Representation Learning of Tangled Key-Value Sequence Data for Early Classification

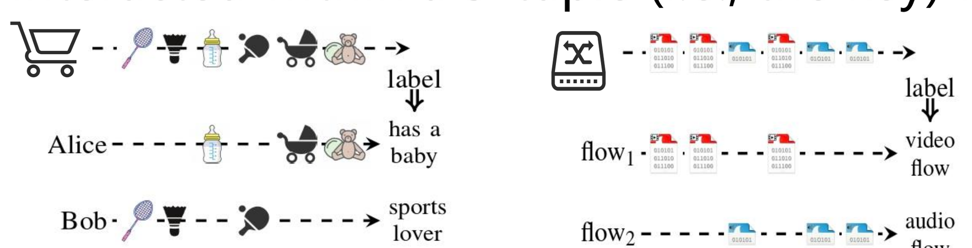
Tao Duan, Junzhou Zhao, Shuo Zhang, Jing Tao, Pinghui Wang

MOE KLINNS Lab, Xi'an Jiaotong University, China

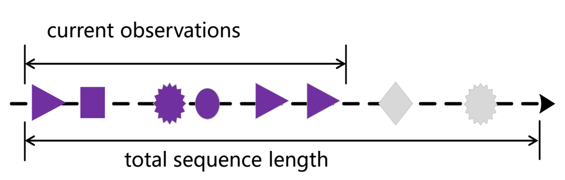
duantao@stu.xjtu.edu.cn

Background & Problem Formulation

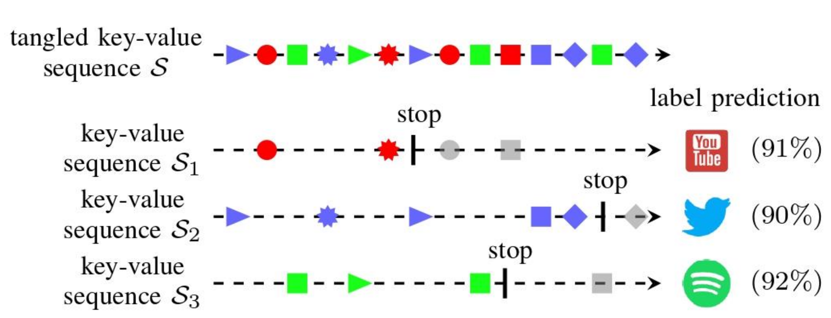
- Key-Value Sequence: temporal sequence of key-value pairs.
  - in user-product sequence, a key-value pair represents a user (i.e., the key) purchasing a product (i.e., the value).
  - in packet sequence, a key-value pair represents a packet (i.e., the value) interacted within a 5-tuple (i.e., the key) flow.



- Tangled Key-Value Sequence: concurrent key-value sequences with different keys.
- Applications:
  - Product Recommendation;
  - Networking QoS Improvement;
  - Malicious Intrusion Detection.



- Two core performances for sequence data classification: besides the requirement of classifying a key-value sequence accurately, it is also desired to classify it early, in order to respond fast.
- However, these two goals are conflicting in nature, and it is challenging to achieve them simultaneously.
- Problem Formulation:



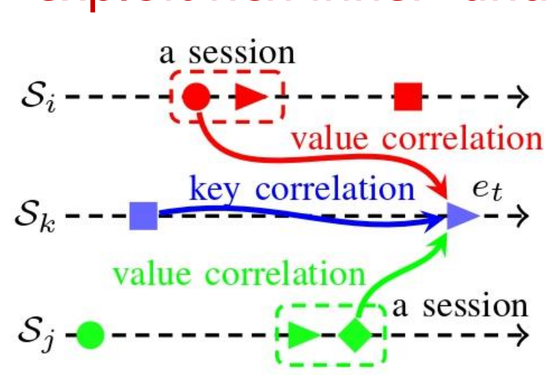
  - Given a tangled key-value sequence:

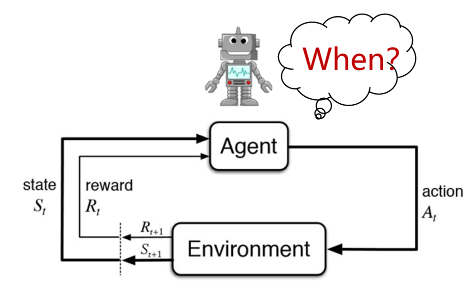
$$\mathcal{S} \triangleq (\langle k, \mathbf{v}\rangle : k \in \mathcal{K}, \mathbf{v} \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_l)$$

  - Classify each key-value sequence sharing a same key $\mathcal{S}_k \triangleq (\langle k, \mathbf{v}\rangle : \langle k, \mathbf{v}\rangle \in \mathcal{S})$ both early and accurately.

Motivations

- Decompose it to two targets:
  - how to learn an informative representation from partial observations of ongoing key-value sequence?
    - ✓ exploit rich inner- and inter-sequence correlations;

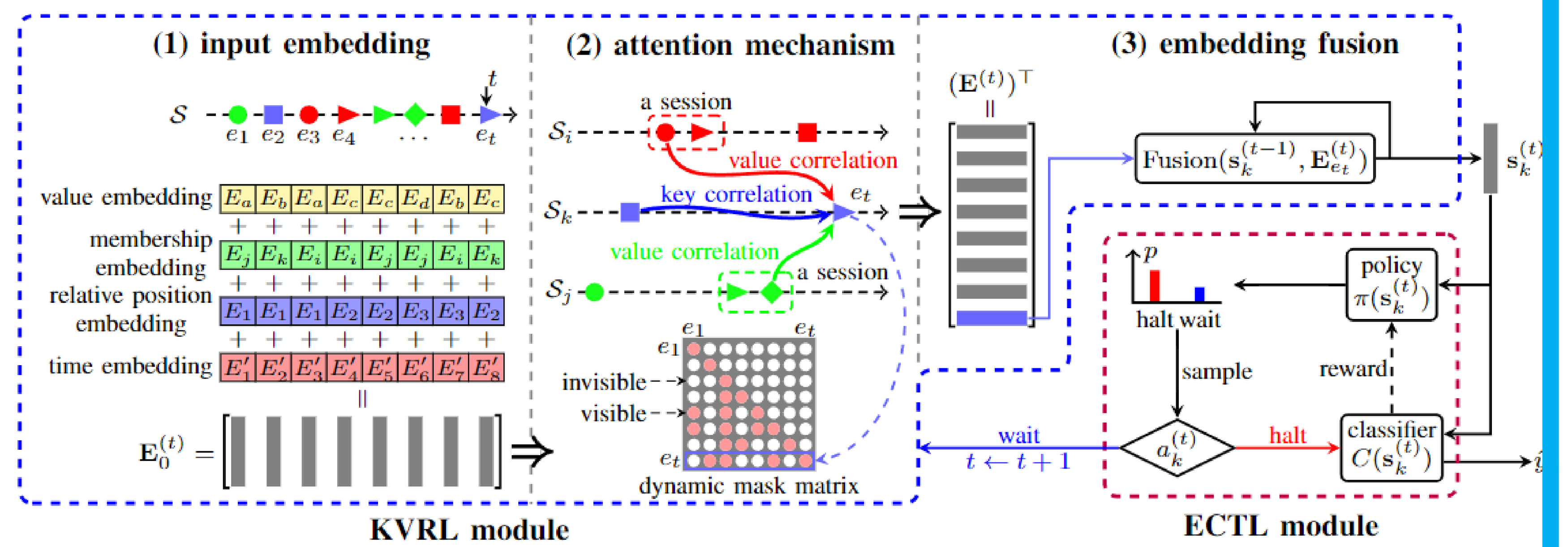




  - how to adaptively determine the number of observations for each key-value sequence?
    - ✓ formulate it as the Partially Observable Markov Decision Process, and solve it through a halting policy.

Methodology

- Ours Key-Value sequence Early Co-classification (KVEC) framework



Overview

✓ The KVRL module is designed to learn an informative representation of the partially observed key-value sequence by exploiting rich item correlations in the tangled key-value sequence.

✓ The ECTL module is designed to adaptively learn to determine a proper number of observations for each key-value sequence and balance the prediction earliness and accuracy.

① Key-Value Sequence Representation Learning (KVRL) module

- Input Embedding: the sum of value embedding, membership embedding, relative position embedding, and time embedding.
- Attention Mechanism: incorporate key correlation and value correlation in the key-value sequence representation learning.
  - ✓ Dynamic Mask Matrix:

$$\mathbf{M}_{ij}^{(t)} \triangleq \begin{cases} 0, & i = j, \\ 0, & (e_i \overset{\text{key}}{\sim} e_j \text{ or } e_i \overset{\text{value}}{\sim} e_j) \text{ and } j \le i, \\ -\infty, & \text{otherwise.} \end{cases}$$

- Embedding Fusion: fuse item embeddings to obtain the sequence representation.

② Early Co-classification Timing Learning (ECTL) module

- State: current key-value sequence representation;
- Policy: decide the next action according to the current sequence representation; $\pi(\mathbf{s}_k^{(t)}) = \sigma(\mathbf{w}_\pi \cdot \mathbf{s}_k^{(t)} + b_\pi)$
- Action: $P(a_k^{(t)} = Halt) = \pi(\mathbf{s}_k^{(t)}),\quad P(a_k^{(t)} = Wait) = 1 - \pi(\mathbf{s}_k^{(t)})$
- Reward:

$$Reward = \begin{cases} +1, & \text{if}(\hat{y}_k = y_k), \\ -1, & \text{otherwise.} \end{cases}$$

③ Model Training

- Minimize the prediction error of the classification network;

$$l_1(\boldsymbol{\theta}_1) \triangleq -\sum_{k=1}^{K}\sum_{c=1}^{C} \mathbf{1}(y_k = c) \log \mathbf{p}_{k,c}(\boldsymbol{\theta}_1)$$

- Maximize the accumulate reward gained by the policy network;

$$l_2(\boldsymbol{\theta}_2) \triangleq -\sum_{k=1}^{K}\sum_{i=1}^{n_k} (R_k^{(i)} - b_k^{(i)}(\boldsymbol{\theta}_b)) \log P(a_k^{(i)} | \mathbf{s}_k^{(i)}; \boldsymbol{\theta}_\pi)$$

- Encourage early prediction: $l_3(\boldsymbol{\theta}_3) \triangleq -\sum_{k=1}^{K}\sum_{i=1}^{n_k} \log P(a_k^{(i)} = \mathrm{Halt} | \mathbf{s}_k^{(i)}; \boldsymbol{\theta}_\pi)$

✓ Total Training Loss: $l(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \triangleq l_1(\boldsymbol{\theta}_1) + \alpha l_2(\boldsymbol{\theta}_2) + \beta l_3(\boldsymbol{\theta}_3)$

Experiments

① Experimental Setup

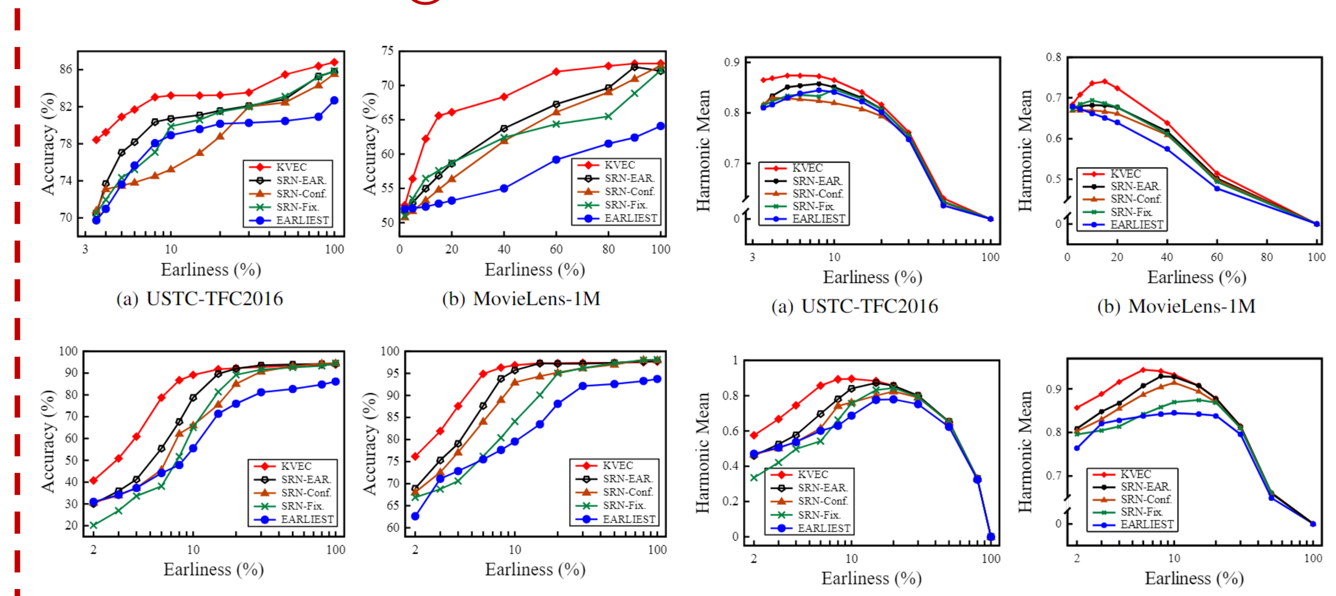
- Data

dataset	#keys	avg $\lvert\mathcal{S}_k\rvert$	avg session length	#classes
USTC-TFC2016	3,200	31.2	8.3	9
MovieLens-1M	6,040	163.5	1.7	2
Traffic-FG	60,000	50.7	2.4	12
Traffic-App	50,000	57.5	2.7	10
Synthetic-Traffic	10,000	100.0	2.1	2

- Metrics
  - Earliness: $\text{Earliness} \triangleq \frac{1}{K}\sum_{k=1}^{K}\frac{n_k}{\lvert\mathcal{S}_k\rvert}$
  - Accuracy, Precision, Recall, F1-score
  - HM: harmonic mean of Accuracy and Earliness, measure the multi-objective balancing ability of different methods.

$$\text{HM} \triangleq \frac{2 \times (1 - \text{Earliness}) \times \text{Accuracy}}{1 - \text{Earliness} + \text{Accuracy}}$$

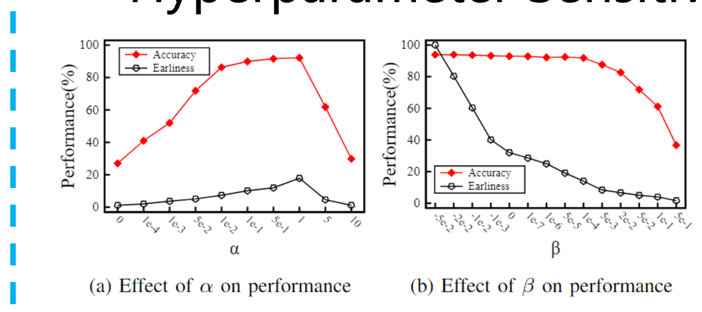
② Overall Performance



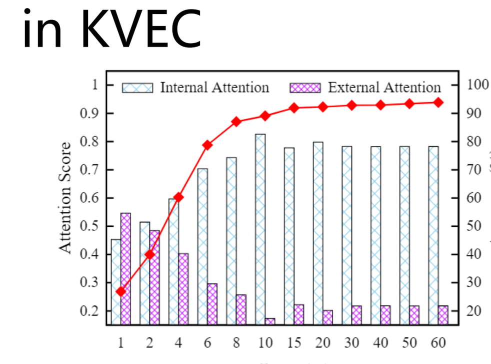
✓ KVEC achieves 4.7−17.5% accuracy improvement, 3.7−14.0% HM improvement.

③ More Discussions

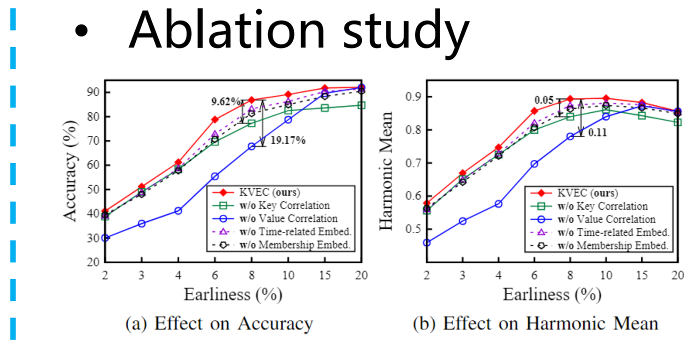
- Hyperparameter Sensitivity



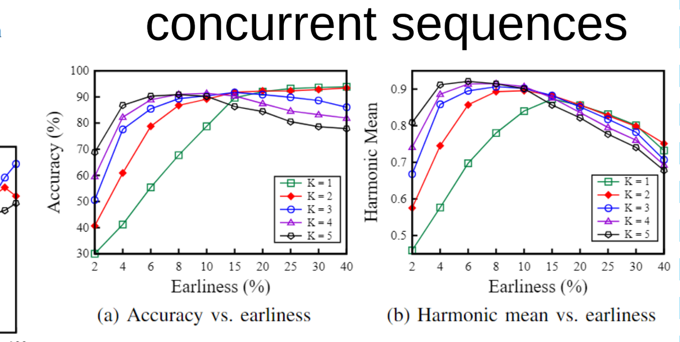
- Attention mechanism in KVEC



- Ablation study



- Effects of the number of concurrent sequences



- Halting policy in KVEC