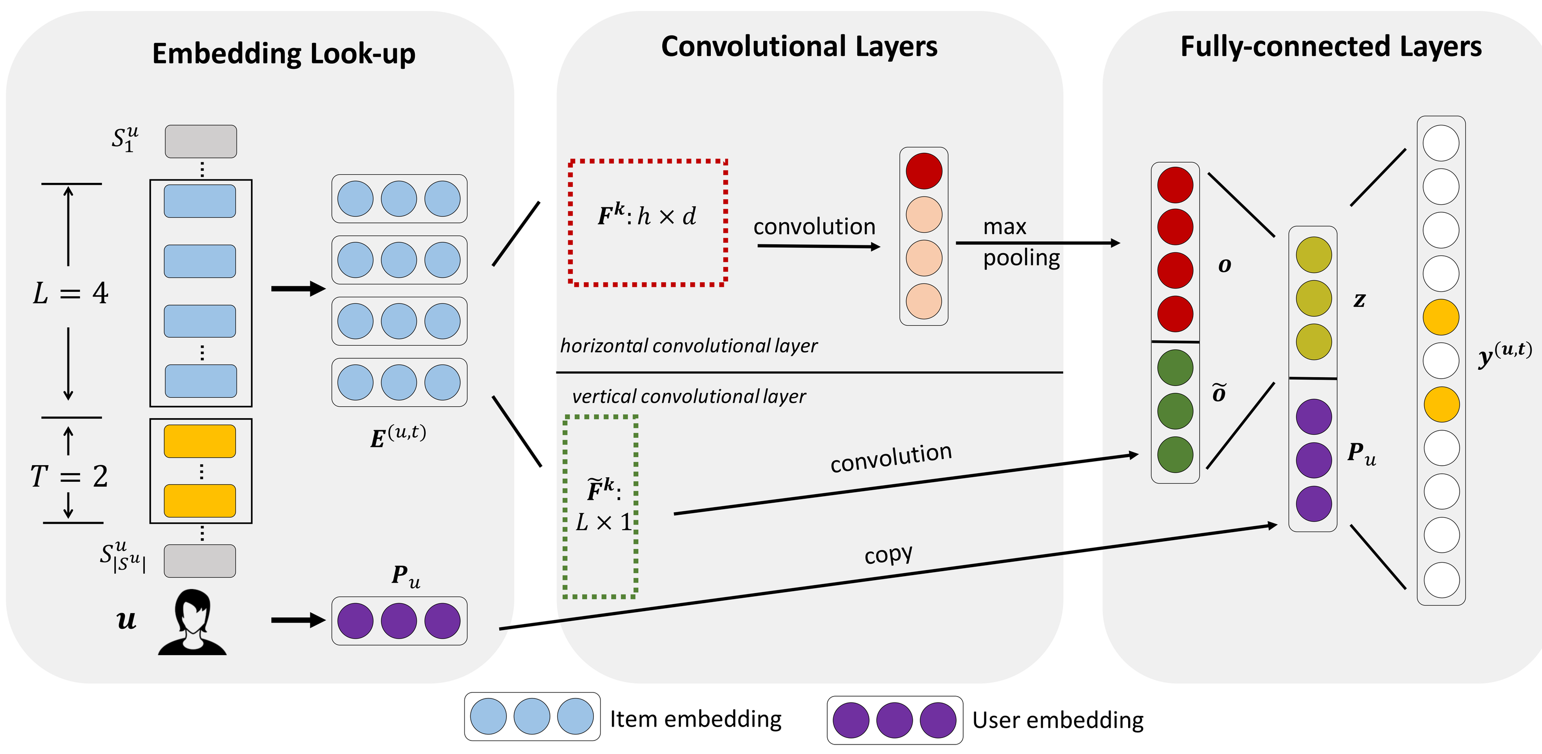
$$(S_{t-L}^u, \cdots, S_{t-2}^u, S_{t-1}^u) \to S_t^u.$$

With confidence=50% and support=5, most of the resulting rules have the length larger than 1 ($L > 1$), indicating the existence of union-level influences.



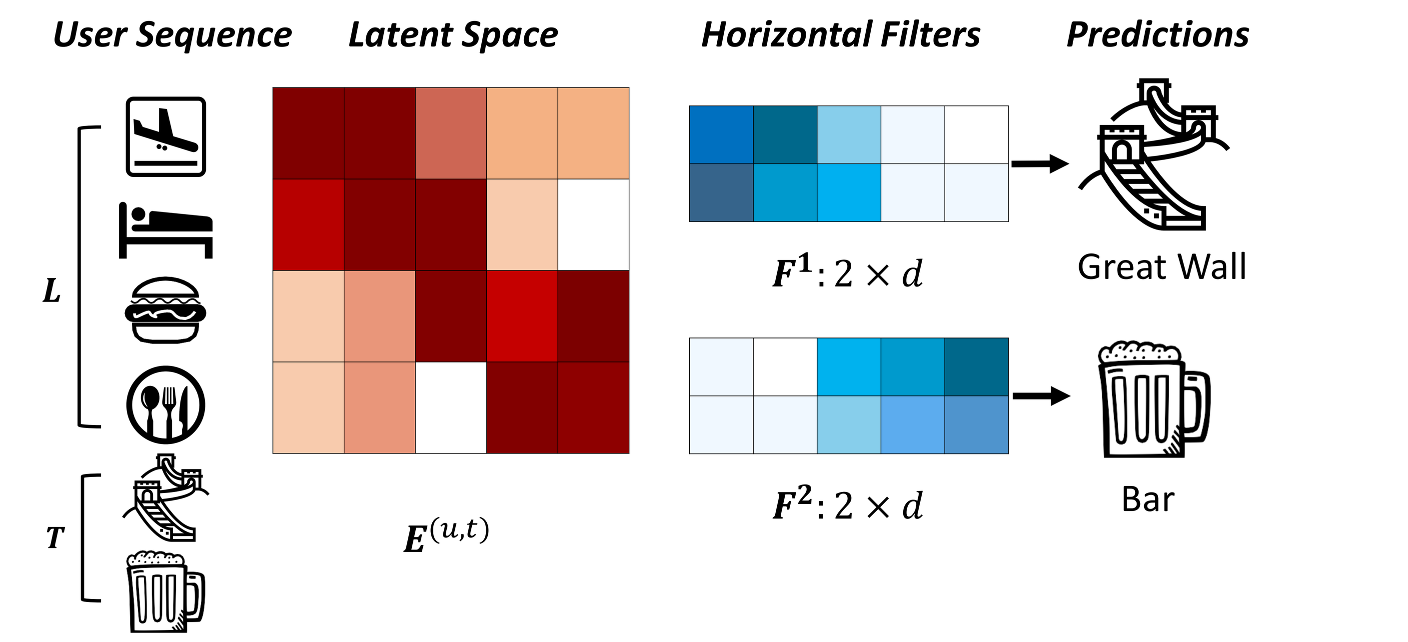
The network architecture of Caser

- *Convolutional Neural Network* (CNN) is used to capture both point-level and union-level sequential patterns.
- By incorporating *Latent Factor Model* (LFM), Caser is also able to capture user's general preferences.



Capture Union-level Sequential Pattern with Horizontal Convolutional Filters

- Borrow the idea of using CNN in text classification, we use convolutional filters to search for sequential patterns.
- Sliding horizontally (from top to bottom), the horizontal convolutional filters are used with different height (multiple union sizes) but same width (same to the latent dimension).
- *Max pooling* operation on the result for extracting the most significant feature from a particular filter.



The first horizontal filter picks up the union-level sequential pattern "(Airport, Hotel) → Great Wall" by having larger values in the latent dimensions where Airport and Hotel have larger values.

Capture Point-level Sequential Pattern with Vertical Convolutional Filters

- Sliding vertically (from left to right), the vertical convolutional filters have same height (*i.e., L*) and same width (*i.e., 1*).
- Vertical convolutional filters are learned to *aggregate* the latent embeddings of previous items.
- In other words, they are performing *weighted sum* over previous items' latent representations, thus capture point-level sequential pattern.

Network Training

- Extract every L item as input, and the next T items as targets.
- Sigmoid Negative Log-Loss with random negative sampling is used as optimization criterion.

Codes and Data are available at: http://www.sfu.ca/~jiaxit/

Experimental Setup

- *Datasets*: 4 datasets with large *Sequential Intensity* is used MovieLens, Gowalla, Foursquare and Tmall.

$$\text{Sequential Intensity } (SI) = \frac{\#\text{rules}}{\#\text{users}}.$$

Datasets	Sequential Intensity	#users	#items	avg. actions per user
MovieLens	0.3265	6.0k	3.4k	165.50
Gowalla	0.0748	13.1k	14.0k	40.74
Foursquare	0.0378	10.1k	23.4k	30.16
Tmall	0.0104	23.8k	12.2k	13.93

- *Baselines (non-sequential):* POP, BPR,
- *Baselines (sequential):* FPMC, Fossil and GRU4Rec

- *Evaluation Metrics:*

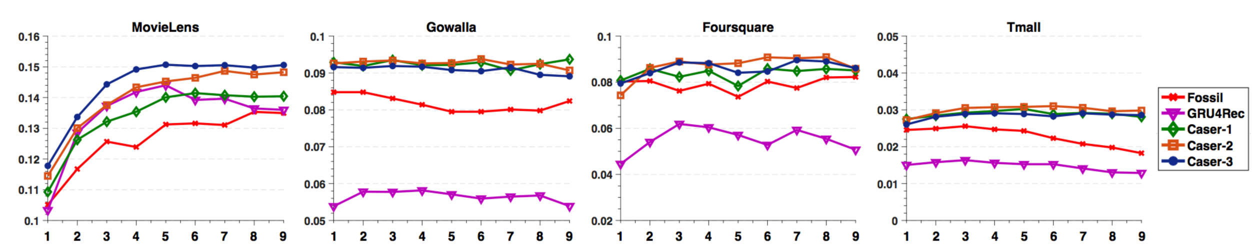
1) Precision@n ($n \in \{1, 5, 10\}$)
2) Recall@n ($n \in \{1, 5, 10\}$)
3) Mean Average Precision (MAP)

Experimental Results

- Overall performance:

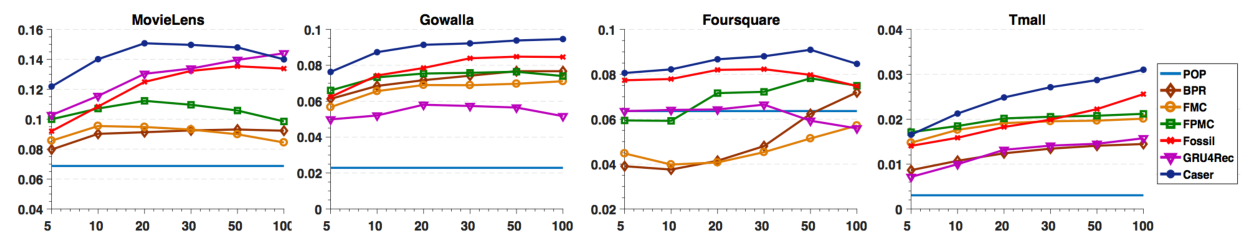
Dataset	Metric	POP	BPR	FMC	FPMC	Fossil	GRU4Rec	Caser	Improv.
MovieLens	Prec@1	0.1280	0.1478	0.1748	0.2022	0.2306	**0.2515**	0.2502	-0.5%
	Prec@5	0.1113	0.1288	0.1505	0.1659	0.2000	0.2146	**0.2175**	1.4%
	Prec@10	0.1011	0.1193	0.1317	0.1460	0.1806	0.1916	**0.1991**	4.0%
	Recall@1	0.0050	0.0070	0.0104	0.0118	0.0144	**0.0153**	0.0148	-3.3%
	Recall@5	0.0213	0.0312	0.0432	0.0468	0.0602	0.0629	**0.0632**	0.5%
	Recall@10	0.0375	0.0560	0.0722	0.0777	0.1061	0.1093	**0.1121**	2.6%
	MAP	0.0687	0.0913	0.0949	0.1053	0.1354	0.1440	**0.1507**	4.7%
Gowalla	Prec@1	0.0517	0.1640	0.1532	0.1555	0.1736	0.1050	**0.1961**	13.0%
	Prec@5	0.0362	0.0983	0.0876	0.0936	0.1045	0.0721	**0.1129**	8.0%
	Prec@10	0.0281	0.0726	0.0657	0.0698	0.0782	0.0571	**0.0833**	6.5%
	Recall@1	0.0064	0.0250	0.0234	0.0256	0.0277	0.0155	**0.0310**	11.9%
	Recall@5	0.0257	0.0743	0.0648	0.0722	0.0793	0.0529	**0.0845**	6.6%
	Recall@10	0.0402	0.1077	0.0950	0.1059	0.1166	0.0826	**0.1223**	4.9%
	MAP	0.0229	0.0767	0.0711	0.0764	0.0848	0.0580	**0.0928**	9.4%
Foursquare	Prec@1	0.1090	0.1233	0.0875	0.1081	0.1191	0.1018	**0.1351**	13.4%
	Prec@5	0.0477	0.0543	0.0445	0.0555	0.0580	0.0475	**0.0619**	6.7%
	Prec@10	0.0304	0.0348	0.0309	0.0385	0.0399	0.0331	**0.0425**	6.5%
	Recall@1	0.0376	0.0445	0.0305	0.0440	0.0497	0.0369	**0.0565**	13.7%
	Recall@5	0.0800	0.0888	0.0689	0.0959	0.0948	0.0770	**0.1035**	7.9%
	Recall@10	0.0954	0.1061	0.0911	0.1200	0.1187	0.1011	**0.1291**	7.6%
	MAP	0.0636	0.0719	0.0571	0.0782	0.0823	0.0643	**0.0909**	10.4%
Tmall	Prec@1	0.0010	0.0111	0.0197	0.0210	0.0280	0.0139	**0.0312**	11.4%
	Prec@5	0.0009	0.0081	0.0114	0.0120	0.0149	0.0090	**0.0179**	20.1%
	Prec@10	0.0007	0.0063	0.0084	0.0090	0.0104	0.0070	**0.0132**	26.9%
	Recall@1	0.0004	0.0046	0.0079	0.0082	0.0117	0.0056	**0.0130**	11.1%
	Recall@5	0.0019	0.0169	0.0226	0.0245	0.0306	0.0180	**0.0366**	19.6%
	Recall@10	0.0026	0.0260	0.0333	0.0364	0.0425	0.0278	**0.0534**	25.6%
	MAP	0.0030	0.0145	0.0197	0.0212	0.0256	0.0164	**0.0310**	21.1%

- Caser outperform other baselines with fewer parameters:



MAP vs. latent dimensions

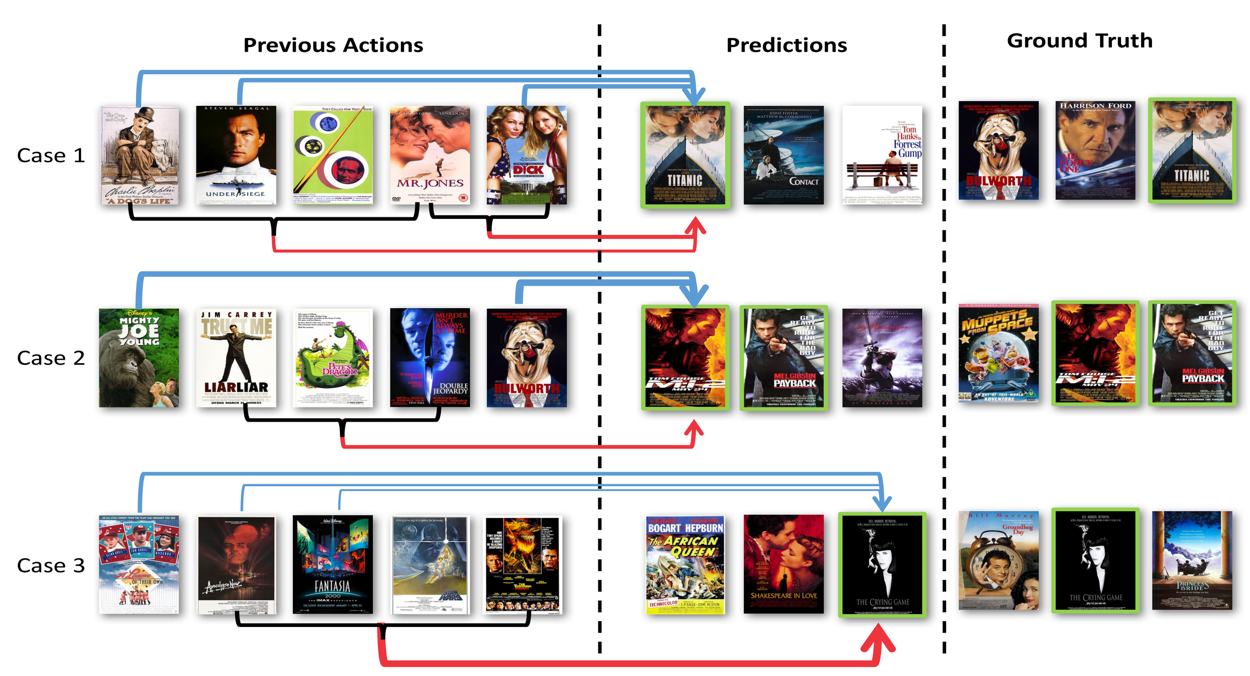
- Caser best utilizes the extra information provided by increasing number of items in the sequence:



MAP vs. sequence length

Case Studies

- Visualize the influences of Caser's prediction when masking out certain items within a sequence.



11th ACM International Conference on Web Search and Data Mining (WSDM 2018), Marina Del Rey, CA, USA