Temporal Learning and Sequence Modeling for a Job Recommender System

Kuan Liu, Xing Shi, Anoop Kumar, Linhong Zhu, and Prem Natarajan

Team: Milk Tea

Presenter: Huaigu Cao

Personalized Job Post Recommendation

Task:

- To recommend job posts to users on *Xing*,
- based on 1) interaction history and 2) user/item features.

Challenges:

- Large volume
  - 1.5M users, 1.3M items, 8.8M interactions, 200M impression
- Rich/Noisy user/item features available.
  - eps. categorical features. e.g. >100K text tokens
- Temporal dynamics/sequence form in interaction history.

Challenges (cont.)

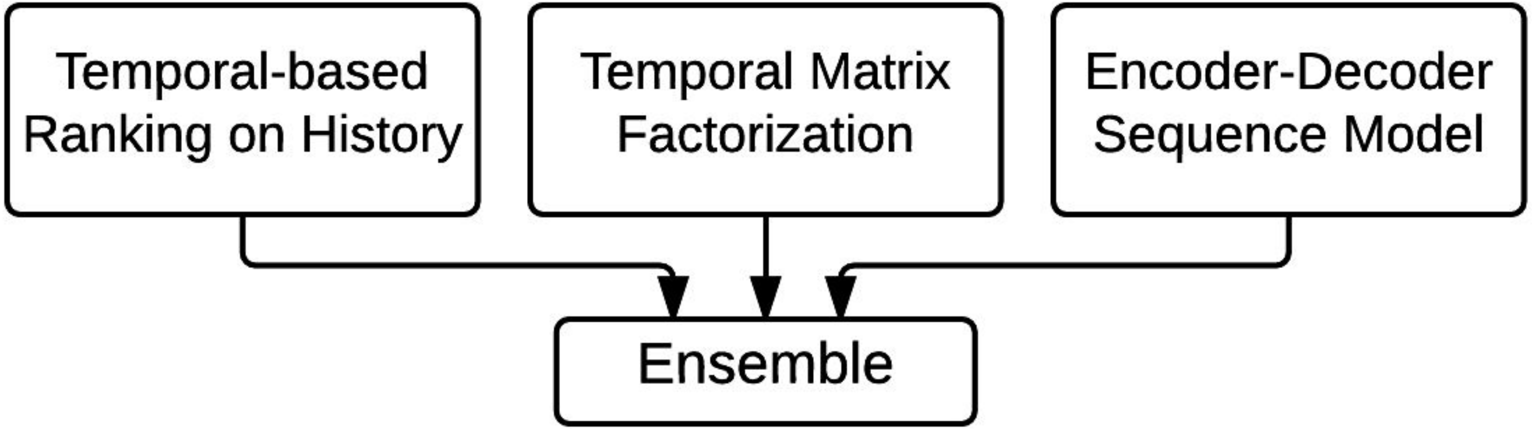
- **Temporal Dynamics**: *Time as a factor to influence a user's future behavior*.
    - Observation: users tend to re-interact with items that they did in the past.
        - e.g. on average 2 of 7 items in a user's Week 45 appeared in his past interaction list.
    - Observation: users are more influenced by what they interacted recently than long time ago.

How to explicitly model temporal influence?

- **Sequence Property**.
    - User-Item interactions are NOT i.i.d. Instead, a user interacts with *a sequence of items*.
    - Conjecture: Item sequence may contain additional useful information that helps improvement recommender systems. (e.g. temporal relation, item-item similarities.)

Does sequence really help? If so, how to model?

Approach Overview



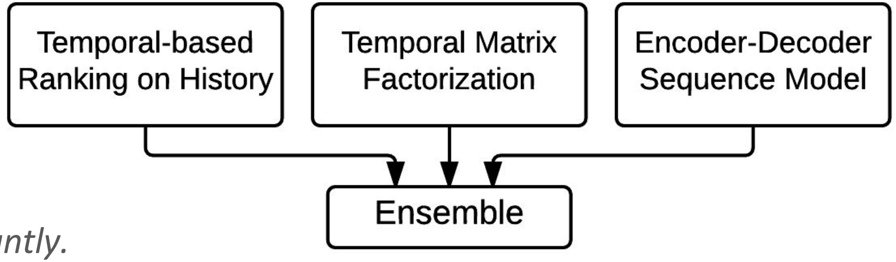
Approach Overview

- **Temporal Learning**
  - **A**. Temporal based Ranking
  - **B**. Temporal MF

Wins over non-temporal counterpart significantly.

- **Sequence Modeling**
  - **C**. LSTM based Encoder-Decoder model.

Beats the best MF model.



A. Temporal Ranking on Historical Items

Motivation:

- Users have a strong tendency to re-interact with items that they already did in the past.
- More recent interactions influence a user’s future behavior more.

Historical items are important! Recency of interaction matters!

Approach:

- A (time reweighted) linear ranking model.
- Minimize a loss incurred on carefully constructed triplet constraints.

A. Temporal Ranking on Historical Items (cont.)

Linear Ranking Model

$$S(u,i,t) = wM_{u,i,t}^{T} \qquad M_{u,i,t} \in \mathbb{N}^{K\times T}$$

$w(k,\tau)$ indicates the relative contribution of k-type interaction at time τ.

Model solving based on triplet constraints

The distribution between training and test stages as similar as possible!

$$\mathcal{T} = \{u \text{ prefers to re-interacting with } i_1 \text{ to } i_2 \text{ at time } \tau\}_{n=1}^{N}$$

Construct such constraints when **u interacted with i1, i2 before t**, but **only interacted with i1 at t**.

B. Temporal Matrix Factorization

- **Matrix Factorization**
  - To recommend new items

- **Hybrid Matrix Factorization** (HMF)
  - Learn categorical features
- **Temporal HMF** (THMF)
  - Re-weight loss of HMF by time

Hybrid Matrix Factorization (recap)

Users/Items are represented as sums of feature embedding. (b: bias.)

$$\vec{q}_u = \sum_{j \in f_u} \vec{x}_j^U, \vec{q}_i = \sum_{j \in f_i} \vec{x}_j^I; \qquad b_u = \sum_{j \in f_u} b_j^U, b_i = \sum_{j \in f_i} b_j^I$$

User-item score is given by inner product

$$S(u,i) = \vec{q}_u \cdot \vec{q}_i + b_u + b_i$$

Model is trained by minimizing the loss (we chose WARP) based on score and ground truth t

$$L = \sum_{\{u,i\} \in I} \ell(S(u,i), t(u,i))$$

Temporal Hybrid Matrix Factorization

A non-negative weight associated with time is placed in the loss

$$L' = \sum_{\{u,i,\tau\} \in I} \ell(S(u,i), t(u,i,\tau)) \times \gamma(\tau)$$

$\gamma(\tau)$ captures contribution of interactions over time. Zero weights in $\gamma(\tau)$ reduce training set size as well.

- Value of $\gamma(\tau)$.
  - in general can be learned jointly with other embedding parameters.
  - in our experiment **are fixed as learned weights** in Model **A**. (to speed up training) and give good performance.

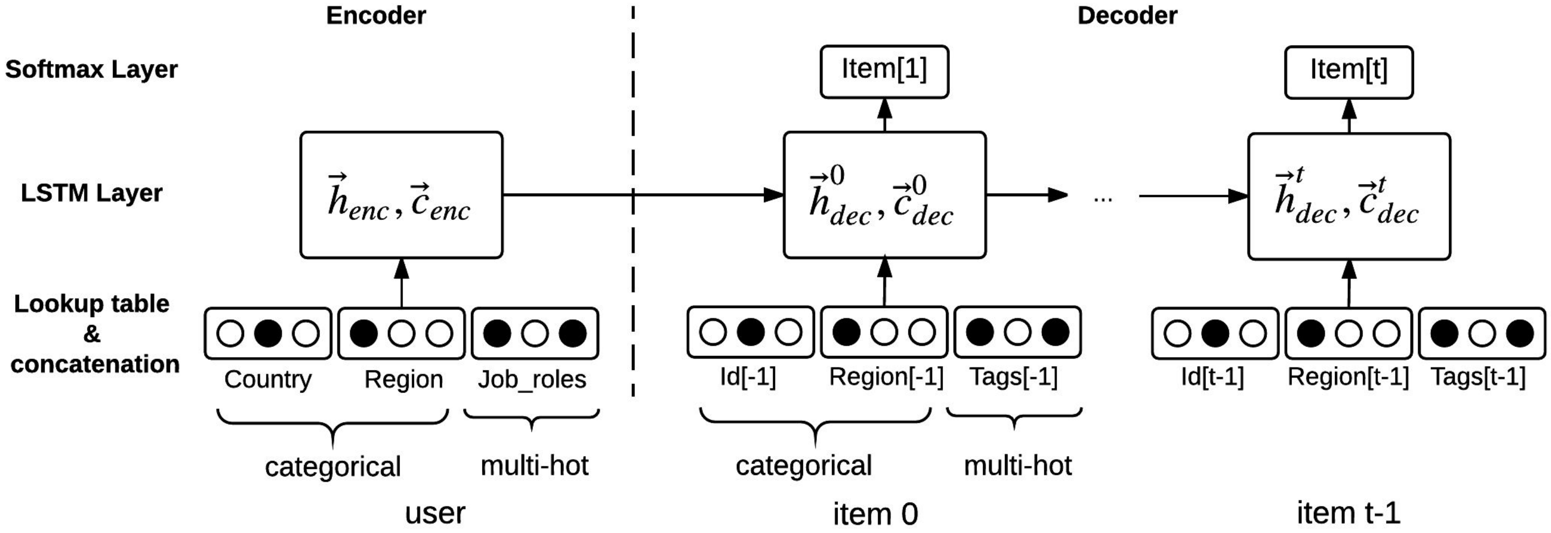
C. Sequence Modeling

- **Sequence of items ordered by time**:

  USER 1: **ITEM 93**, **ITEM 5**, …, **ITEM 27** (-> **??, ??, ??**)
  USER 8: **ITEM 55**, **ITEM 24**, …, **ITEM 5** (-> **??, ??, ??**)
  ...
  USER 65: **ITEM 47**, **ITEM 7**, …, **ITEM 62** (-> **??, ??, ??**)

- **Tools**:
  - Encoder(users)-Decoder(items) framework: next item recommendation is based on both user and previous items.
  - LSTM to model ‘user encoding’ and ‘item transition’.
  - Embedding layer to incorporate feature learning.

Implementation



Important model designs

- **Features**
  - Continuous embedding is used to learn categorical features.
  - New layer (look-up table and concatenation) is used connect input and RNN cells.
- **Anonymous users**
  - *Item IDs* are treated as categorical features.
  - *User IDs are removed* to prevent overfitting.
- **Sampling and data augmentation**
  - *No sampling*.
  - Original sequence gives better empirical results.

Experiments

Settings:

- 26 to 44 week as training data. 45 as validation.
- Validations are reported.
  - Submitting quota limit
  - Consistent validation/test scores

Evaluation metric:

- Score (all): The challenge score.
- Score(new): The score after *removing all user-item pair in the history*.

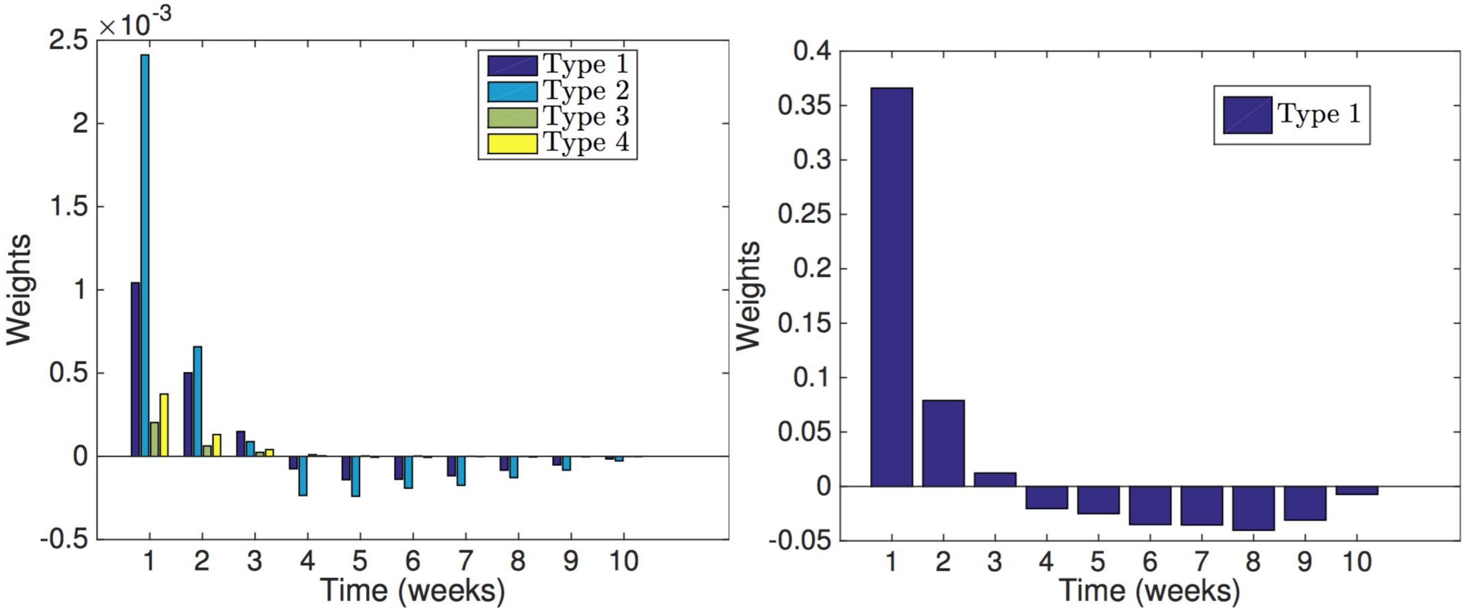
Recommend from history

Scores (in thousands) only based on historical items.

Models	Rand	TSort	TRank
INTS	266	284	**299**
IMPS	324	375	**380**
INTS+IMPS	463	509	**524**

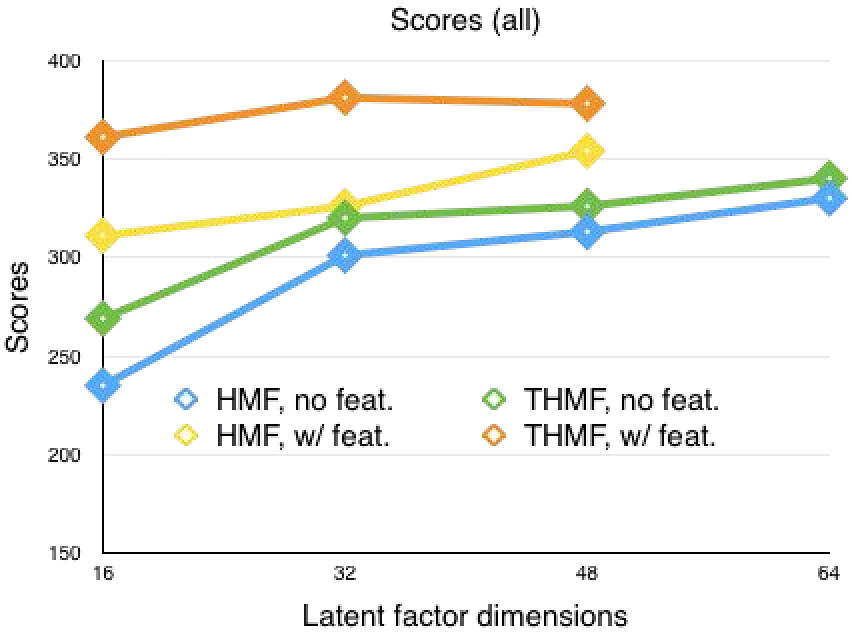
(*The higher, the better.*)

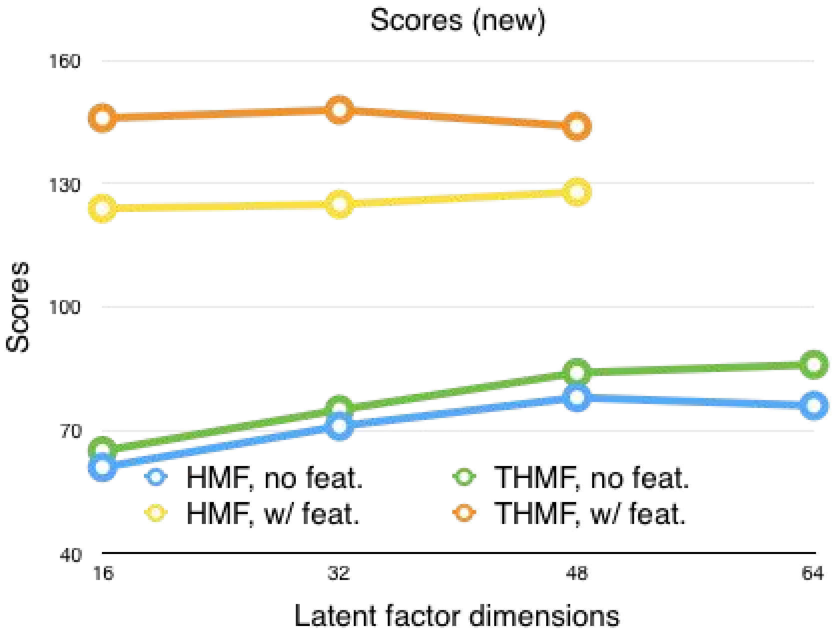
Weights associated with time/interaction types.



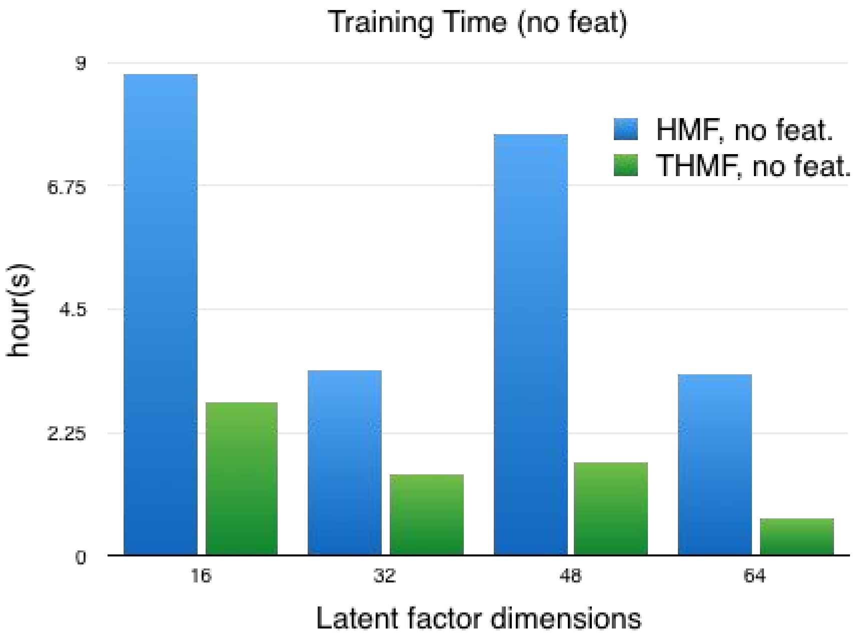
(a) interactions (b) impressions

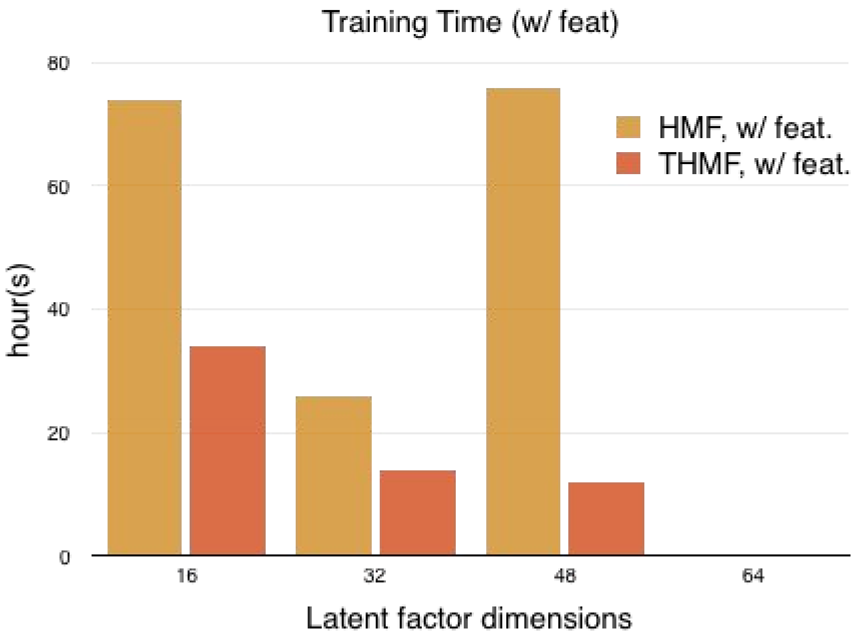
Temporal HMF Improves HMF





THMF Reduces Training Time

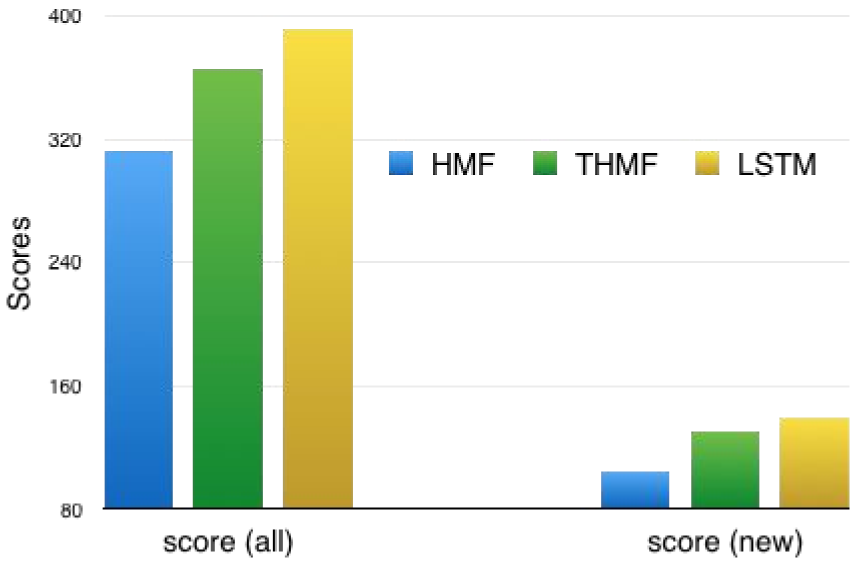




Recommend via LSTMs

Performance comparison.

- HMF
- THMF
- LSTM



(*The higher, the better.*)

Does sequence help?

Implicit assumption: sequence or order provides additional information beyond that provided by item frequency alone.

Experiment:

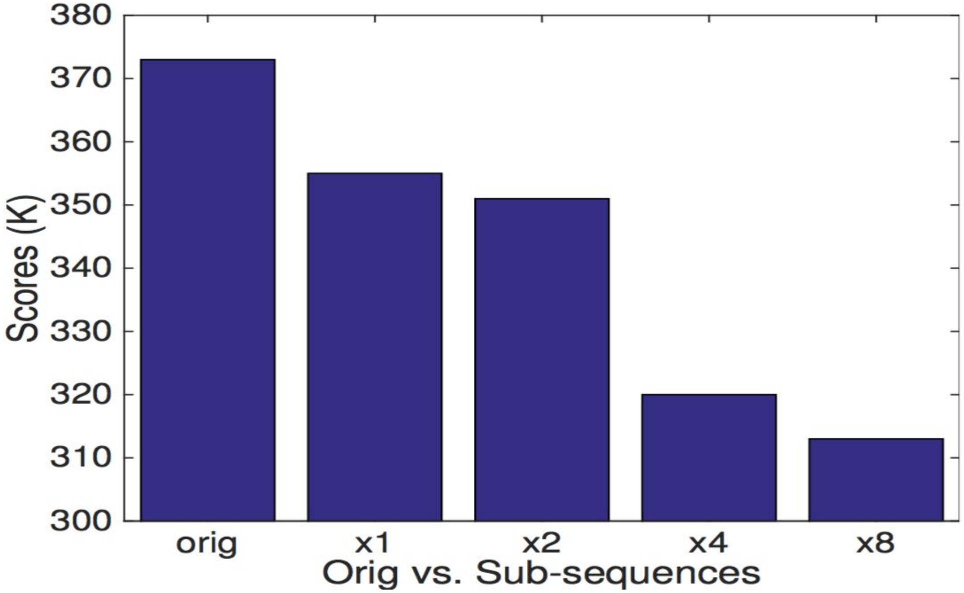
- Original sequence.
- Sub-sequence sampling.

Does sequence help?

Implicit assumption: sequence or order provides additional information beyond that provided by item frequency alone.

Experiment:

- Original sequence.
- Sub-sequence sampling.



Conclusion

Our empirical study verifies the effectiveness of

1) utilizing historical information in predicting users’ preferences
2) both temporal and sequence modeling in improving recommendation

Notably, the proposed RNN-based model outperforms the commonly used matrix factorization models.

Future research includes RNN model designs (e.g. to incorporate feature learning in the output layer) and analysis why and when sequence modeling helps recommendation.

Q & A

Thanks you!

Other slides

Outline

- RecSys Challenge 2016
- Approach Overview
- Temporal Learning
- Sequence modeling
- Experiments
- Conclude

Recommend via MF

Models		HMF			THMF		
Fea	d	score$_{all}$	score$_{new}$	T	score$_{all}$	score$_{new}$	T
No	16	235	61	8.8	269	65	2.8
	32	301	71	3.4	320	75	1.5
	48	313	78	7.7	326	84	1.7
	64	330	76	3.3	340	86	0.7
Yes	16	311	124	74	361	146	34
	32	326	125	26	**381**	**148**	14
	48	354	128	76	378	144	12

Recommend via LSTMs

Fea	No			Yes		
Models	HMF	THMF	LSTM	HMF	THMF	LSTM
score_{all}	313	**347**	313	312	366	**391**
score_{new}	78	87	**89**	104	130	**140**

Final score before/after ensemble

Component	History	MF (ints+imps)	LSTMs	Ensemble
Valid	524	438	391	613
Test	502	441	384	615