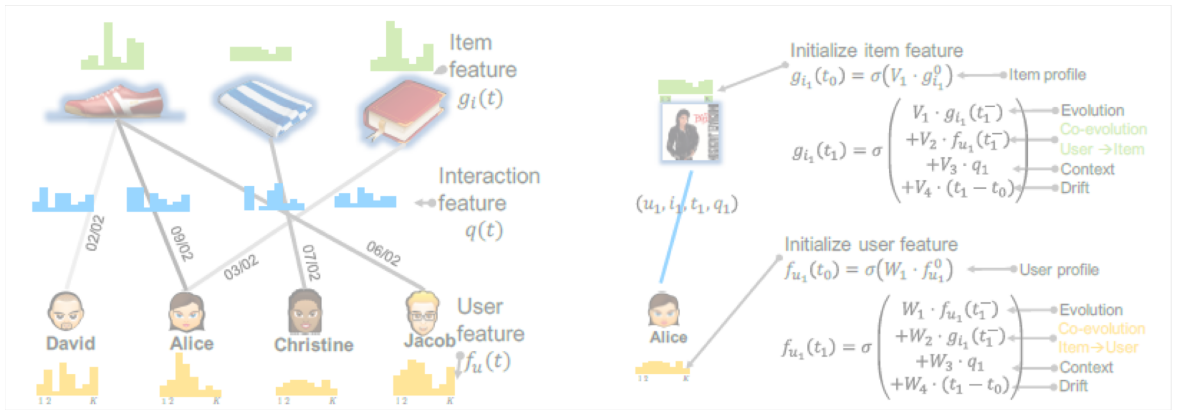
Deep Coevolutionary Network
: Embedding User and Item Features for Recommendation

Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song

ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)`17


Item feature $g_i(t)$
Interaction feature $q(t)$
User feature $f_u(t)$
02/02
09/02
03/02
07/02
06/02
David
Alice
Christine
Jacob
(u_1, i_1, t_1, q_1)
Alice
Initialize item feature
$g_{i_1}(t_0) = \sigma(V_1 \cdot g^0_{i_1})$
Item profile
$g_{i_1}(t_1) = \sigma\left(V_1 \cdot g_{i_1}(t_1^-) + V_2 \cdot f_{u_1}(t_1^-) + V_3 \cdot q_1 + V_4 \cdot (t_1 - t_0)\right)$
Evolution
Co-evolution User → Item
Context
Drift
Initialize user feature
$f_{u_1}(t_0) = \sigma(W_1 \cdot f^0_{u_1})$
User profile
$f_{u_1}(t_1) = \sigma\left(W_1 \cdot f_{u_1}(t_1^-) + W_2 \cdot g_{i_1}(t_1^-) + W_3 \cdot q_1 + W_4 \cdot (t_1 - t_0)\right)$
Evolution
Co-evolution Item→User
Context
Drift

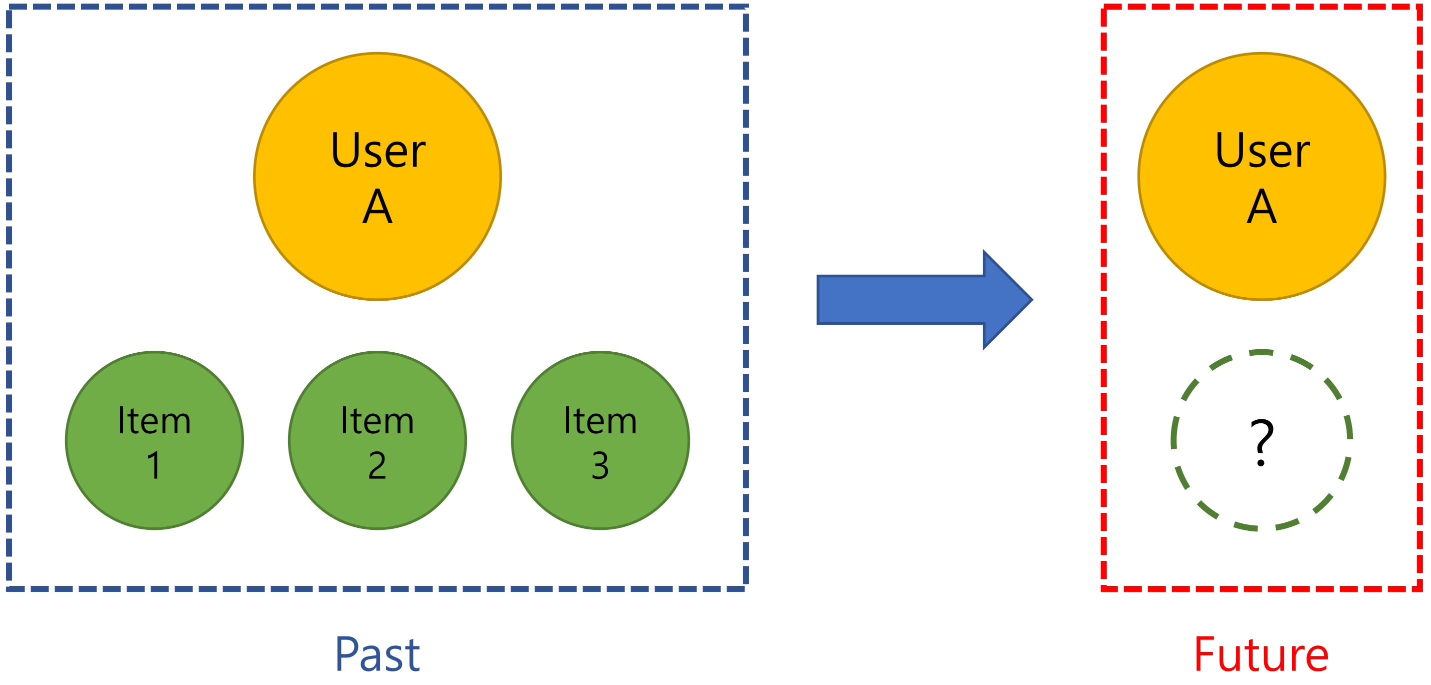
Hanyang University, Computer Science,
Computer Vision & Pattern Recognition,
Cho yong-chae (yccho@visionlab.or.kr)

2018.11.27

Contents

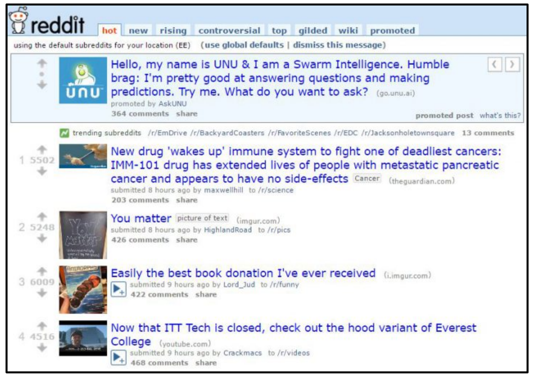
a. Problem definition

- User behavior prediction
  : predict which item will be selected



a. Problem definition

- Use three datasets



reddit hot new rising controversial top gilded wiki promoted

using the default subreddits for your location (EE) (use global defaults | dismiss this message)

Hello, my name is UNU & I am a Swarm Intelligence. Humble brag: I'm pretty good at answering questions and making predictions. Try me. What do you want to ask? (go.unu.ai)
promoted by AskUNU
364 comments share
promoted post what's this?

trending subreddits /r/EmDrive /r/BackyardCoasters /r/FavoriteScenes /r/EDC /r/Jacksonholetownsquare 13 comments

1 5502 New drug 'wakes up' immune system to fight one of deadliest cancers: IMM-101 drug has extended lives of people with metastatic pancreatic cancer and appears to have no side-effects Cancer (theguardian.com)
submitted 8 hours ago by maxwellhill to /r/science
203 comments share

2 5248 You matter picture of text (imgur.com)
submitted 8 hours ago by HighlandRoad to /r/pics
426 comments share

3 6009 Easily the best book donation I've ever received (i.imgur.com)
submitted 9 hours ago by Lord_Jud to /r/funny
422 comments share

4 4516 Now that ITT Tech is closed, check out the hood variant of Everest College (youtube.com)
submitted 9 hours ago by Crackmacs to /r/videos
468 comments share

IPTV
user & TV programs

Yelp
user & locations

Reddit
user & group (topic)

a. Problem definition

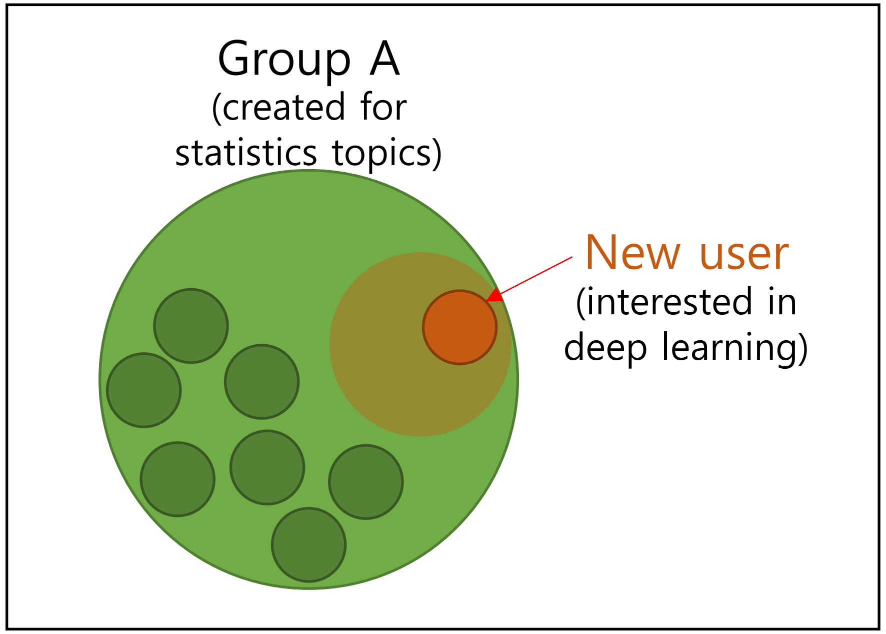
- Additionally, predict **when** item will be selected
- Making proper recommendation of items to users at the right time
  → fundamental task in e-commerce platforms

IPTV
user & TV programs

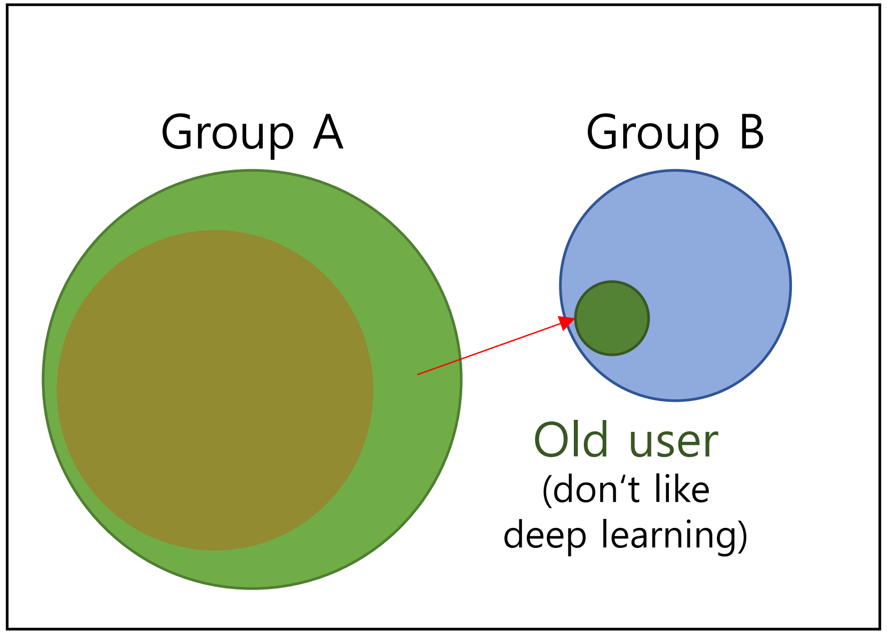
“**When** will the user watch this program?”

b. Coevolution between users and items

- Users interest and item's property is a good predictive factor.
- As users interact with different items, users' interests and items' features can also co-evolve over time.


Group A
(created for statistics topics)
New user
(interested in deep learning)

From user to item
(statistics topics → deep learning topics)



From item to user
(gain interests in other groups)

b. Coevolution between users and items

- Such coevolutionary nature of user and item features raises challenging questions:

  1. How to model coevolutionary features?
  2. How to efficiently train such models on large scale data?

  (Problem: user behavior prediction with time)

c. Previous methods

- **Tensor factorization**: divide time into epochs, and perform tensor factorization to learn the latent features
- Limitation: not able to capture the fine grained temporal dynamics of user-item interactions

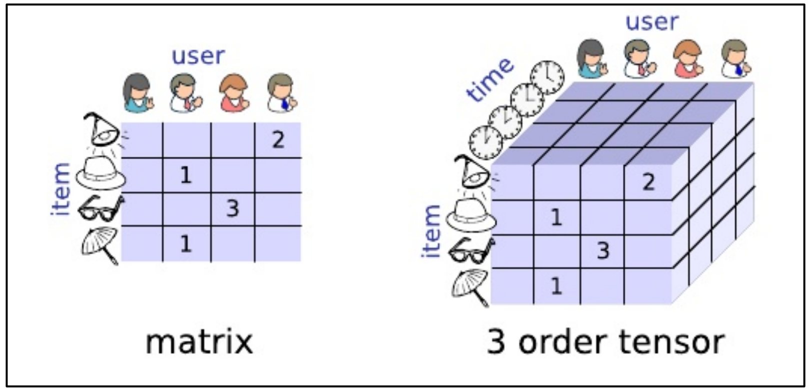


Fig. Tensor factorization
(ref. Kohei Hayashi, Post-doc researcher, https://www.slideshare.net/KoheiHayashi1/talk-in-jokyonokai-12989223)

c. Previous methods

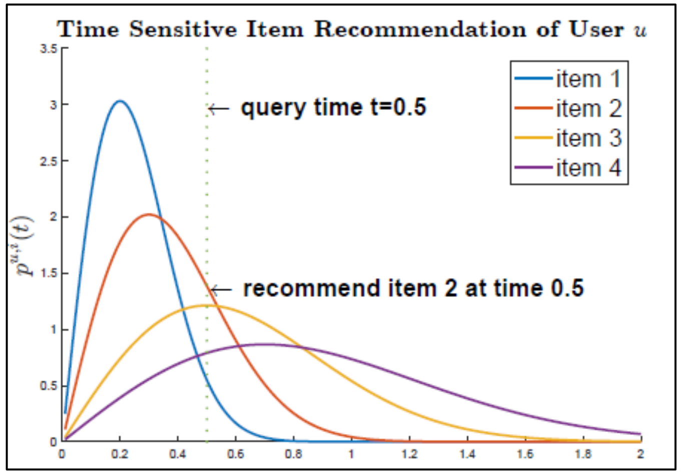
- **Temporal point process**: a stochastic model for the time of the next event given all the previous events
- Treat event times as random variables
- Limitation: make strong parametric assumptions about the functional form

$(e_0, t = -10),$
$(e_1, t = -9.8),$

…

$(e_n, t = -0.1)$

all the previous events



stochastic model for the time of the next event

d. Proposed method and contributions

- Proposed method
  1. Feature embedding using coevolution

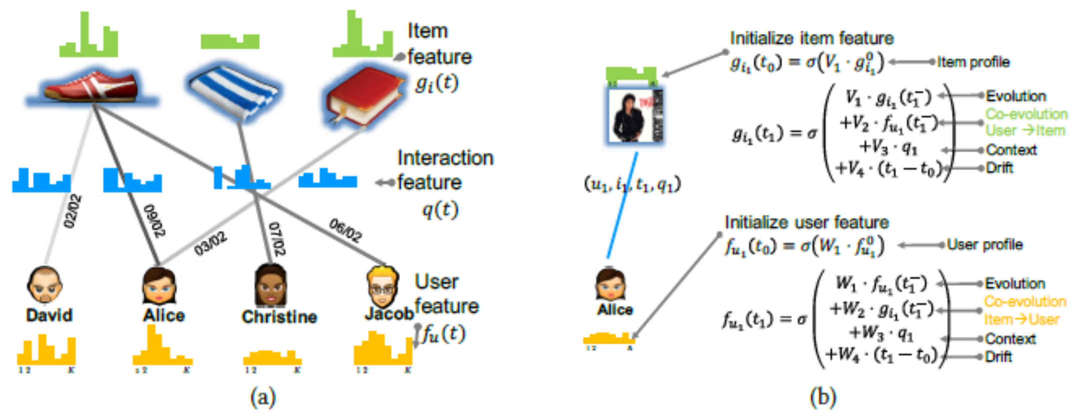

Item feature $g_i(t)$
Interaction feature $q(t)$
User feature $f_u(t)$
02/02
09/02
03/02
07/02
06/02
David
Alice
Christine
Jacob
(a)
Initialize item feature
$g_{i_1}(t_0) = \sigma(V_1 \cdot g^0_{i_1})$
Item profile
$g_{i_1}(t_1) = \sigma\begin{pmatrix} V_1 \cdot g_{i_1}(t_1^-) \\ +V_2 \cdot f_{u_1}(t_1^-) \\ +V_3 \cdot q_1 \\ +V_4 \cdot (t_1 - t_0) \end{pmatrix}$
Evolution
Co-evolution User →Item
Context
Drift
(u_1, i_1, t_1, q_1)
Alice
Initialize user feature
$f_{u_1}(t_0) = \sigma(W_1 \cdot f^0_{u_1})$
User profile
$f_{u_1}(t_1) = \sigma\begin{pmatrix} W_1 \cdot f_{u_1}(t_1^-) \\ +W_2 \cdot g_{i_1}(t_1^-) \\ +W_3 \cdot q_1 \\ +W_4 \cdot (t_1 - t_0) \end{pmatrix}$
Evolution
Co-evolution Item→User
Context
Drift
(b)

Figure 1: Model illustration. (a) User-item interaction as evolving bipartite graph. Each edge stands for a tuple and contains the information of user, item, interaction time, and interaction feature. (b) Deep coevolutionary feature embedding processes. The embeddings of users and items are updated at each event time, by a nonlinear activation function $\sigma(\cdot)$ and four terms: self evolution, co-evolution, context (interaction feature), and self drift.

  2. Use temporal point process for compatibility between embeddings

d. Proposed method and contributions

- Contributions
  1. **Novel model**
     - Latent feature process is modeled with two parallel component with RNN.
       (a user's latent feature is determined by the items
       which user interacted with)
  2. **Efficient Training**
     - Coevolution model makes the samples inter-dependent,
       which is significantly more challenging than other deep learning models.
     - We propose an efficient stochastic training algorithm.
  3. **Strong performance**
     - We evaluate our method over multiple datasets,
       verifying that our method leads to significant improvements
       compared to state-of-the-arts.

Contents

a. Notations

- We model the user-item interactions by a multi-dimensional temporal point process
- Ordered list of N observed events

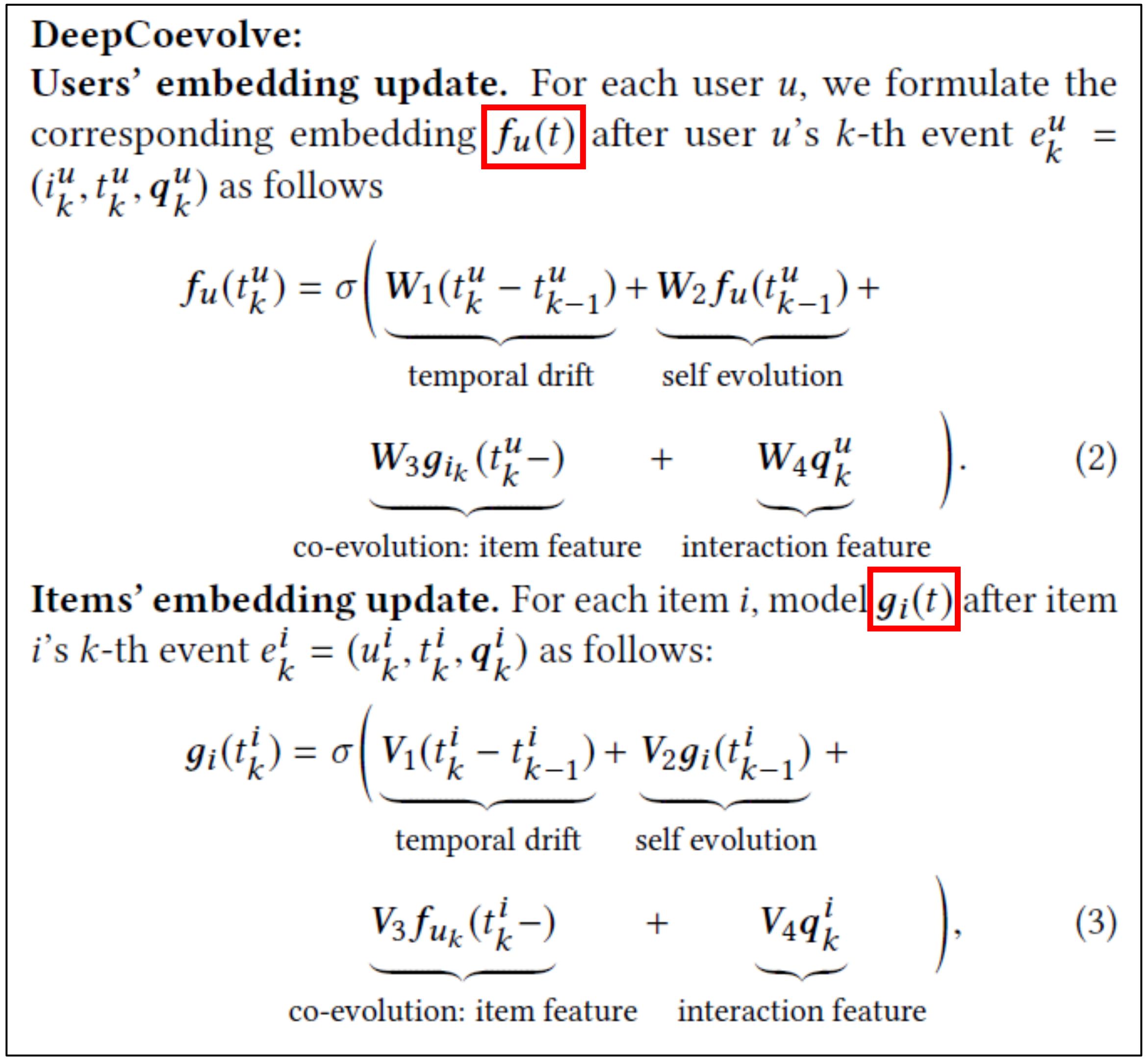
$$O = \{e_j = (\underbrace{u_j, i_j}_{①}, \underbrace{t_j}_{②}, \underbrace{q_j}_{③})\}_{j=1}^{N}$$

① User $u_j \in \{1, \dots, m\}$ and item $i_j \in \{1, \dots, n\}$
② Time t_j on the time window $[0, T]$
③ High dimension vector q_j,
such as the text review, or user's profile and item's category.

- Ordered list with user u and item i

$$O^u = \{e_j^u = (i_j, t_j, q_j)\}$$
$$O^i = \{e_j^i = (u_j, t_j, q_j)\}, \quad (t_0^i = t_0^u = 0)$$

b. Deep coevolutionary network



DeepCoevolve:

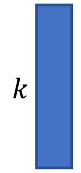
Users' embedding update. For each user u, we formulate the corresponding embedding $\boxed{f_u(t)}$ after user u's k-th event $e_k^u = (i_k^u, t_k^u, q_k^u)$ as follows

$$f_u(t_k^u) = \sigma\Big(\underbrace{W_1(t_k^u - t_{k-1}^u)}_{\text{temporal drift}} + \underbrace{W_2 f_u(t_{k-1}^u)}_{\text{self evolution}} + \underbrace{W_3 g_{i_k}(t_k^u-)}_{\text{co-evolution: item feature}} + \underbrace{W_4 q_k^u}_{\text{interaction feature}} \Big). \quad (2)$$

Items' embedding update. For each item i, model $\boxed{g_i(t)}$ after item i's k-th event $e_k^i = (u_k^i, t_k^i, q_k^i)$ as follows:

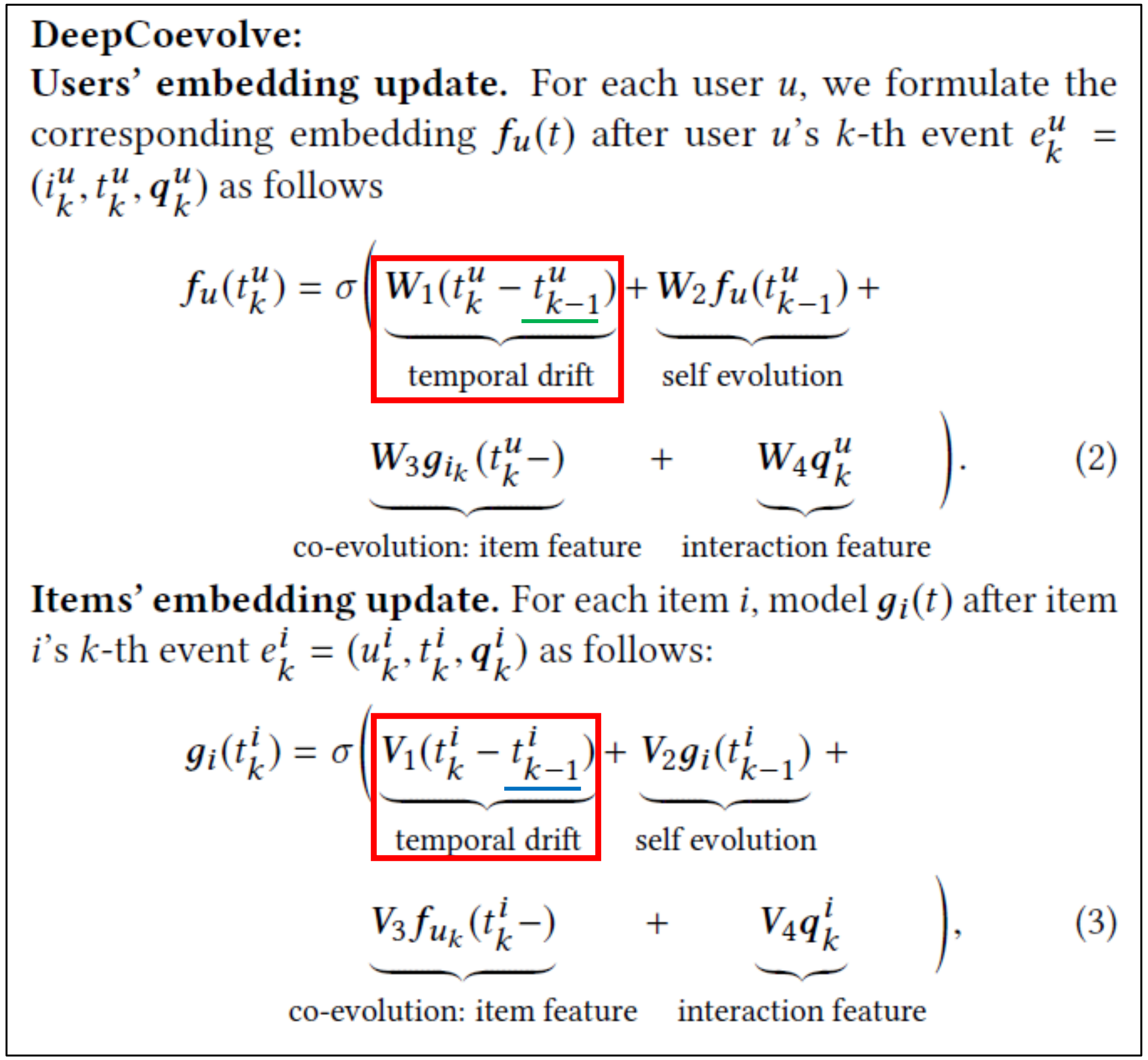
$$g_i(t_k^i) = \sigma\Big(\underbrace{V_1(t_k^i - t_{k-1}^i)}_{\text{temporal drift}} + \underbrace{V_2 g_i(t_{k-1}^i)}_{\text{self evolution}} + \underbrace{V_3 f_{u_k}(t_k^i-)}_{\text{co-evolution: item feature}} + \underbrace{V_4 q_k^i}_{\text{interaction feature}} \Big), \quad (3)$$

- $f_u(t), g_i(t)$
  - Embedding function over time
  - For each users and each items
  - Updated with four expressions
  - Output: $k \times 1$ dimension matrix



- Updating
  - Occurred at one event
    $e_j = (u_j, i_j, t_j, q_j),\qquad \text{for } j = 1 \text{ to } N$

b. Deep coevolutionary network



DeepCoevolve:

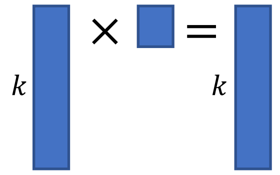
Users' embedding update. For each user u, we formulate the corresponding embedding $f_u(t)$ after user u's k-th event $e_k^u = (i_k^u, t_k^u, q_k^u)$ as follows

$$f_u(t_k^u) = \sigma\Big(\boxed{\underbrace{W_1(t_k^u - t_{k-1}^u)}_{\text{temporal drift}}} + \underbrace{W_2 f_u(t_{k-1}^u)}_{\text{self evolution}} + \underbrace{W_3 g_{i_k}(t_k^u-)}_{\text{co-evolution: item feature}} + \underbrace{W_4 q_k^u}_{\text{interaction feature}}\Big). \tag{2}$$

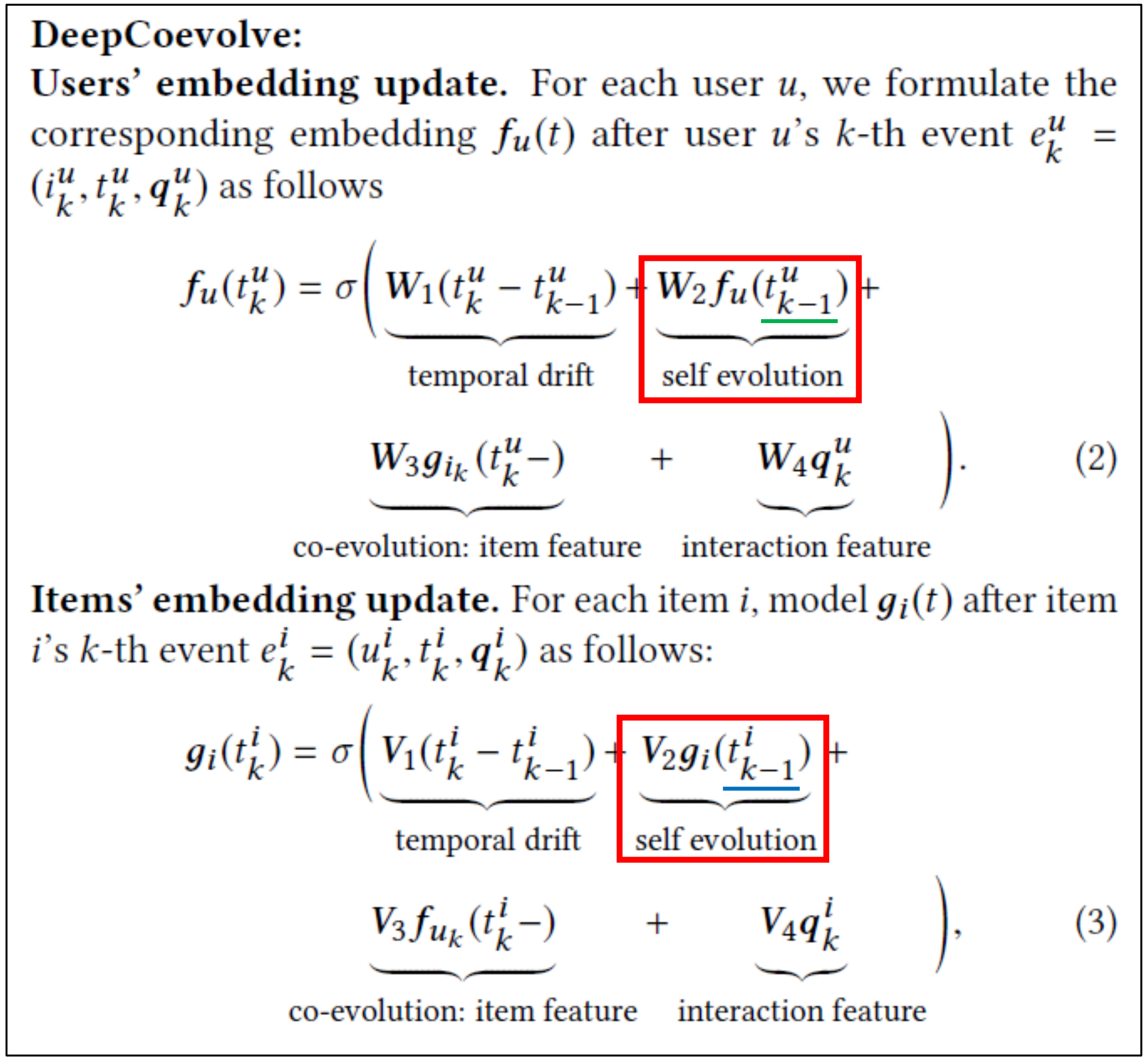
Items' embedding update. For each item i, model $g_i(t)$ after item i's k-th event $e_k^i = (u_k^i, t_k^i, q_k^i)$ as follows:

$$g_i(t_k^i) = \sigma\Big(\boxed{\underbrace{V_1(t_k^i - t_{k-1}^i)}_{\text{temporal drift}}} + \underbrace{V_2 g_i(t_{k-1}^i)}_{\text{self evolution}} + \underbrace{V_3 f_{u_k}(t_k^i-)}_{\text{co-evolution: item feature}} + \underbrace{V_4 q_k^i}_{\text{interaction feature}}\Big), \tag{3}$$

- Temporal drift
  - Time difference between consecutive events of specific user or item
  - t_{k-1}^u and t_{k-1}^i maybe different
  - t_{k-1}^u: time that user u select any item previously
  - t_{k-1}^i: time that item i is selected by any user previously
  - $W_1, V_1 \in R^k$ (k shape matrix), and time difference is scalar
    → Output: $k \times 1$ dimension matrix



b. Deep coevolutionary network



DeepCoevolve:

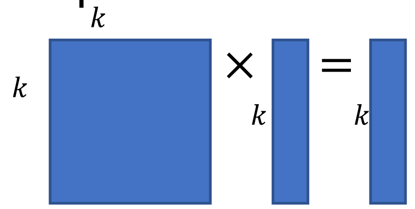
Users' embedding update. For each user u, we formulate the corresponding embedding $f_u(t)$ after user u's k-th event $e_k^u = (i_k^u, t_k^u, q_k^u)$ as follows

$$f_u(t_k^u) = \sigma\Big(\underbrace{W_1(t_k^u - t_{k-1}^u)}_{\text{temporal drift}} + \boxed{\underbrace{W_2 f_u(t_{k-1}^u)}_{\text{self evolution}}} + \underbrace{W_3 g_{i_k}(t_k^u-)}_{\text{co-evolution: item feature}} + \underbrace{W_4 q_k^u}_{\text{interaction feature}}\Big). \quad (2)$$

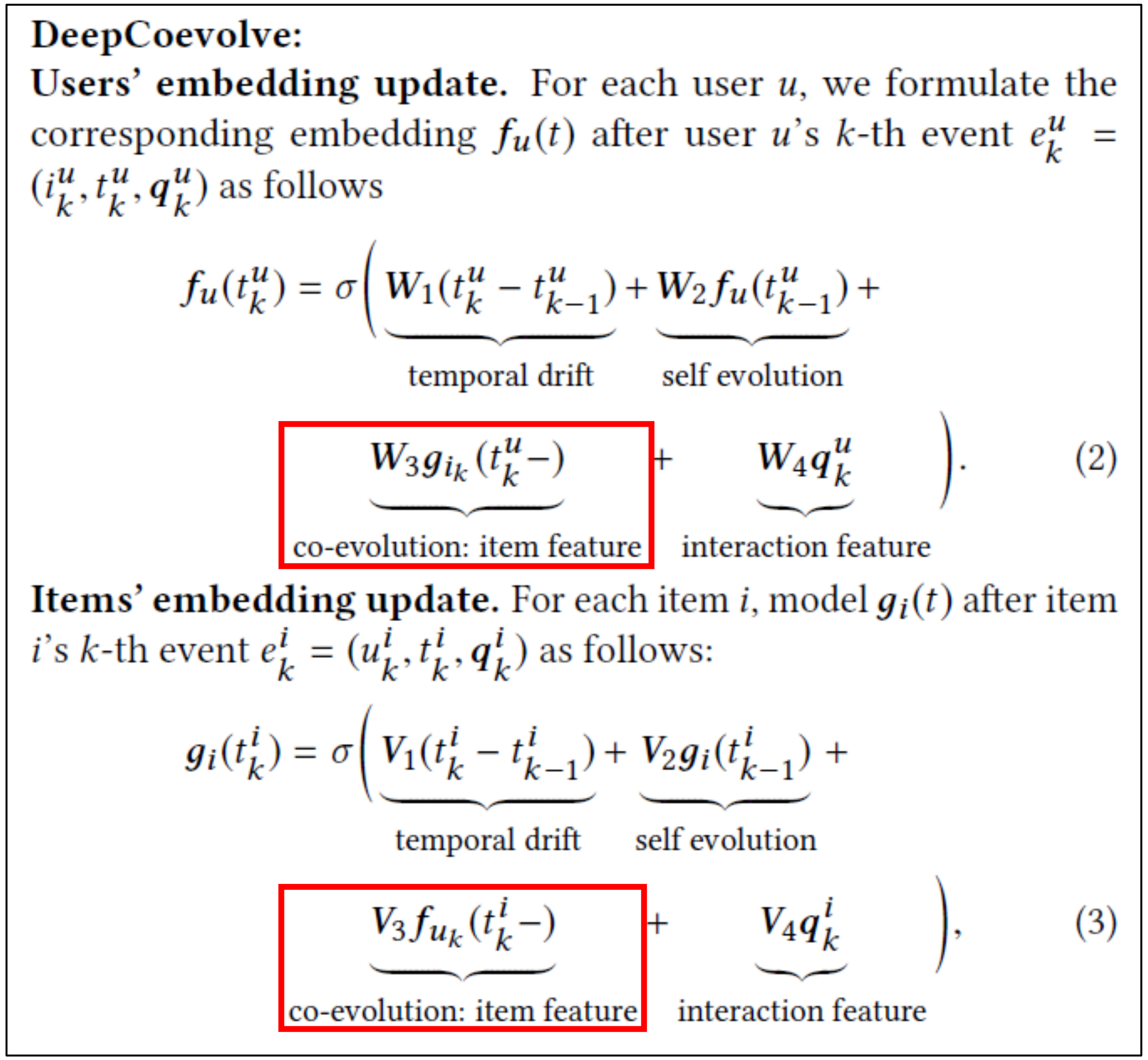
Items' embedding update. For each item i, model $g_i(t)$ after item i's k-th event $e_k^i = (u_k^i, t_k^i, q_k^i)$ as follows:

$$g_i(t_k^i) = \sigma\Big(\underbrace{V_1(t_k^i - t_{k-1}^i)}_{\text{temporal drift}} + \boxed{\underbrace{V_2 g_i(t_{k-1}^i)}_{\text{self evolution}}} + \underbrace{V_3 f_{u_k}(t_k^i-)}_{\text{co-evolution: item feature}} + \underbrace{V_4 q_k^i}_{\text{interaction feature}}\Big), \quad (3)$$

- Self evolution
  - Influenced by its feature at the earlier time $(k-1)$
  - t_{k-1}^u and t_{k-1}^i maybe different
  - $W_2, V_2 \in R^{k\times k}$,
    and right side output is $k\times 1$
    → Output: $k\times 1$ dimension matrix



b. Deep coevolutionary network



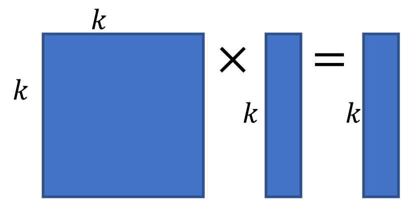
DeepCoevolve:
Users' embedding update. For each user u, we formulate the corresponding embedding $f_u(t)$ after user u's k-th event $e_k^u = (i_k^u, t_k^u, q_k^u)$ as follows

$$f_u(t_k^u) = \sigma\Big(\underbrace{W_1(t_k^u - t_{k-1}^u)}_{\text{temporal drift}} + \underbrace{W_2 f_u(t_{k-1}^u)}_{\text{self evolution}} + \boxed{\underbrace{W_3 g_{i_k}(t_k^u -)}_{\text{co-evolution: item feature}}} + \underbrace{W_4 q_k^u}_{\text{interaction feature}}\Big). \quad (2)$$

Items' embedding update. For each item i, model $g_i(t)$ after item i's k-th event $e_k^i = (u_k^i, t_k^i, q_k^i)$ as follows:

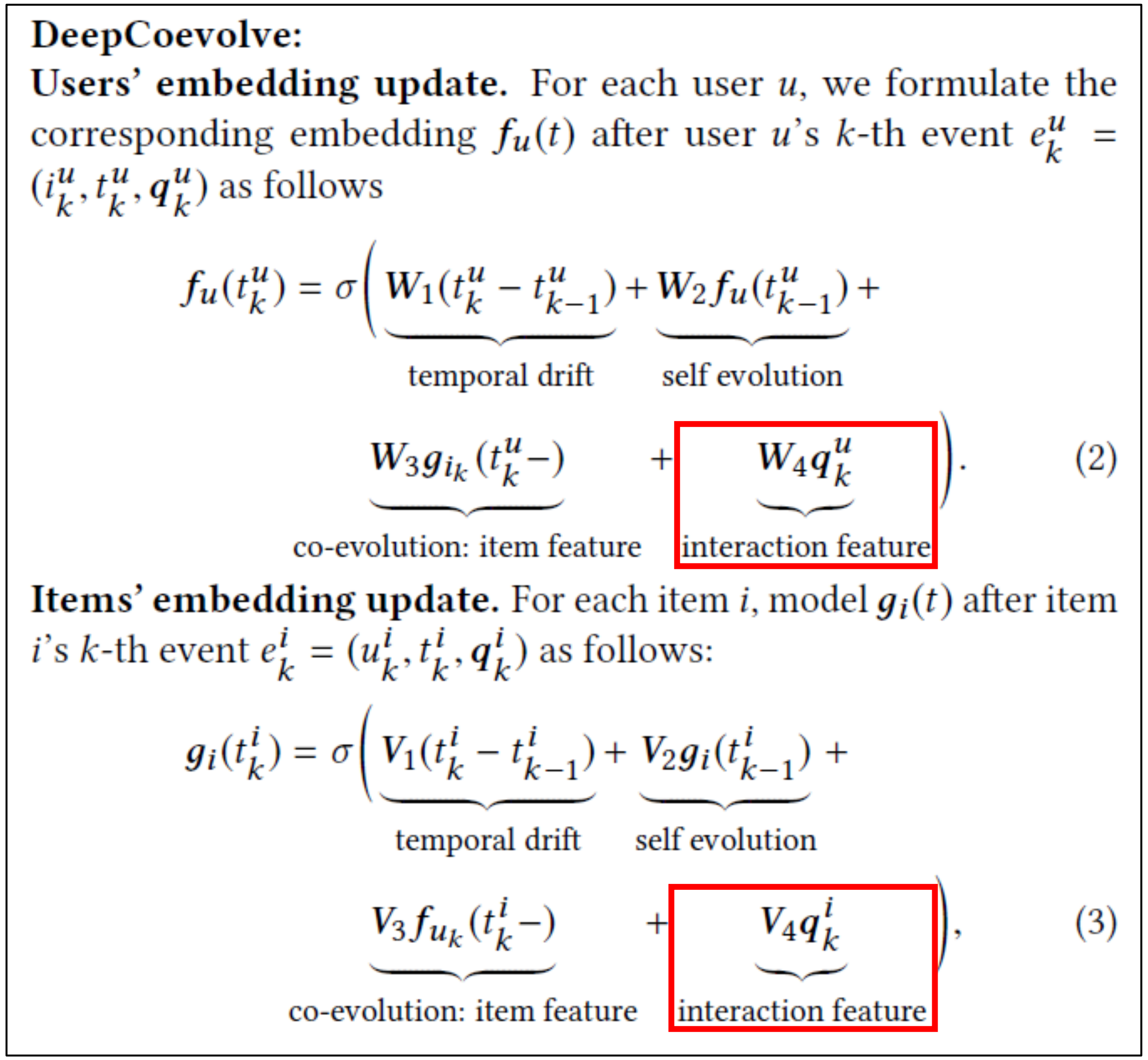
$$g_i(t_k^i) = \sigma\Big(\underbrace{V_1(t_k^i - t_{k-1}^i)}_{\text{temporal drift}} + \underbrace{V_2 g_i(t_{k-1}^i)}_{\text{self evolution}} + \boxed{\underbrace{V_3 f_{u_k}(t_k^i -)}_{\text{co-evolution: item feature}}} + \underbrace{V_4 q_k^i}_{\text{interaction feature}}\Big), \quad (3)$$

- Co-evolution
  - $t-$: timepoint just before t
  - A user's embedding is determined by the latent features of the items user interacted with.
  - Conversely, an item's embedding is determined by the feature embedding of the user who just interacts with the item.
  - $W_3, V_3 \in R^{k \times k}$,
    and right side output is k×1
    → Output: k×1 dimension matrix



2. Method

b. Deep coevolutionary network



DeepCoevolve:

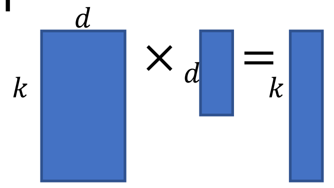
Users' embedding update. For each user u, we formulate the corresponding embedding $f_u(t)$ after user u's k-th event $e_k^u = (i_k^u, t_k^u, q_k^u)$ as follows

$$f_u(t_k^u) = \sigma\Big(\underbrace{W_1(t_k^u - t_{k-1}^u)}_{\text{temporal drift}} + \underbrace{W_2 f_u(t_{k-1}^u)}_{\text{self evolution}} + \underbrace{W_3 g_{i_k}(t_k^u-)}_{\text{co-evolution: item feature}} + \boxed{\underbrace{W_4 q_k^u}_{\text{interaction feature}}} \Big). \quad (2)$$

Items' embedding update. For each item i, model $g_i(t)$ after item i's k-th event $e_k^i = (u_k^i, t_k^i, q_k^i)$ as follows:

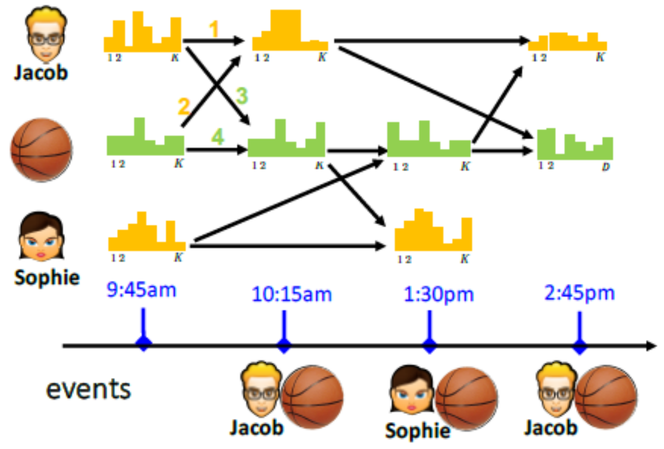
$$g_i(t_k^i) = \sigma\Big(\underbrace{V_1(t_k^i - t_{k-1}^i)}_{\text{temporal drift}} + \underbrace{V_2 g_i(t_{k-1}^i)}_{\text{self evolution}} + \underbrace{V_3 f_{u_k}(t_k^i-)}_{\text{co-evolution: item feature}} + \boxed{\underbrace{V_4 q_k^i}_{\text{interaction feature}}} \Big), \quad (3)$$

- Interaction feature
  - Additional information happened in the user-item interactions.
  - For example, in online discussion forums such as Reddit,
    the interaction features are the posts and comments.
  - The shape of feature is reduced to a d-dimension.
    (such as bag of words …)
  - $W_4, V_4 \in R^{k \times d}$,
    and right side output is $d \times 1$
    → Output: $k \times 1$ dimension matrix

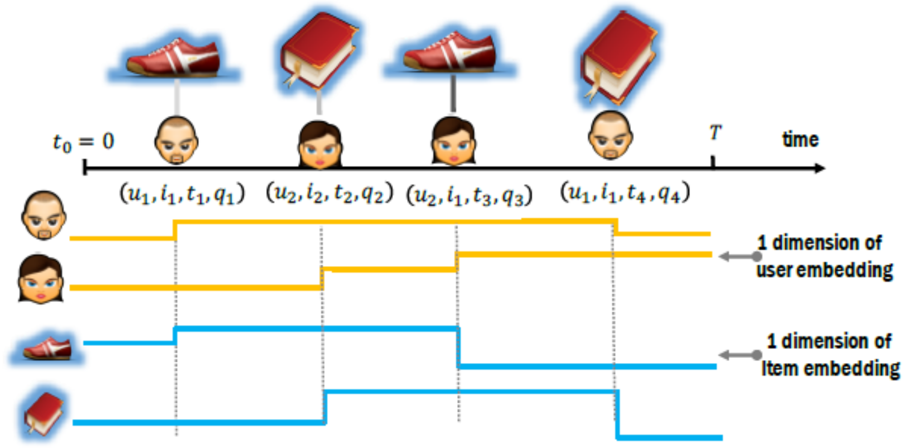


2. Method

b. Deep coevolutionary network



(a) Graph of embedding computation

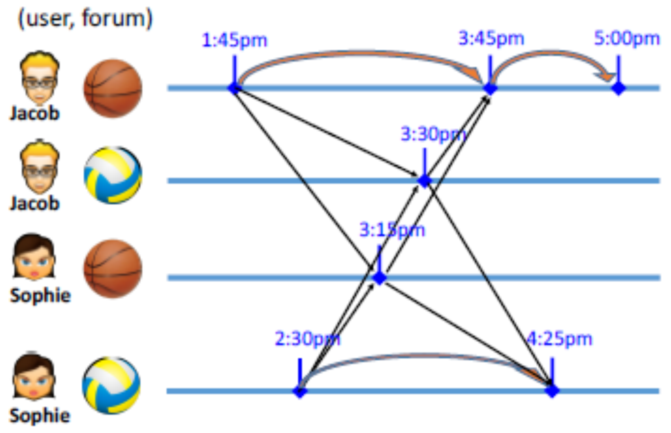


(b) Dependency between events

Figure 2: (a) The arrows indicate the dependency structures in the embedding updates, *e.g.*, Jacob interacts with basketball at 10:15am. Then the feature embeddings are updated: the new feature embedding at 10:15 am is influenced by his previous feature embedding and the basketball's previous feature embedding at 9:45am (arrow 1 and 2); the basketball's feature embedding is also influenced by Jacob's previous feature embedding and its previous embedding feature (arrow 3 and 4). (b) A user or item's feature embedding is piecewise constant over time and will change *only* after an interaction event happens. Only one dimension of the feature embedding is shown.

2. Method

b. Deep coevolutionary network



(a) Dependency between events

Figure 3: (a) The events dependency for two users and two forums (items). It shows how event at one dimension influence other dimensions. Each orange arrow represents the dependency within each dimension, and the black arrow denotes the cross-dimension dependency. For example, Sophie interacts with volleyball at 2:30pm, and this event changes the volleyball embedding, thus will affect Jacob's visit at 3:30pm.

c. Intensity function as the compatibility between embeddings

- We model the repeated occurrences of all users interaction with all items as a multi-dimensional temporal point process
- Each user-item pair is one dimension.
- Mathematically, we model the intensity function in the (u, i)-th dimension as a Rayleigh process:

$$\lambda^{u,i}(t|t') = \underbrace{\exp\left(f_u(t')^\top g_i(t')\right)}_{\text{user-item compatibility}} \cdot \underbrace{(t - t')}_{\text{time lapse}} \quad (4)$$

(t': last time point where either u's embedding or i's embedding changes before time t)

Contents

$\longrightarrow$

a. Objective function
b. Prediction
c. Complexity analysis

- Efficient learning algorithm

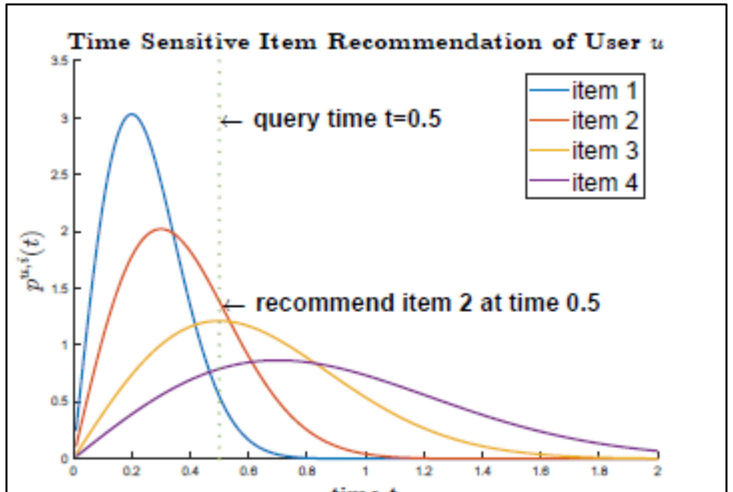
a. Objective function

$$\ell = \underbrace{-\sum_{j=1}^{N} \log\left(\lambda^{u_j,i_j}(t_j|t_j')\right)}_{\text{happened events}} + \underbrace{\sum_{u=1}^{m}\sum_{i=1}^{n}\int_0^T \lambda^{u,i}(\tau|\tau')\,\mathrm{d}\tau}_{\text{survival of not-happened events}} \quad (5)$$

- The negative intensity summation term ensures the probability of all interaction events is maximized.
- The second survival probability term penalizes the non-presence of an interaction between all possible user-item pairs on the observation window.
  → Can explain why an event did not happen.

b. Prediction

- Next item prediction
  : what is the item the user u will interact at time t?
  - At different point in time, a different prediction can be made.
  - $p^{u,i}(t) = \lambda^{u,i}(t) S^{u,i}(t)$



(c) Item recommendation

b. Prediction

- Time prediction
  : when this user will interact with this item in the future?
  - Expected next event time under $p^{u,i}(t)$
  - Since the intensity model is a Rayleigh model,
    the expected event time can be computed in closed form as

$$\mathbb{E}_{t\sim p^{u,i}(t)}[t] = \sqrt{\frac{\pi}{2\exp\left(f_u(t-)^\top g_i(t-)\right)}} \qquad (9)$$

c. Complexity analysis

- Model size
  - Basic feature embedding takes O(#user + #item) parameters.
  - Interaction features are independent of the number of users and items.
  - Parameters of RNN are also independent of the dataset.

  → More scalable than traditional matrix factorization methods.

- Training complexity
  - With M samples, O($(n+m)\times M$) operations are needed for forward.

  → Can be reduced to O(M) by mini-batch learning.

- Test/Prediction complexity
  - Comparing each item with the current user has a closed form.
    → O(N) for item prediction
  - Event time prediction is in closed form.
    → O(1) for time prediction

Contents

4. Experiments

a. Datasets

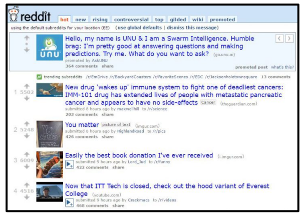
- IPTV

7,100 users / 436 TV programs in 11 months
/ with around 2M events

- Yelp

1,005 users / 47,924 businesses in 14 months
/ with 291,716 reviews

- Reddit



using the default subreddits for your location (EE) (use global defaults | dismiss this message)

Hello, my name is UNU & I am a Swarm Intelligence. Humble brag: I'm pretty good at answering questions and making predictions. Try me. What do you want to ask? (go.unu.ai)
promoted by AskUNU
364 comments share
promoted post what's this?

trending subreddits /r/EmDrive /r/BackyardCoasters /r/FavoriteScenes /r/EDC /r/Jacksonholetownsquare 13 comments

1 5502 New drug 'wakes up' immune system to fight one of deadliest cancers: IMM-101 drug has extended lives of people with metastatic pancreatic cancer and appears to have no side-effects Cancer (theguardian.com)
submitted 8 hours ago by maxwellhill to /r/science
203 comments share

2 5248 You matter picture of text (imgur.com)
submitted 8 hours ago by HighlandRoad to /r/pics
426 comments share

3 6009 Easily the best book donation I've ever received (i.imgur.com)
submitted 9 hours ago by Lord_Jud to /r/funny
422 comments share

4 4516 Now that ITT Tech is closed, check out the hood variant of Everest College (youtube.com)
submitted 9 hours ago by Crackmacs to /r/videos
468 comments share

1,000 users / 1,403 groups in 1 month
/ with 14,816 discussions

a. Datasets

- Sparsity in terms of the number of events per user
  - In IPTV dataset, users have longer length of history than other two datasets.

- Sparsity in terms of diversity of items to recommend
  - Yelp dataset has higher density than the other two datasets, hence higher diversity.
    → Yelp dataset is the most challenging one.

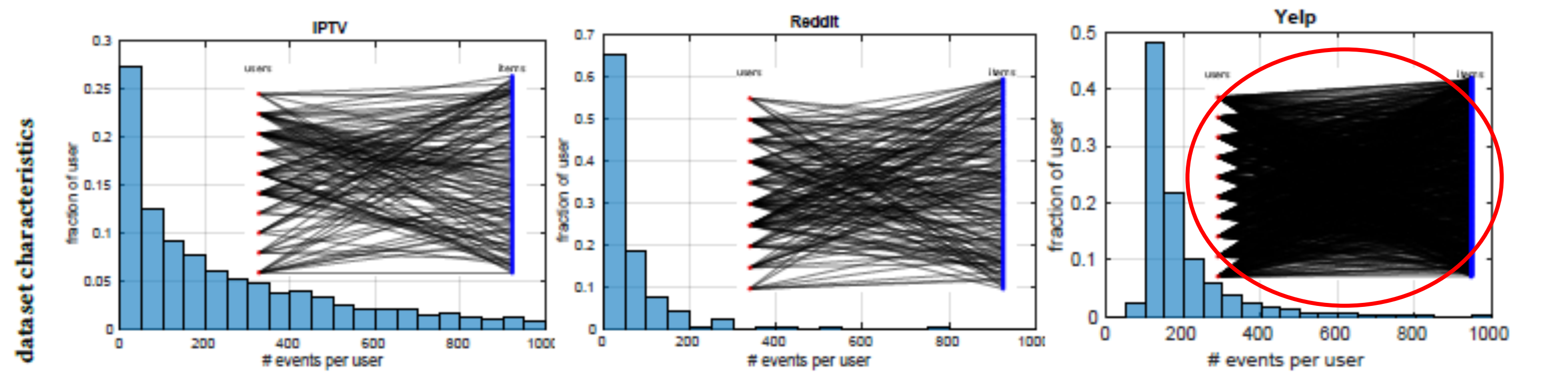


Figure 4: Visualization of the sparsity property in each dataset. The bar plot shows the distribution of number of events per user. Within each plot there is a user-item interaction graph. This graph is generated as follows. For each dataset, we randomly pick 10 users with 100 history events each user and collect all items they have interacted with. The interaction graph itself is a bipartite graph, and we put users on left side, and items on the right side.

b. Competitors

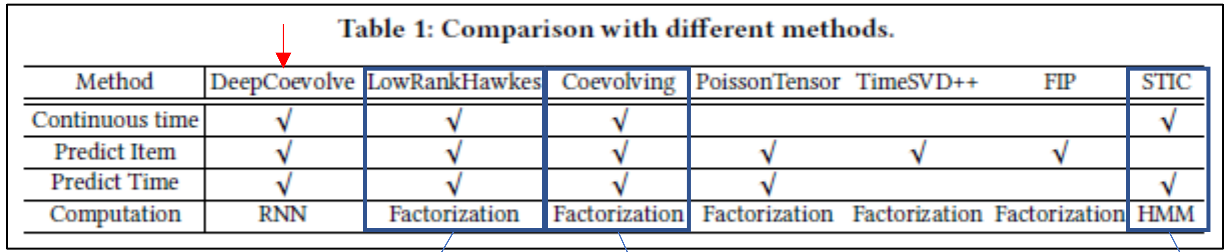


Table 1: Comparison with different methods.

Method	DeepCoevolve	LowRankHawkes	Coevolving	PoissonTensor	TimeSVD++	FIP	STIC
Continuous time	√	√	√				√
Predict Item	√	√	√	√	√	√	
Predict Time	√	√	√	√			√
Computation	RNN	Factorization	Factorization	Factorization	Factorization	Factorization	HMM

Point process based,
simple linear embedding

Markov model based

Point process based,
Assumes user-item interaction to be independent

c. Overall performance

- Use p ratio of events to training data, and others as the testing data
- Average results over five runs ($p \in \{0.7, 0.72, 0.74, 0.76, 0.78\}$)
- Use two metric
  - Item prediction metric: Mean Average Rank (MAR),
    lower is better, and value one means best.
  - Time prediction metric: Mean Absolute Error (MAE),
    lower is better.

4. Experiments

c. Overall performance comparison

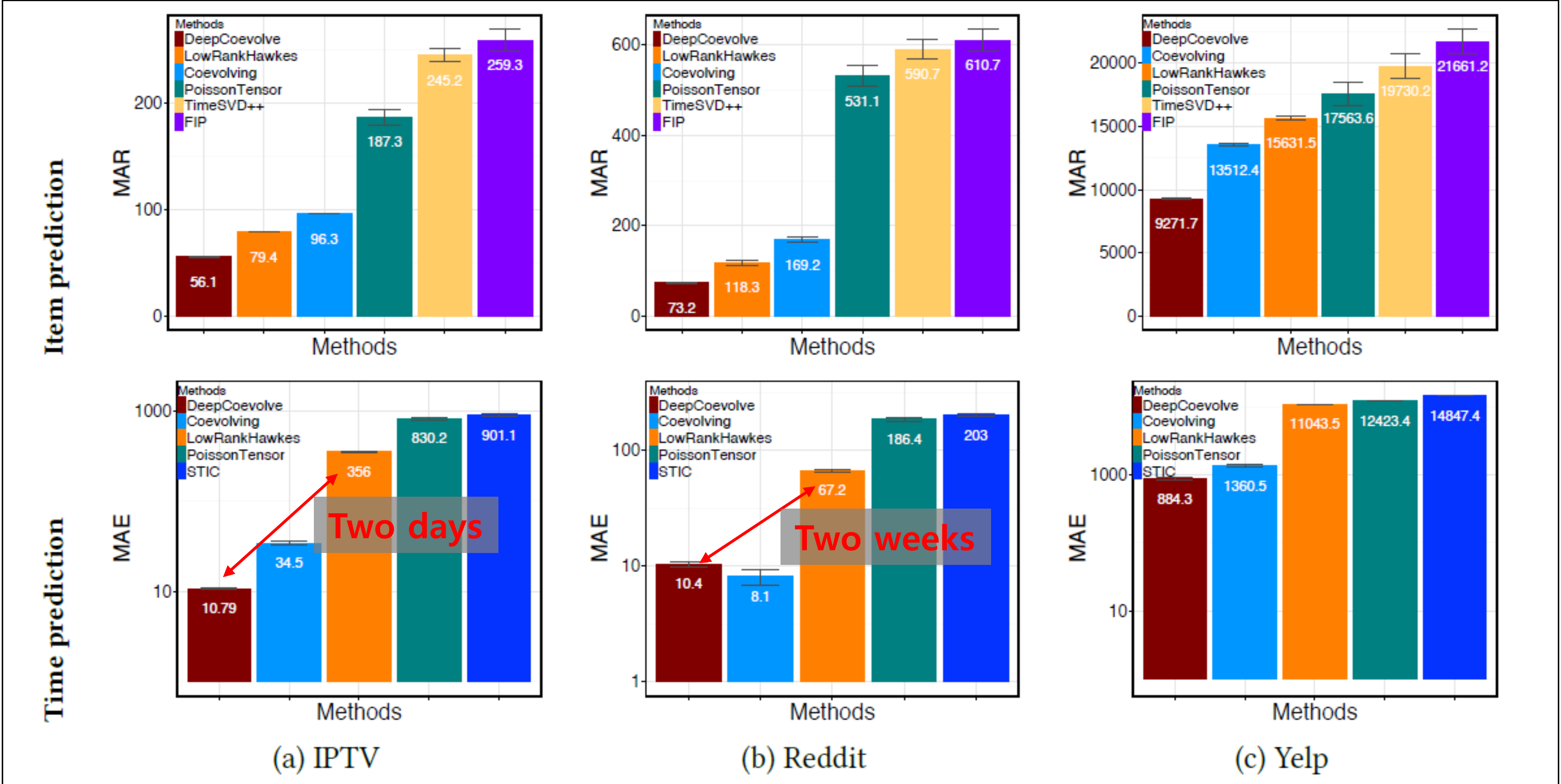


Figure 5: Prediction results on three real world datasets. The MAE for time prediction is measured in hours.

4. Experiments

d. Refined performance comparison

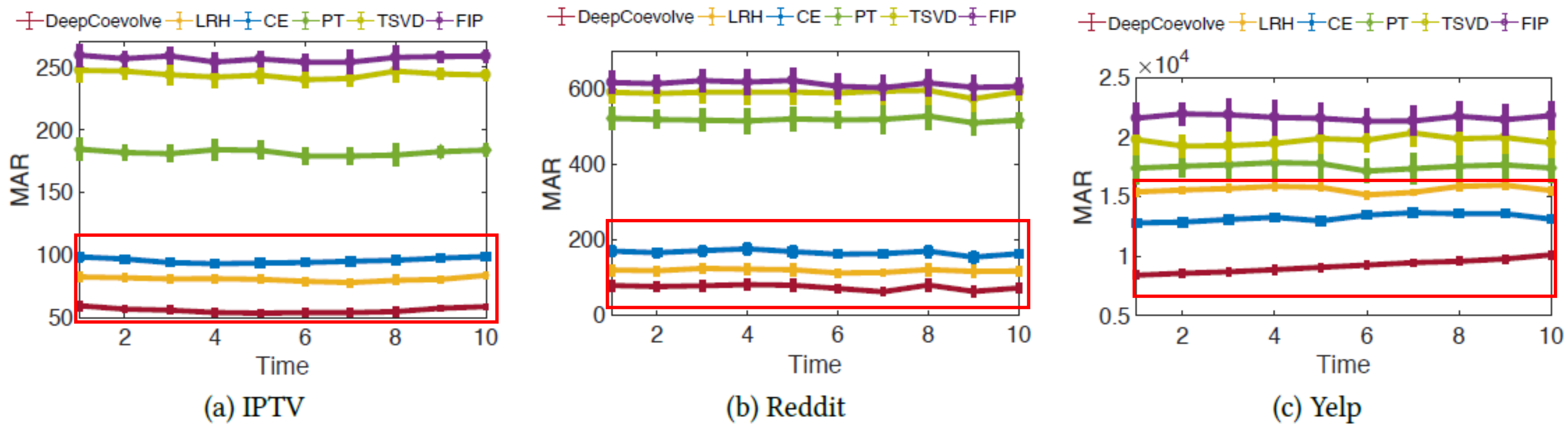


Figure 6: Item prediction over different time windows. We divide the entire test set into 10 bins, and report MAR in each bin.

- Comparison in different time windows
  - Deep-Coevolve is consistently better in different time periods.
  - Moreover, all the point process based methods have stable performance with small variance. (LRH: LowRankHawkes, CE: Coevolving, and our method)

4. Experiments

d. Refined performance comparison

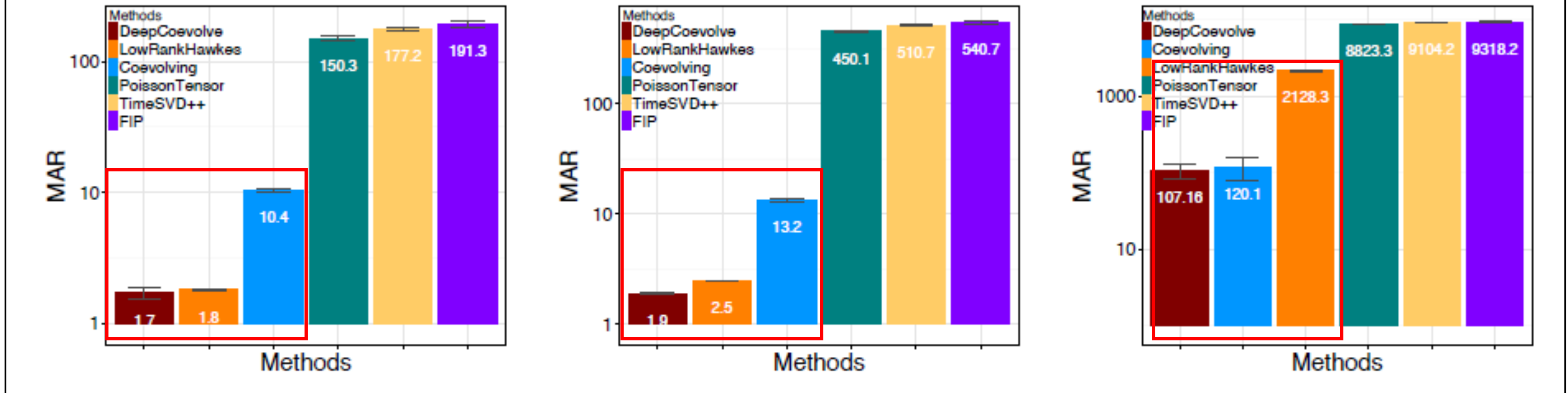


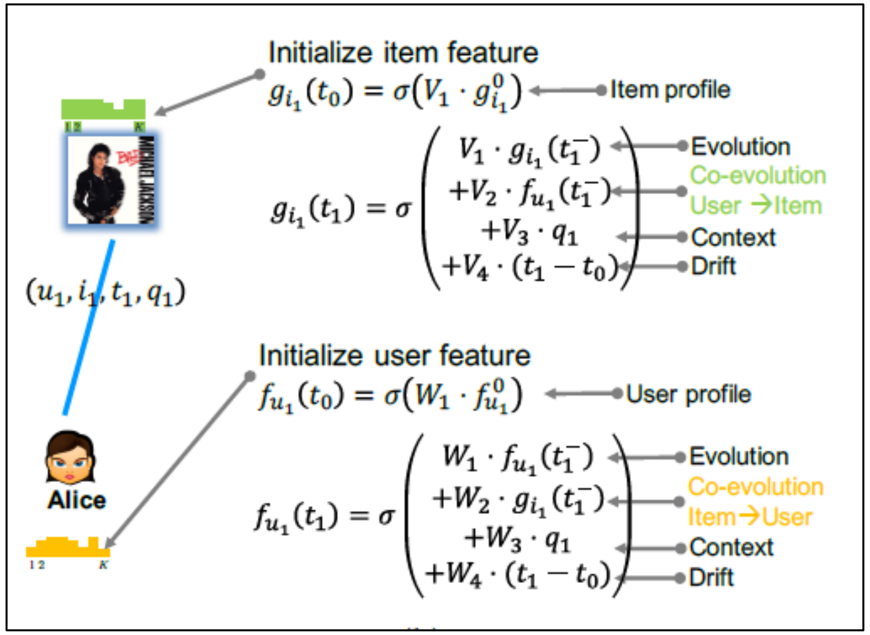
Figure 7: Item prediction performance on recurring events.

- Performance on recurring events
  - It is also important to understand whether/when your customers will com back again.
  - Our method wins others.
  - Moreover, point process based methods benefit more from predicting recurring events.

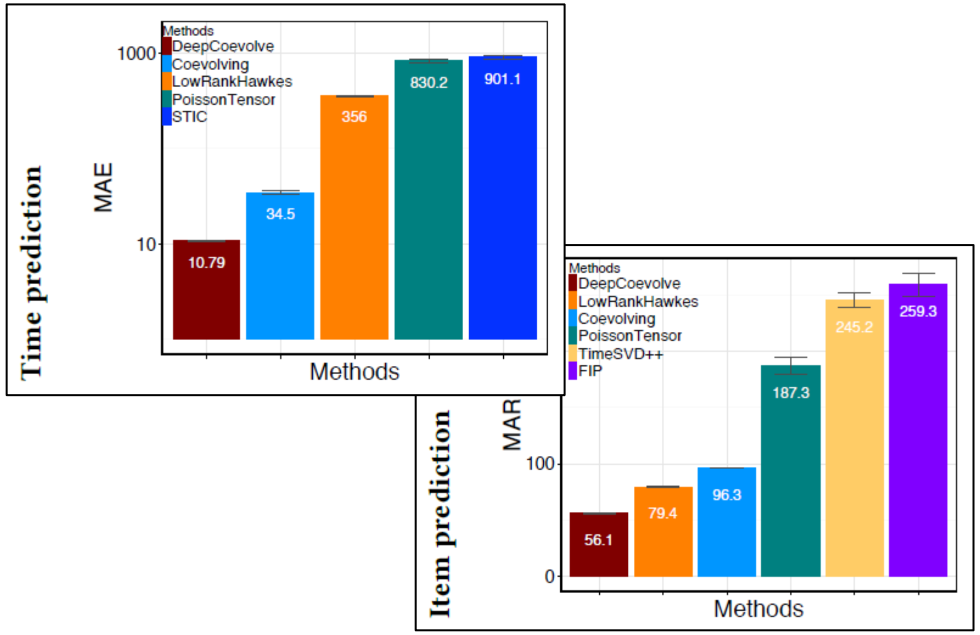
Contents

5. Conclusion

- We model coevolution nature of users' and items' embeddings.
- The user and item's evolving and coevolving processes are captured by the RNN.


Initialize item feature
$g_{i_1}(t_0) = \sigma(V_1 \cdot g_{i_1}^0)$
Item profile
$g_{i_1}(t_1) = \sigma\begin{pmatrix} V_1 \cdot g_{i_1}(t_1^-) \\ +V_2 \cdot f_{u_1}(t_1^-) \\ +V_3 \cdot q_1 \\ +V_4 \cdot (t_1 - t_0) \end{pmatrix}$
Evolution
Co-evolution
User →Item
Context
Drift
(u_1, i_1, t_1, q_1)
Initialize user feature
$f_{u_1}(t_0) = \sigma(W_1 \cdot f_{u_1}^0)$
User profile
$f_{u_1}(t_1) = \sigma\begin{pmatrix} W_1 \cdot f_{u_1}(t_1^-) \\ +W_2 \cdot g_{i_1}(t_1^-) \\ +W_3 \cdot q_1 \\ +W_4 \cdot (t_1 - t_0) \end{pmatrix}$
Evolution
Co-evolution
Item→User
Context
Drift
Alice

- We demonstrate the superior performance of our method on both the time and item prediction task.



Q & A

Thanks for listening!