Multi-Aspect Mining of Complex Sensor Sequences

Takato Honda[1] , Yasuko Matsubara[1], Ryo Neyama[2], Mutsumi Abe[2], Yasushi Sakurai[1]

[1]AIRC-ISIR, Osaka University

[2]Toyota Motor Corporation

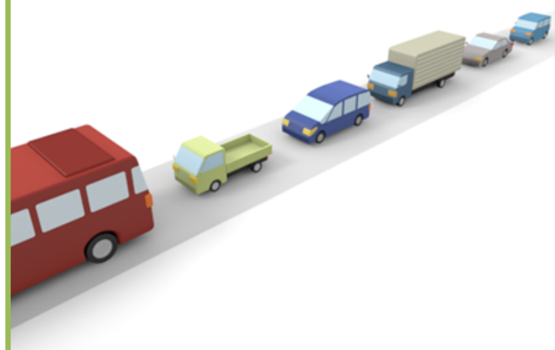
Motivation

Analysis of IoT sensor data, e.g., car
- Advanced driving assistance service

Risk

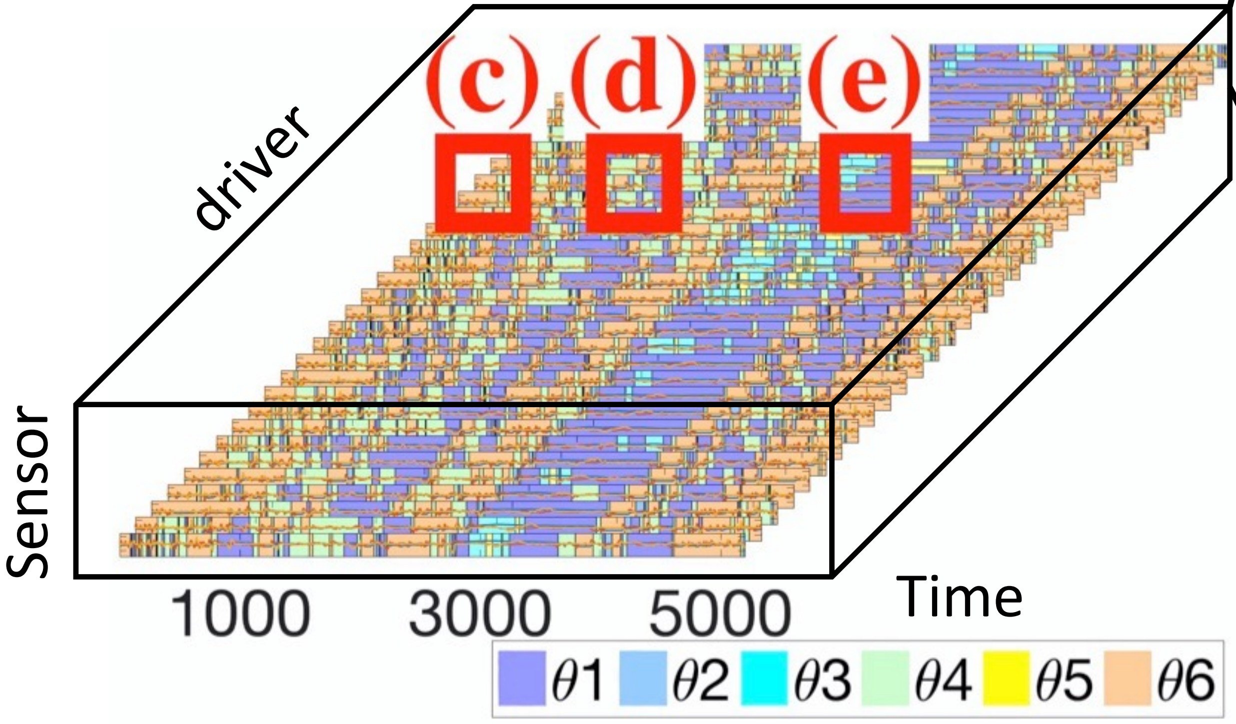
Fuel

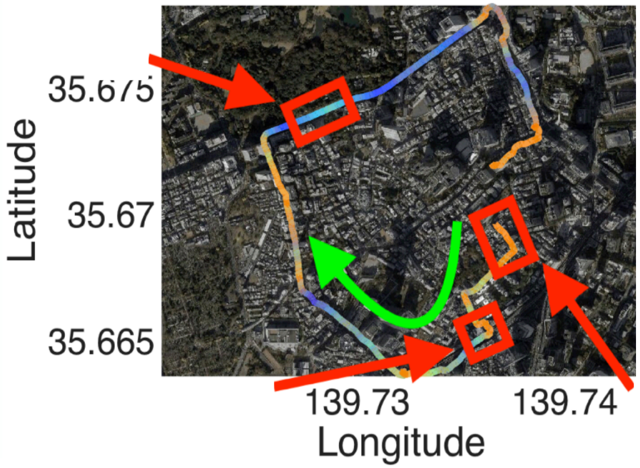
Congestion



Motivation

IoT sensor data is a tensor (sensor × driver × time)



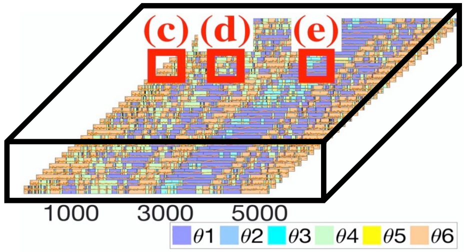


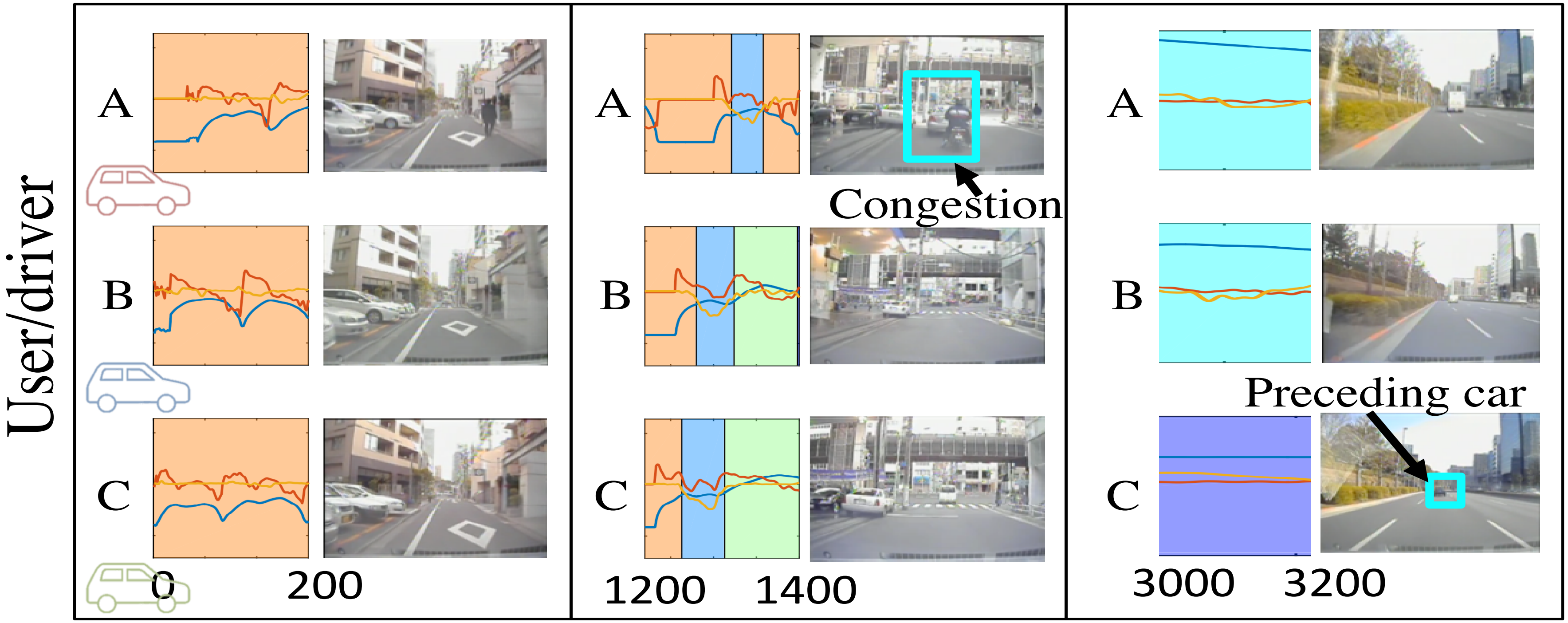
(a) Time series tensor of automobile dataset

(b) On a map

Motivation

Tensor has multi-aspect patterns: time-aspect and user-aspect

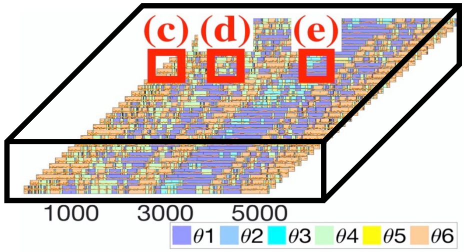


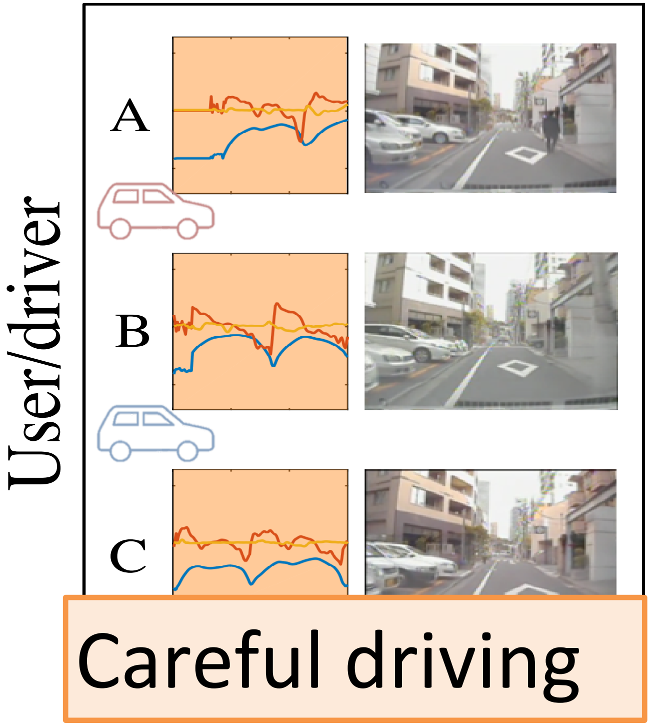


(c) Narrow road (d) Intersection (c) Wide road

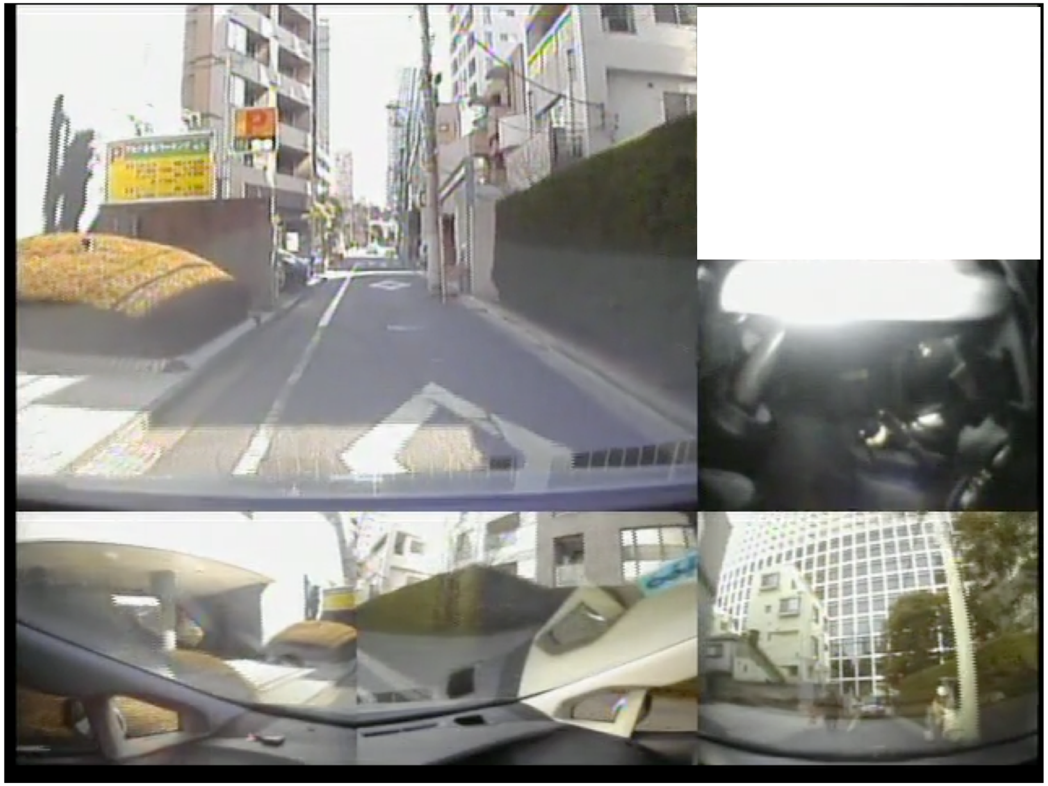
Motivation

Tensor has multi-aspect patterns: time-aspect and user-aspect



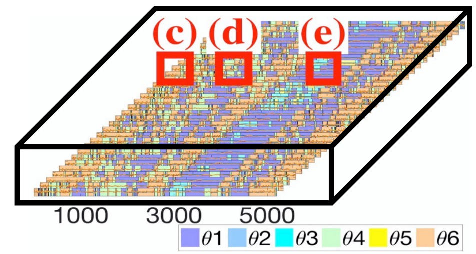


(c) Narrow road

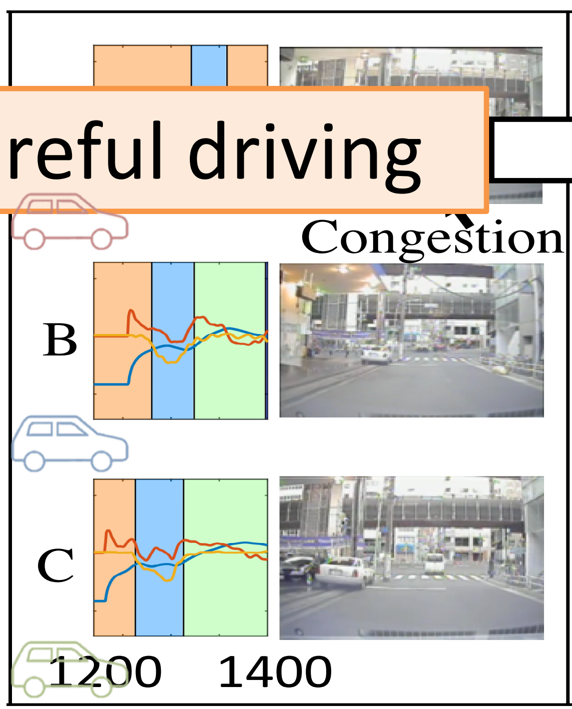


Motivation

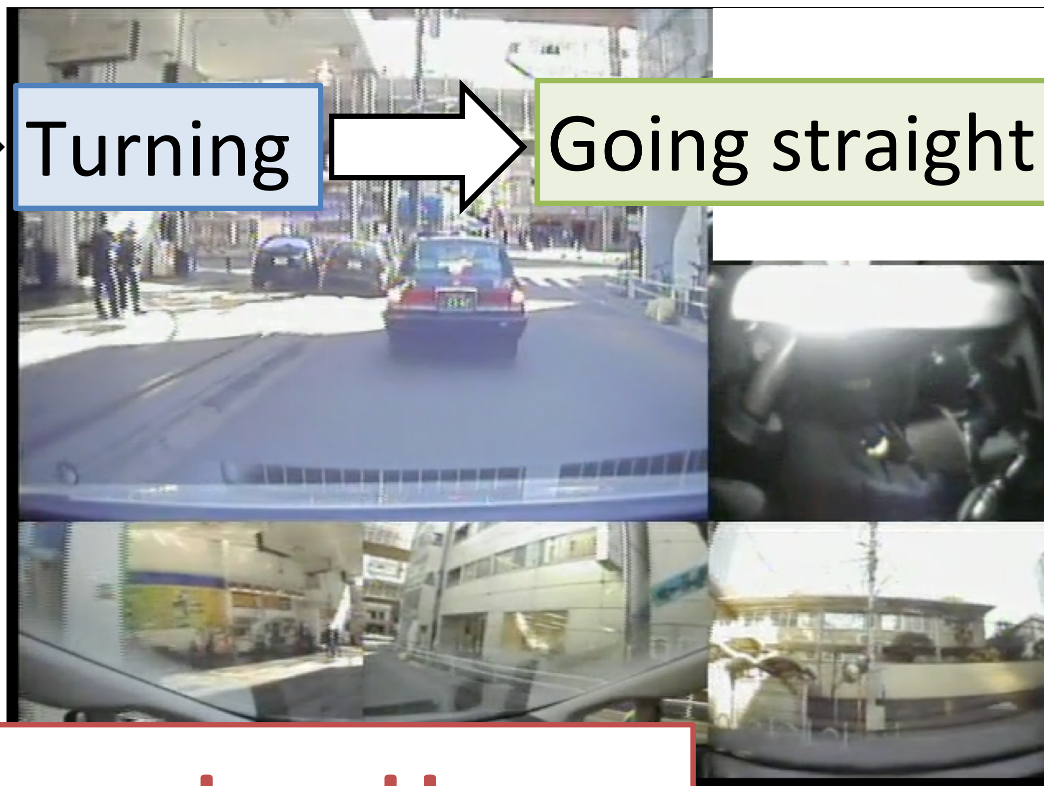
Tensor has multi-aspect patterns: time-aspect and user-aspect



Careful driving → Turning → Going straight



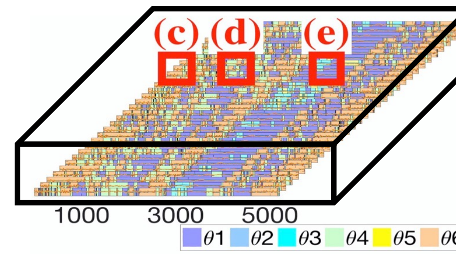
(d) Inters

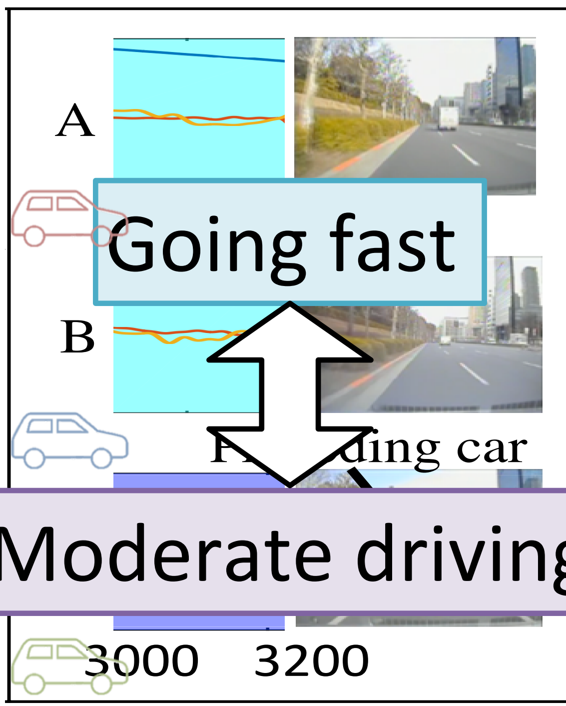


Time-aspect patterns

Motivation

Tensor has multi-aspect patterns:
time-aspect and user-aspect





Going fast

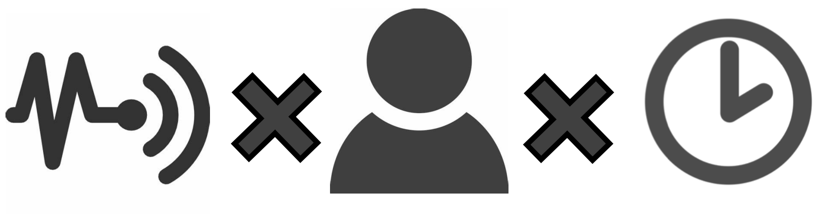
Moderate driving

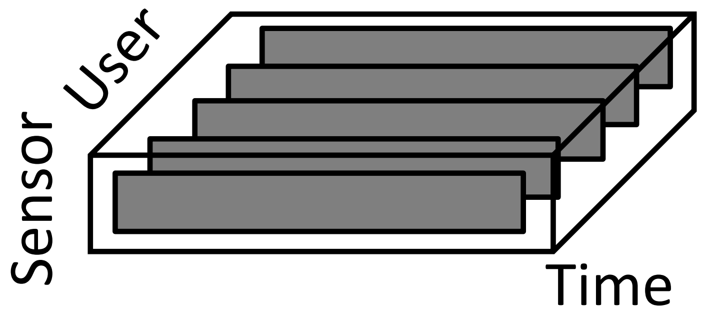
(e) Wide

User-aspect patterns

Motivation

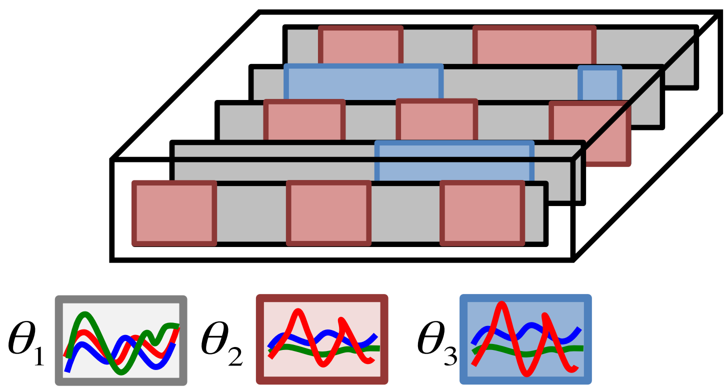
Given: Time-series tensor
(sensor × user × time)





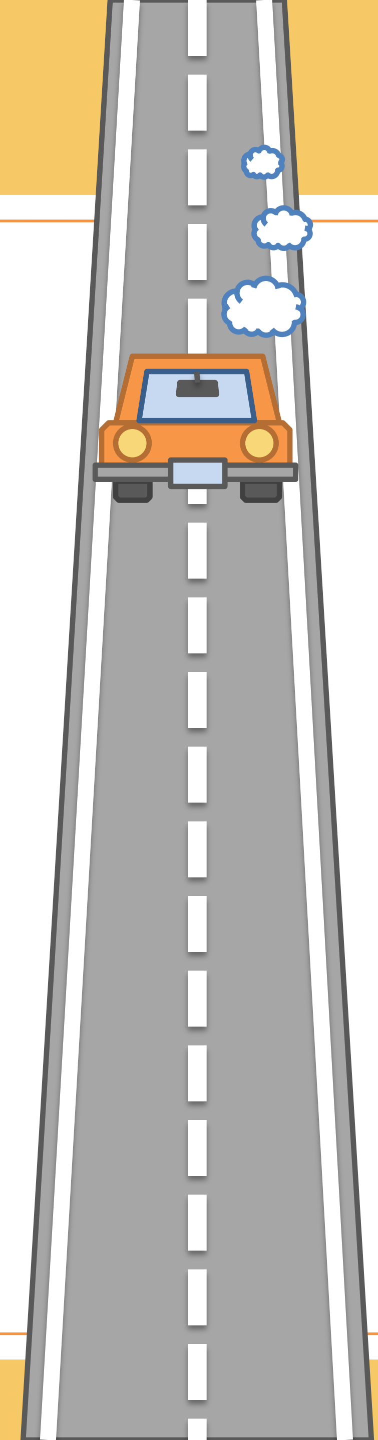
Find: Multi-aspect patterns
(time and user-aspect)

Automatically & quickly



Outline

- Motivation
- **Problem definition**
- Main ideas
- Algorithms
- Experiments
- Conclusions



Problem definition

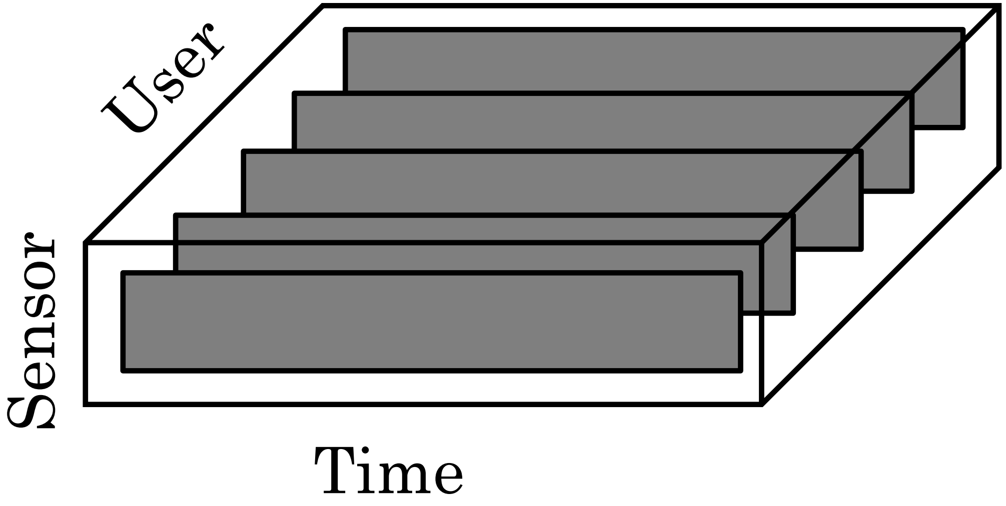
Key concepts

- **Tensor:** $\mathcal{X}$ given
- **Segment:** $\mathcal{S}$ hidden
- **Regime:** Θ hidden
- **Segment-membership:** $\mathcal{F}$ hidden

Problem definition

Tensor : $\mathcal{X} \in R^{d \times w \times n} = \{X_1, \dots, X_w\}$

given



Problem definition

Segment : $S = \{s_1, \dots, s_m\}$

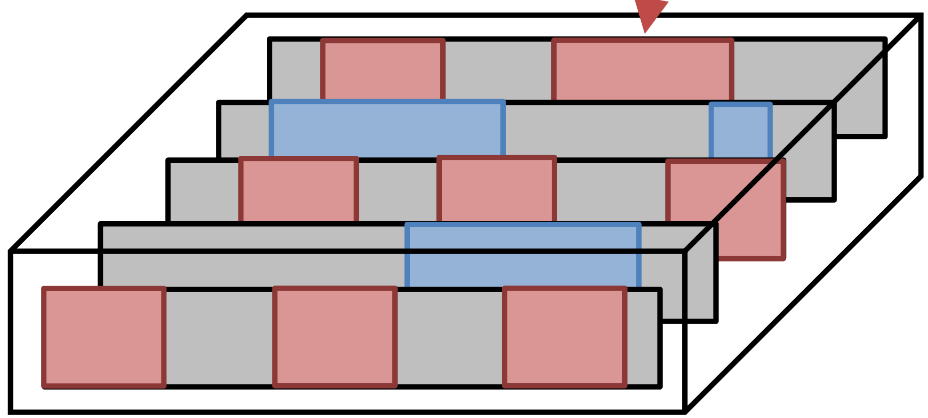
hidden

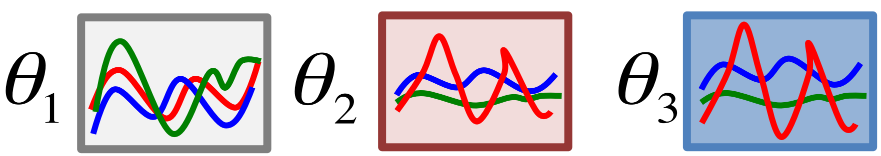
$s_i = \{\boxed{t_s}, \boxed{t_e}, userID\}$

start position

end position

m = 25 segments





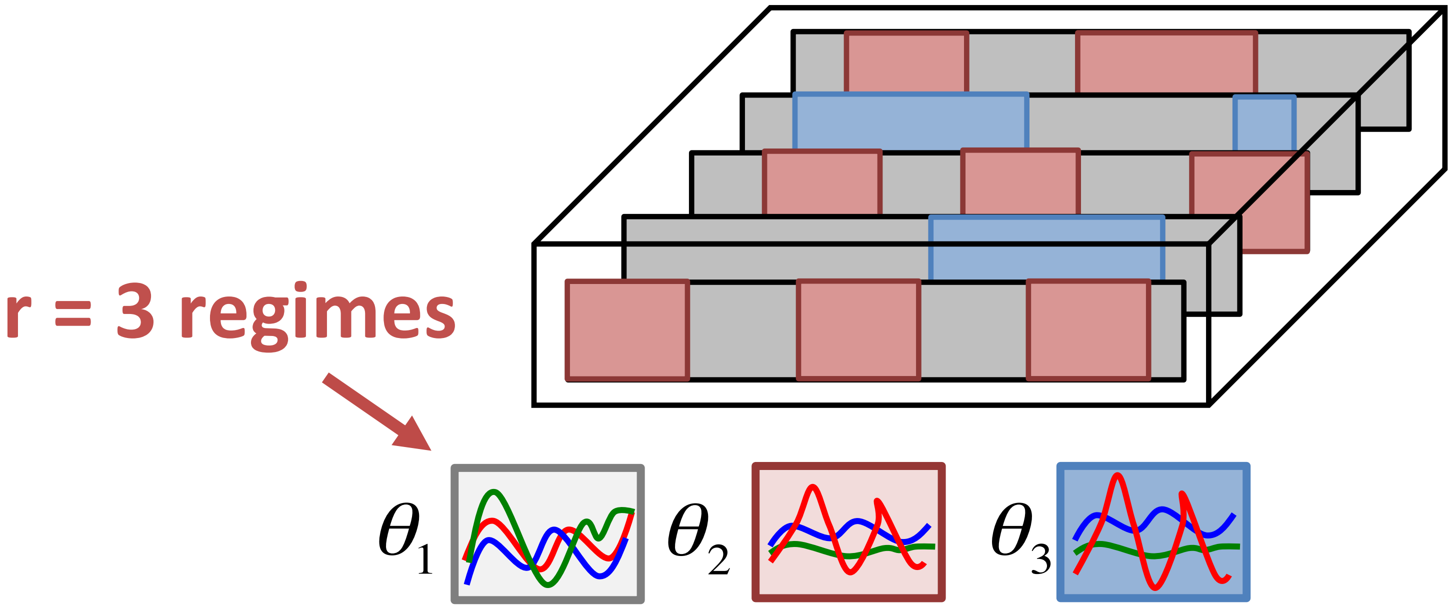
Problem definition

Regime: $\Theta = \{\theta_1, \theta_2, \dots, \theta_r, \Delta_{r \times r}\}$

hidden

$\theta_i = \{\boxed{\pi}, \boxed{A}, \boxed{B}\}$ (hidden Markov model)

Initial prob. transition prob. output prob.



Problem definition

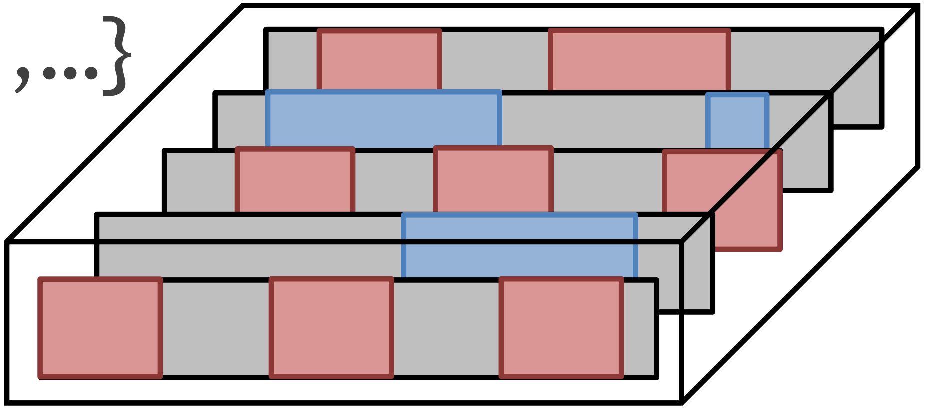
Membership: $F = \{f_1, f_2, \dots, f_m\}$

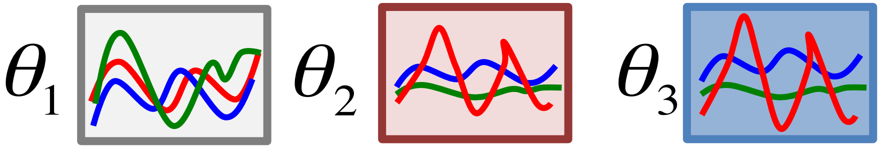
hidden

$1 \leq f_i \leq r$

Example:

F = {1,2,1,2,1,...}

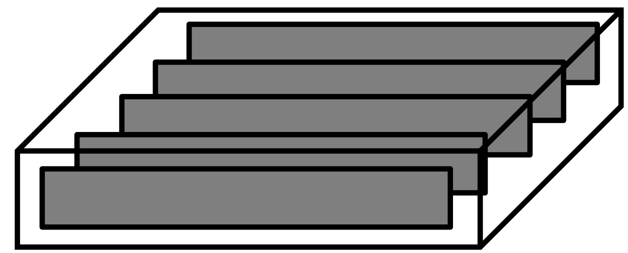




Problem definition

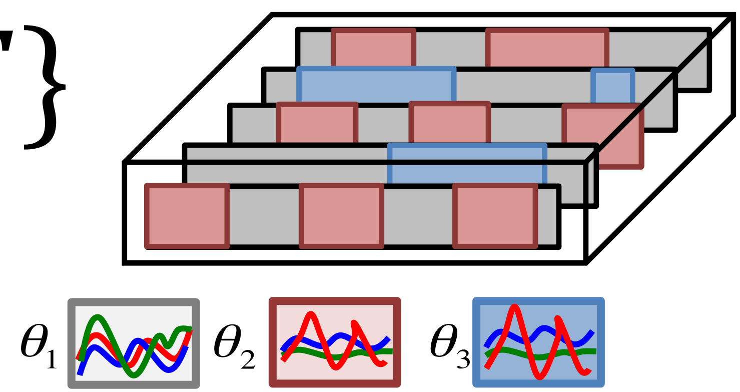
Given: tensor $\mathcal{X}$

$$\mathcal{X} = \{X_1, \dots, X_w\}$$



Find: compact description C of $\mathcal{X}$

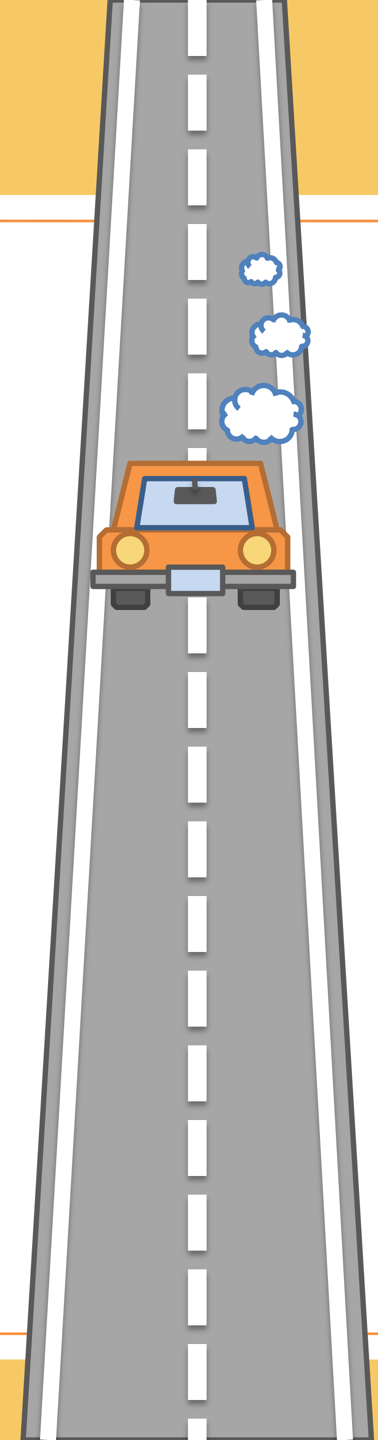
$$C = \{m, r, S, \Theta, F\}$$



Automatically & quickly

Outline

- Motivation
- Problem definition
- **Main ideas**
- Algorithms
- Experiments
- Conclusions



Main ideas

Goal: compact description of

$$C = \{m, r, S, \Theta, F\}$$

without user intervention

Challenges:

Q1. How to decide m and r **automatically**

Q2. How to find **multi-aspect** regimes

Main ideas

Goal: compact description of

$$C = \{m, r, S, \Theta, F\}$$

without user intervention

Challenges:

Q1. How to decide m and r automatically

Idea 1: Model description cost

Q2. How to find multi-aspect regimes

Idea 2: Multi-splitting algorithm

(1): model description cost

Q1. How to decide # of regimes/segments?

Idea 1: Model description cost

- Minimize coding cost
- Optimal # of segments/regimes

Good compression ⇄ Good description

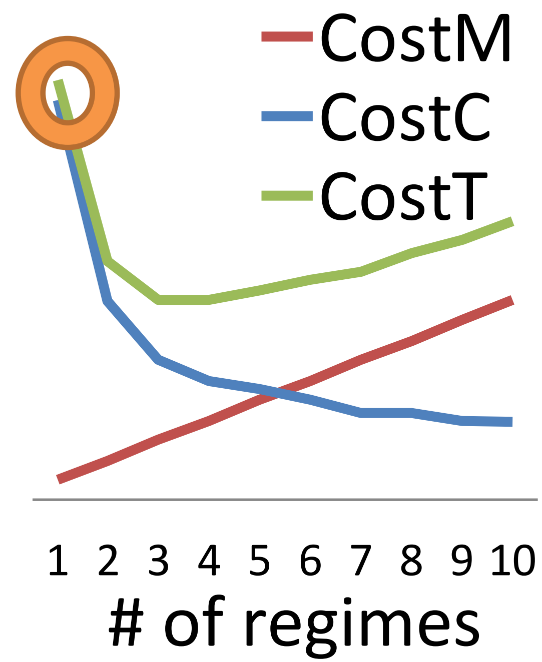
(1): model description cost

Idea: Minimize total cost

$$\min \left(\boxed{Cost_M(M)} + \boxed{Cost_c(X \mid M)} \right)$$

Model cost

Coding cost



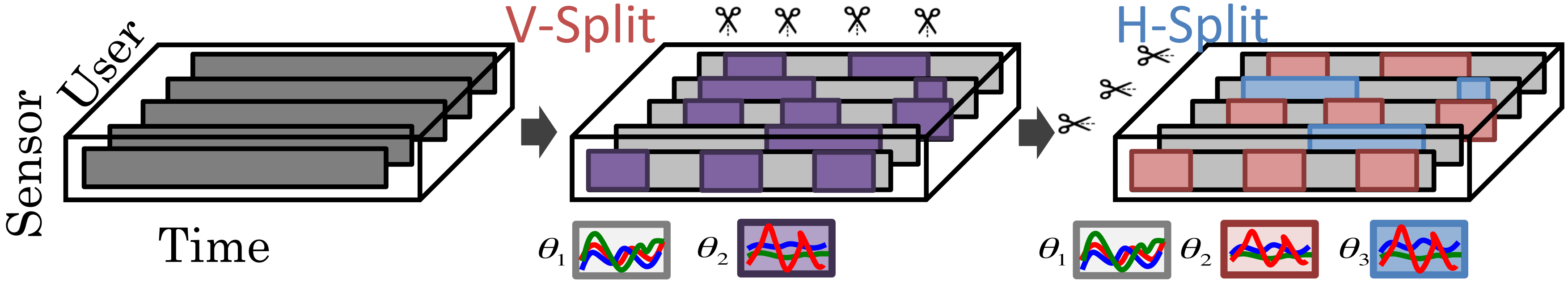
Good compression ⟲ Good description

(2): Multi-aspect mining

Q2. How to find multi-aspect regimes?

Idea 2: Multi-aspect splitting algorithm

- Find time-aspect transitions
- And their differences between users



Tensor X
Regimes
Regimes

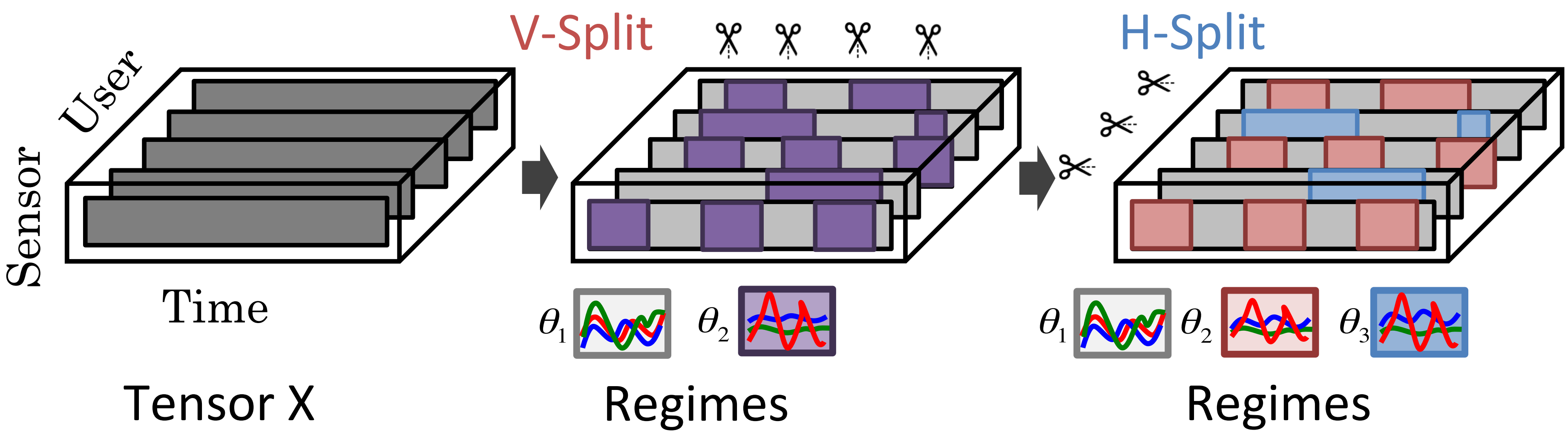
(2): Multi-aspect mining

V-Split (vertical):

split $\mathcal{X}$ into time-aspect

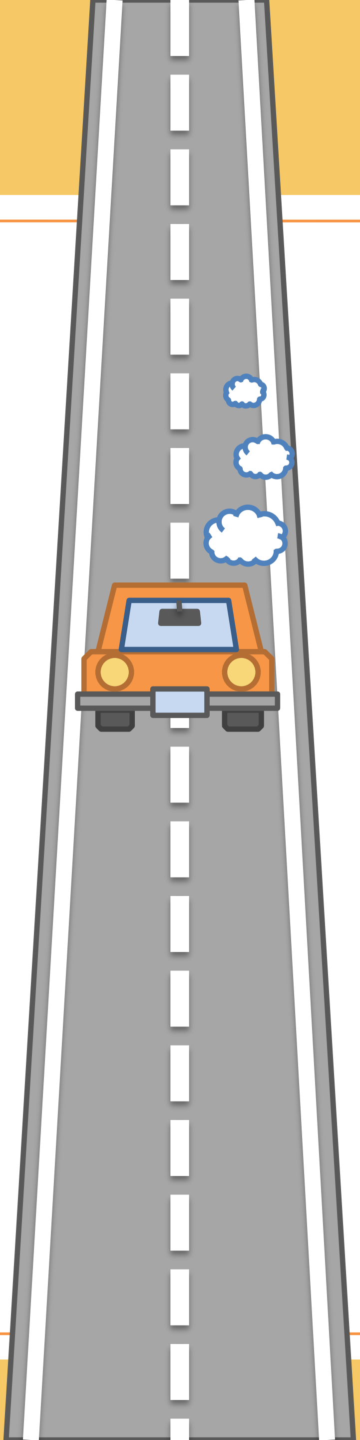
H-Split (horizontal):

split $\mathcal{X}$ into user-aspect



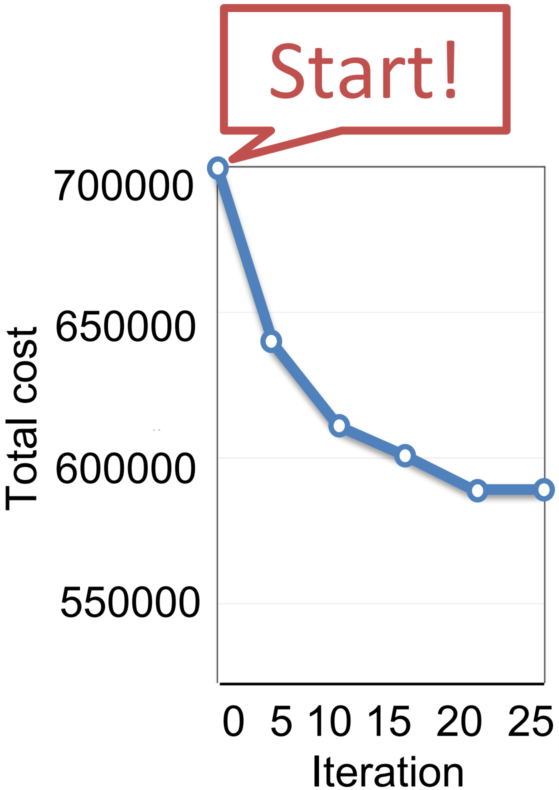
Outline

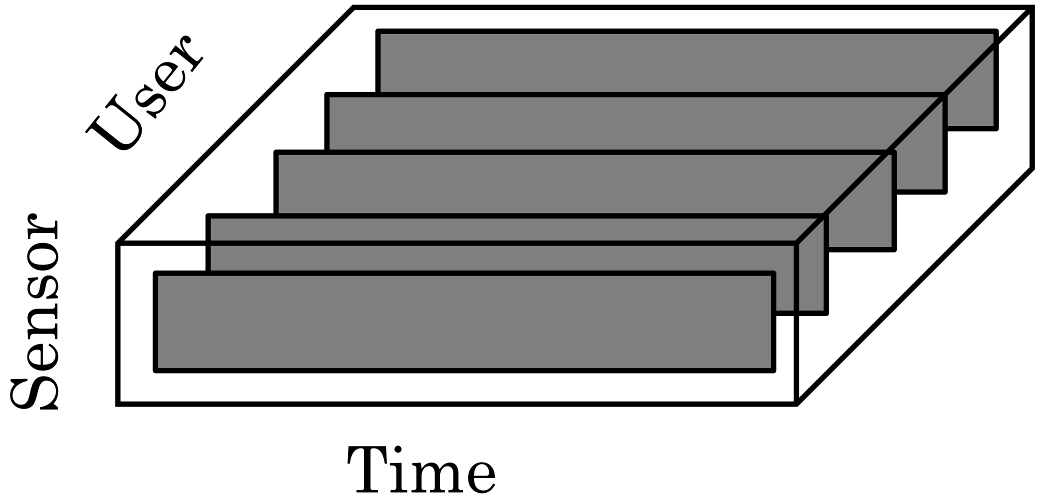
- Motivation
- Problem definition
- Main ideas
- **Algorithms**
- Experiments
- Conclusions



Proposed algorithm

Overview

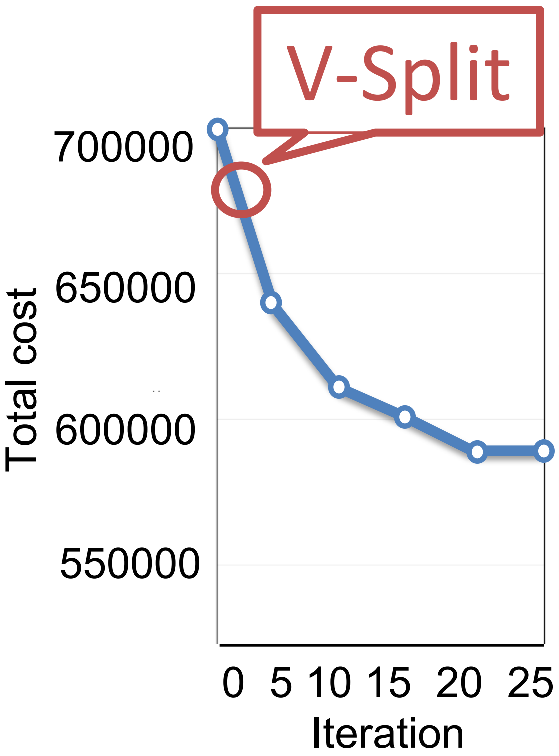


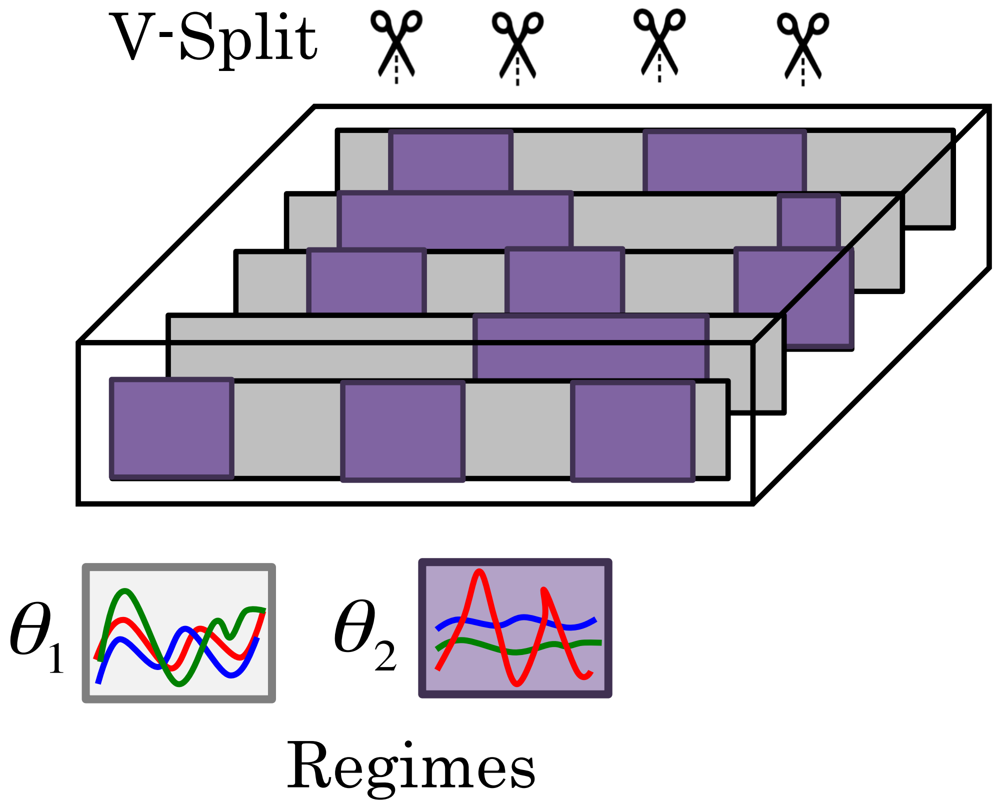


Iteration 0 (r=1)

Proposed algorithm

Overview

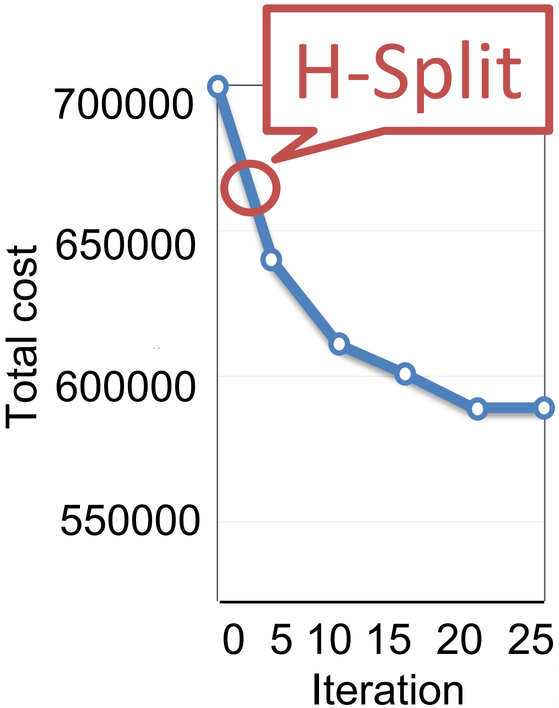


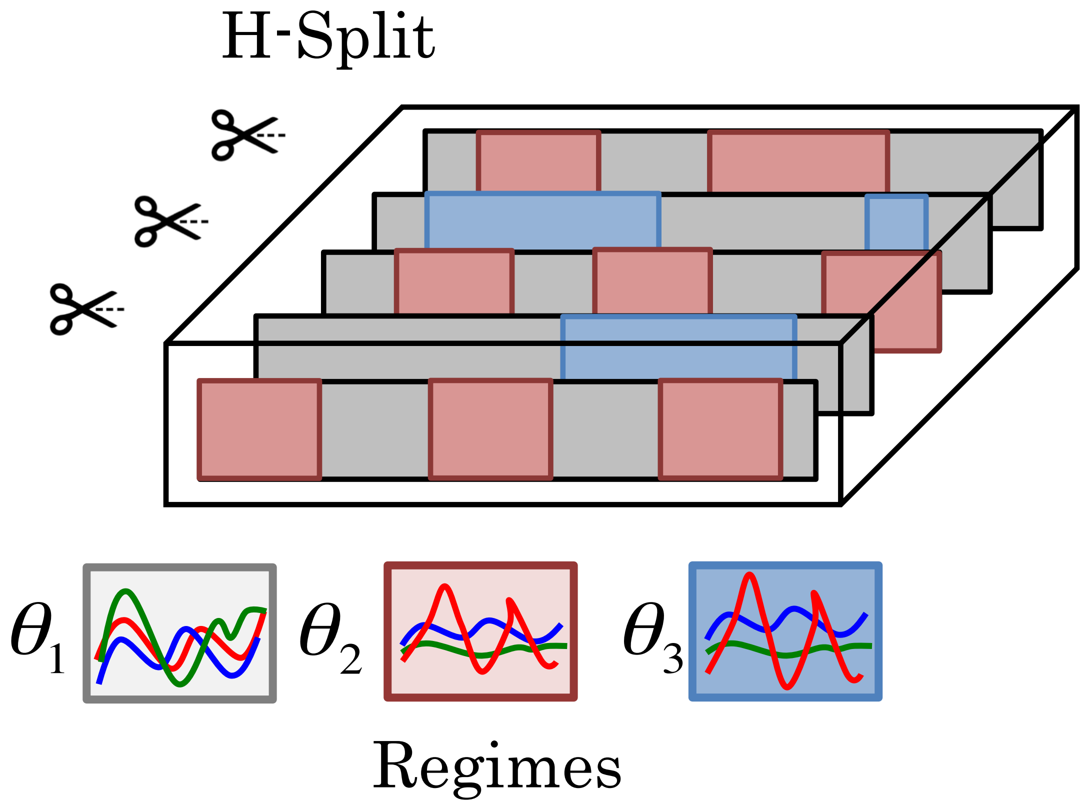


Iteration 1 (r=2)

Proposed algorithm

Overview





Iteration 2 (r=3)

Algorithms

Algorithms of our method

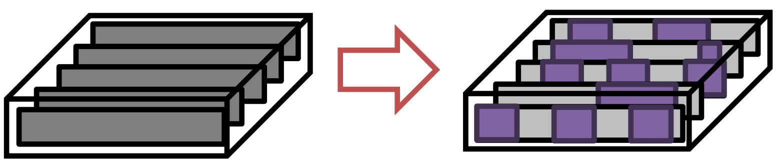
CubeMarker

Outer loop

V-Split

Inner loop

- V-Assignment
- ModelEstimation



Find time-aspect regime

H-Split

Inner loop

- H-Assignment
- ModelEstimation

Find user-aspect regime

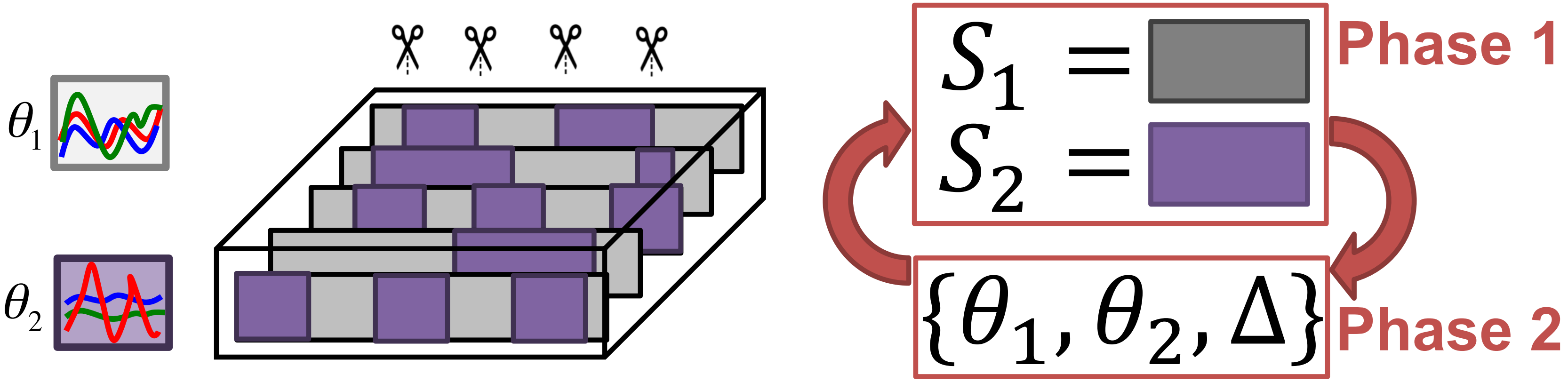
Decide splitting algorithm

V-Split

Inner loop

Two phase iterative approach

- **Phase 1:** (V-Assignment)
  - Split segments into two groups: S_1, S_2
- **Phase 2:** (ModelEstimation)
  - Update model parameters: $\Theta = \{\theta_1, \theta_2, \Delta\}$



Algorithms of our method

CubeMarker

Outer loop

V-Split

Inner loop

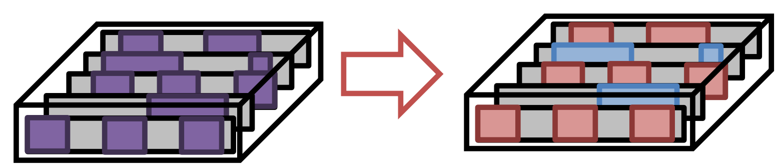
- V-Assignment
- ModelEstimation

Find time-aspect regime

H-Split

Inner loop

- H-Assignment
- ModelEstimation



Find user-aspect regime

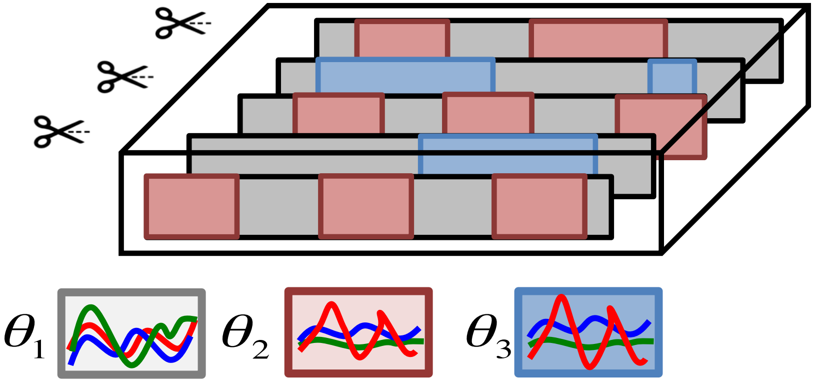
Decide splitting algorithm

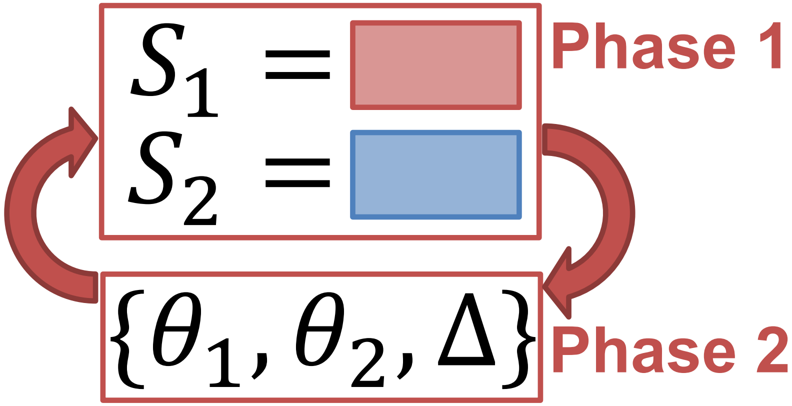
H-Split

Inner loop

<u>**Two phase iterative approach**</u>

- **Phase 1: (H-Assignment)**
  - Split segments into two groups: S_1, S_2
- **Phase 2: (ModelEstimation)**
  - Update model parameters: $\Theta = \{\theta_1, \theta_2, \Delta\}$





H-Split

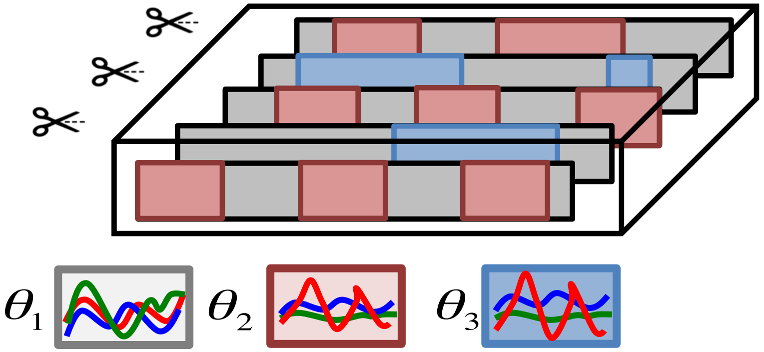
Inner loop

Given:

- tensor $\mathcal{X}$
- model parameter set $\Theta = \{\theta_1, \theta_2, \Delta\}$

Find: two user-aspect regimes based on the similarity: $Cost_C(X_i|\theta_j)$

$$\left.\begin{matrix} \mathcal{X} \\ \\ \{\theta_1, \theta_2, \Delta\} \end{matrix}\right\}$$



Algorithms

Algorithms of our method

CubeMarker

Outer loop

V-Split

Inner loop

- V-Assignment
- ModelEstimation

Find time-aspect regime

H-Split

Inner loop

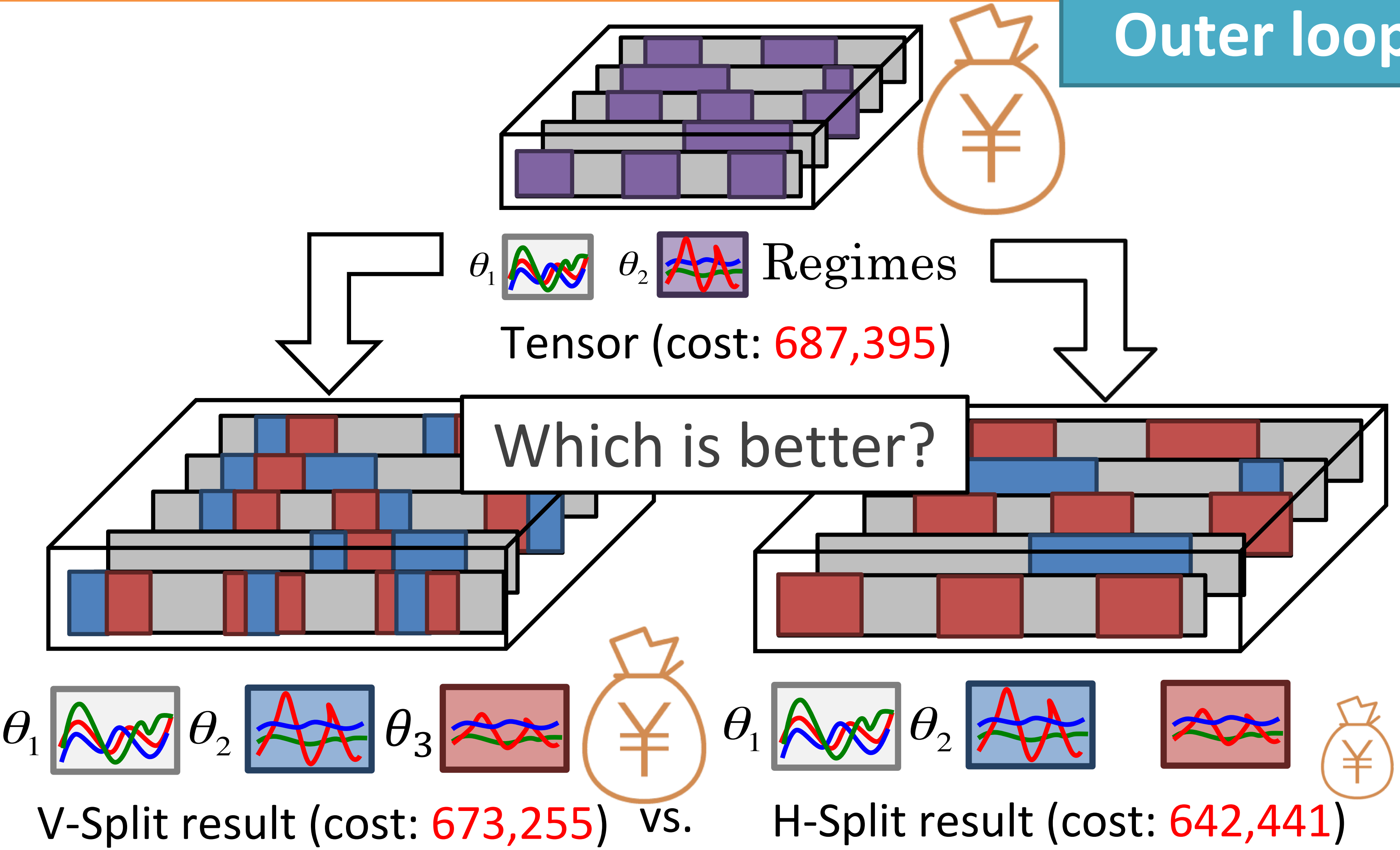
- H-Assignment
- ModelEstimation

Find user-aspect regime

Decide splitting algorithm

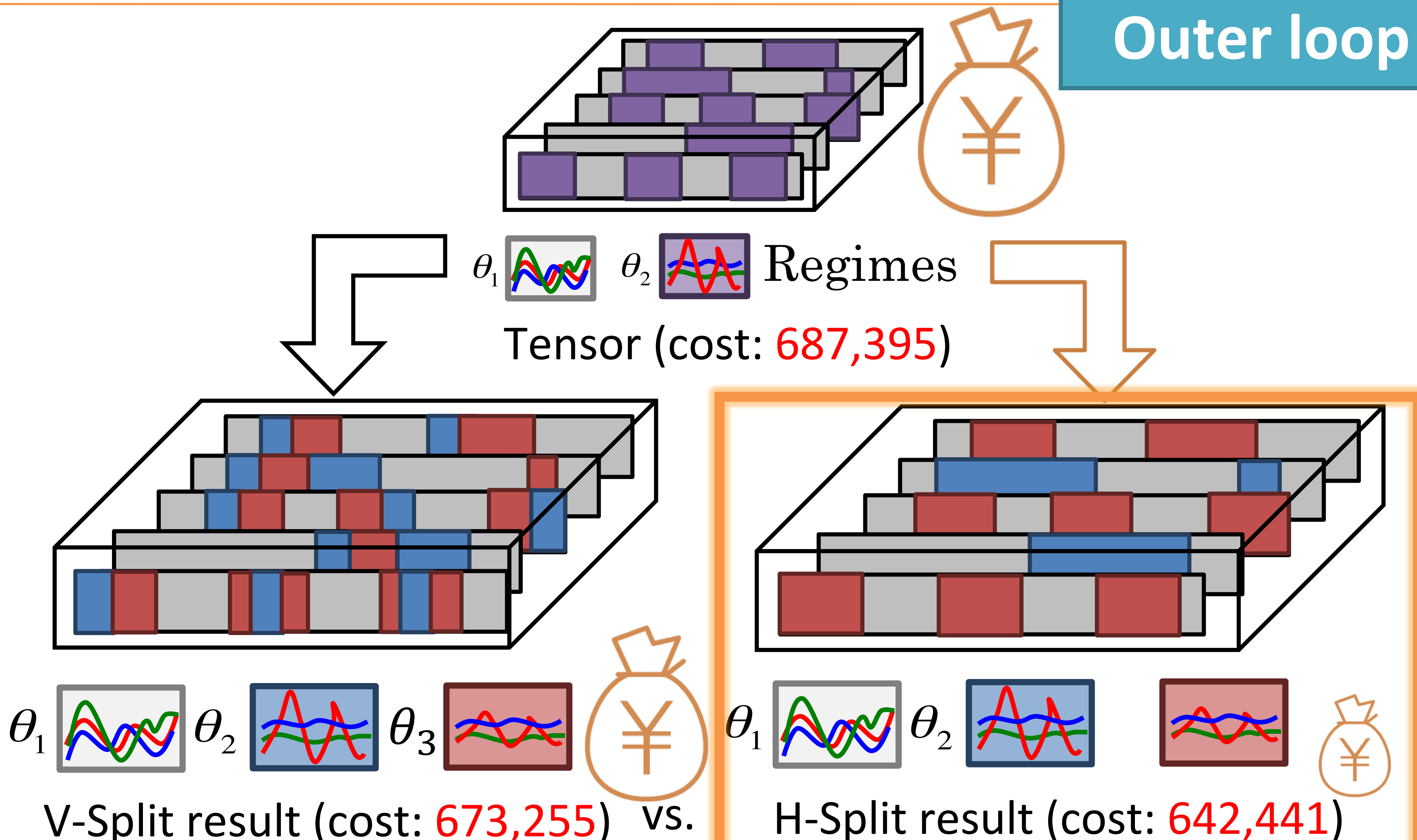
CubeMarker

Outer loop



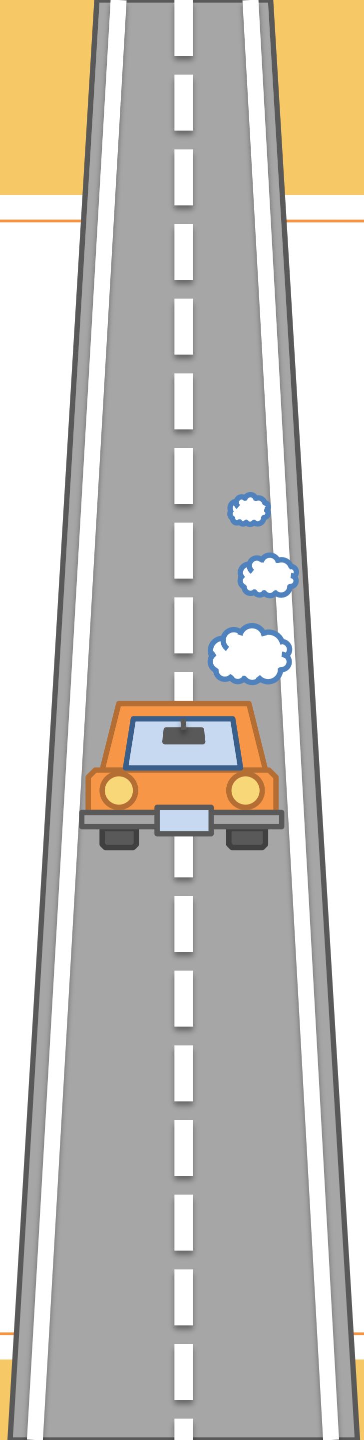
CubeMarker

Outer loop



Outline

- Motivation
- Problem definition
- Main ideas
- Algorithms
- **Experiments**
- Conclusions



Experiments

Q1. Effectiveness

Can it help us understand the given tensor?

Q2. Scalability

How does it scale in terms of computational cost?

Q3. Accuracy

How well does it find segments and regimes?

Competitors:
pHMM (SIGMOD’11)
AutoPlait (SIGMOD’14)
TICC (KDD’17)
CubeMarker-V (naïve ver. of our method)

Datasets

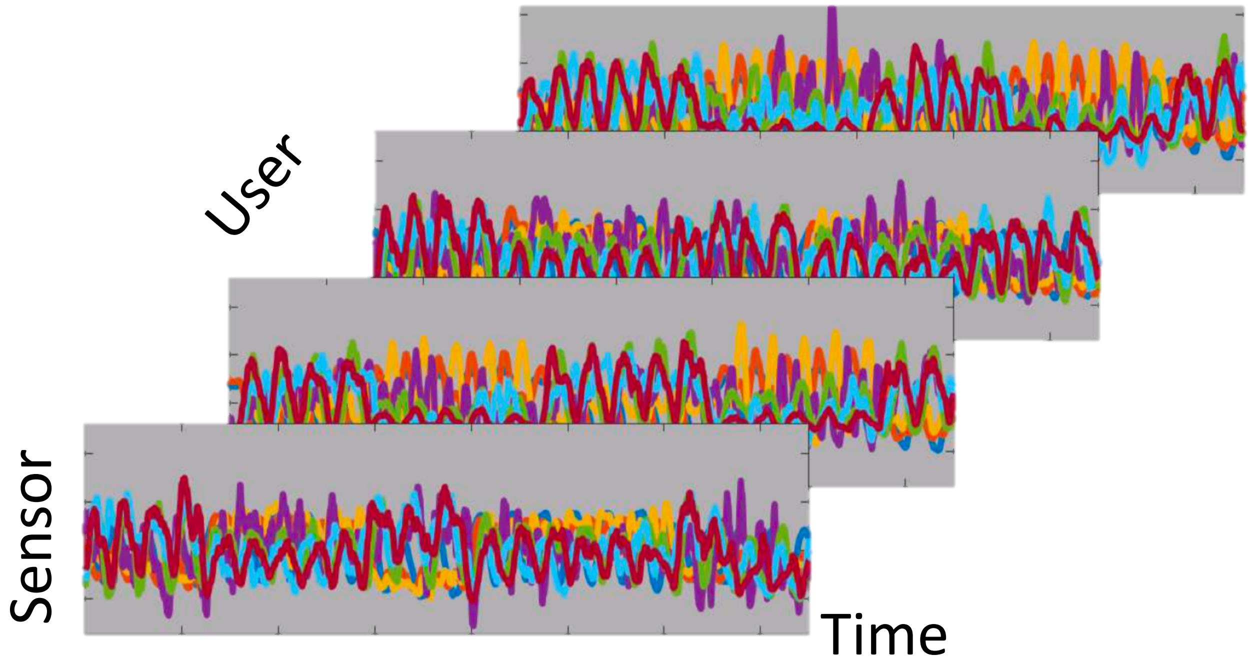
Experiments on the 8 real-world datasets:

Dataset	Data size ($w \times n \times d$)
(#1) Workout	$182 \times 4000 \times 7$
(#2) Tennis	$100 \times 4500 \times 7$
(#3) Factory	$60 \times 3000 \times 7$
(#4) Reading	$71 \times 10000 \times 5$
(#5) Free throw	$170 \times 2000 \times 7$
(#6) Automobile-Tokyo	$171 \times 2400 \times 3$
(#7) Automobile-Expressway	$13 \times 9100 \times 3$
(#8) Automobile-Togu	$32 \times 5200 \times 3$

Summary of the datasets

Q1. Effectiveness - Workout

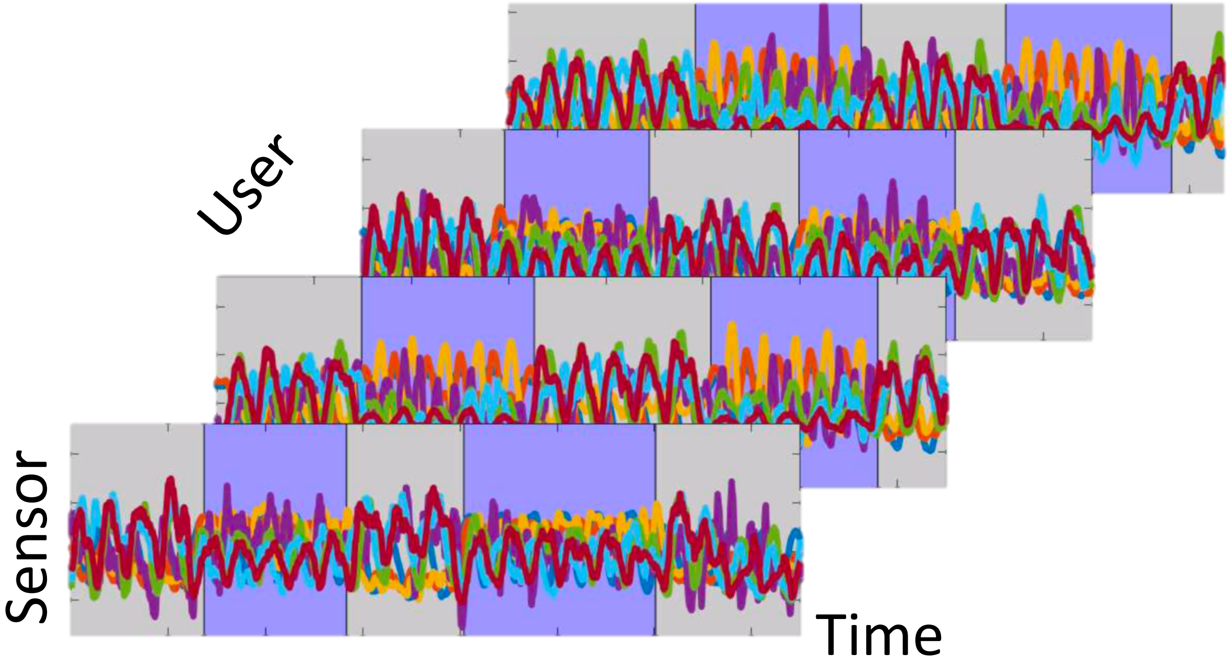
How many and what kind of patterns does it include?



Original workout dataset

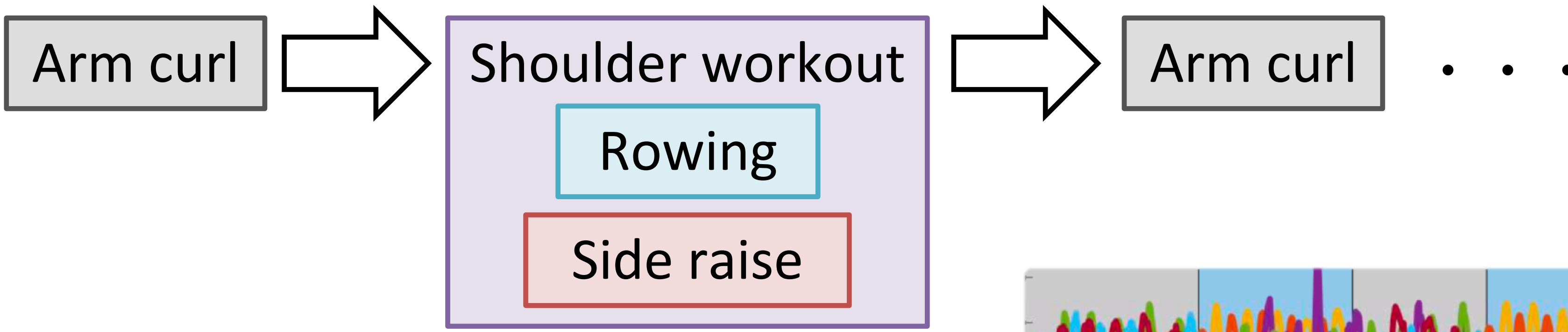
Q1. Effectiveness - Workout

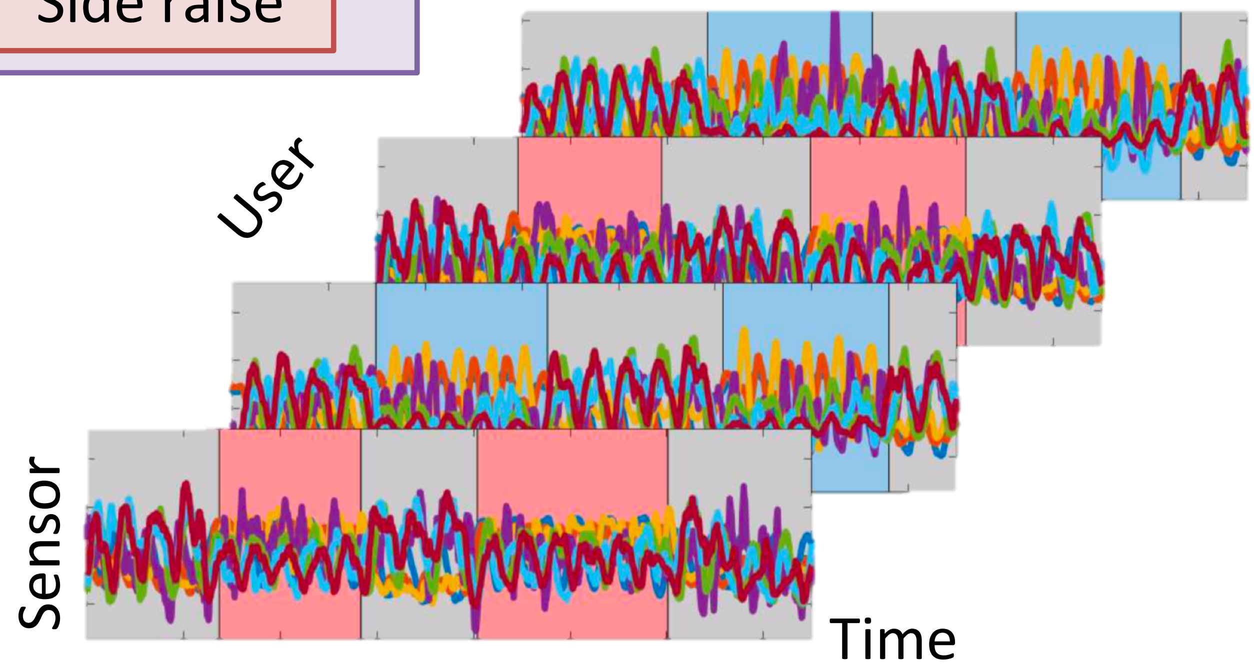
Arm curl → Shoulder workout → Arm curl . . .



Time-aspect patterns for a workout dataset

Q1. Effectiveness - Workout



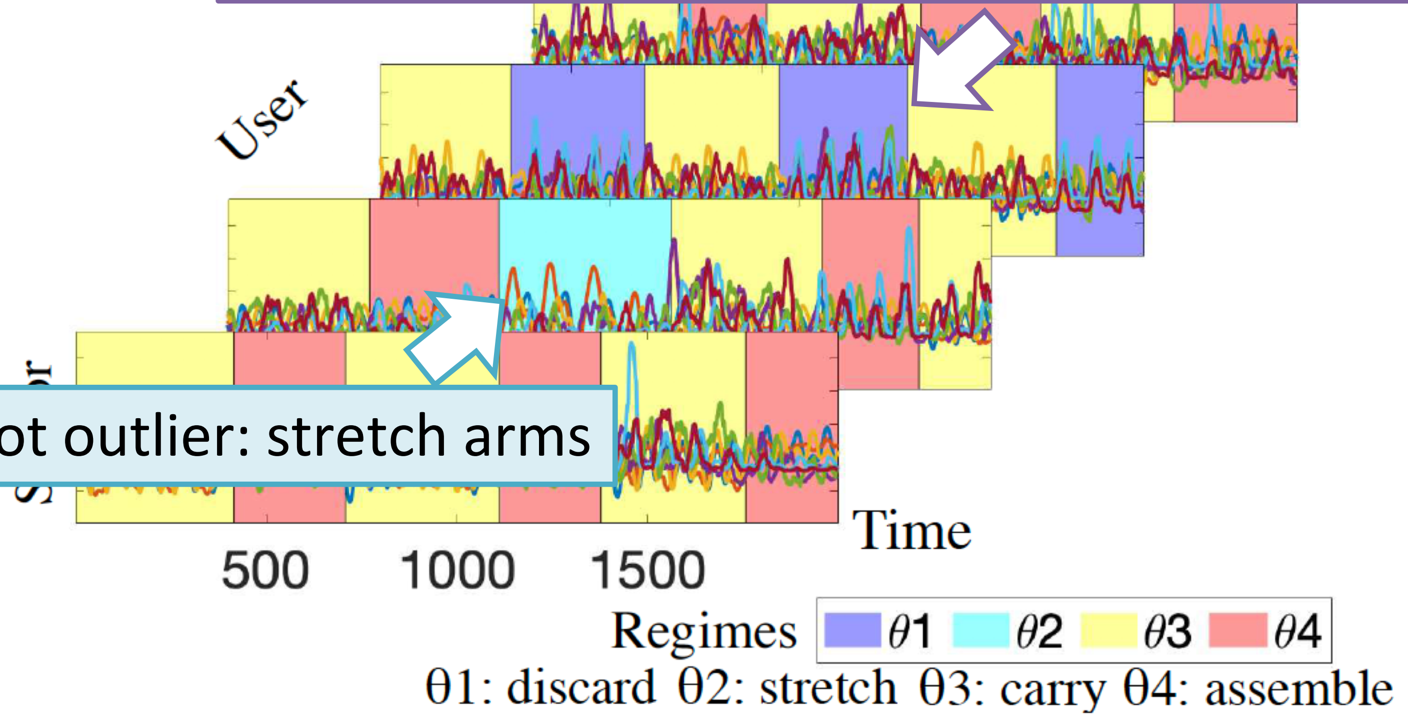


Multi-aspect patterns for a workout dataset

Q1. Effectiveness - Factory worker

Basic pattern transitions: carrying ⇨ assembling ⇨ · · ·

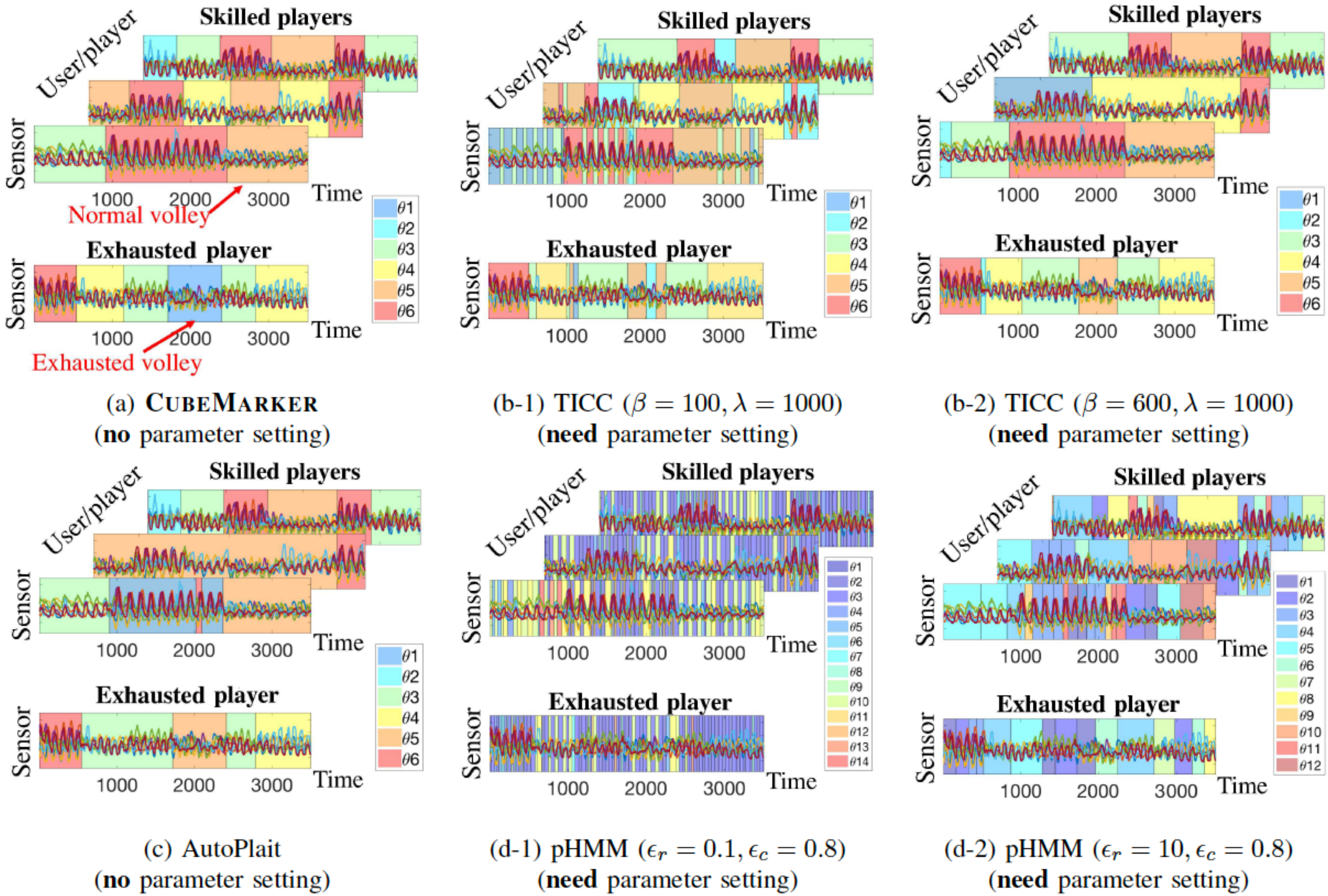
User-aspect pattern: Discarding defective products



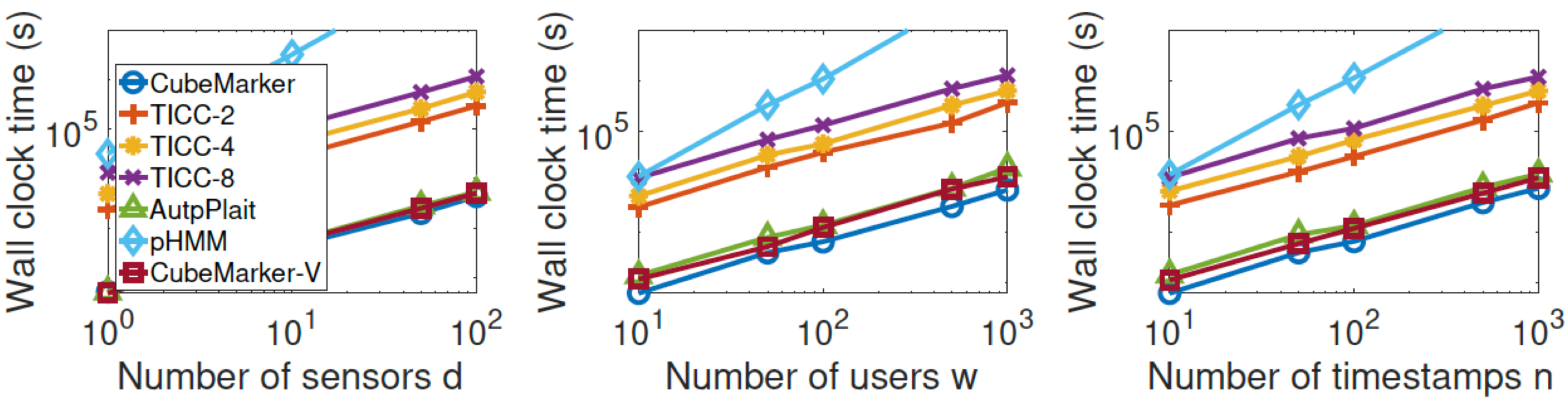
One-shot outlier: stretch arms

Multi-aspect patterns for a factory workers

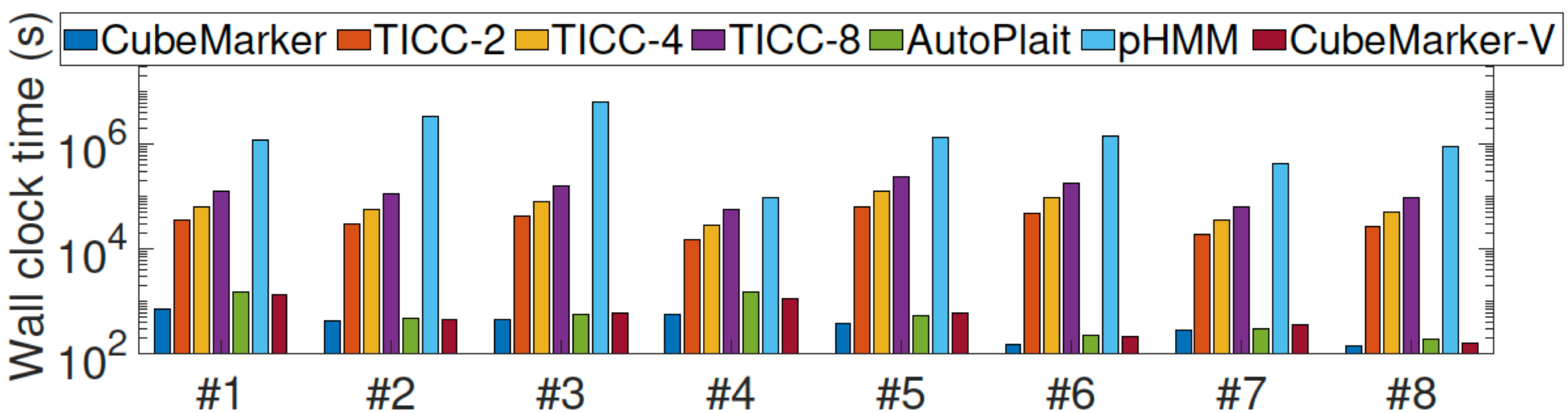
Q1. Effectiveness - Tennis



Q2. Scalability

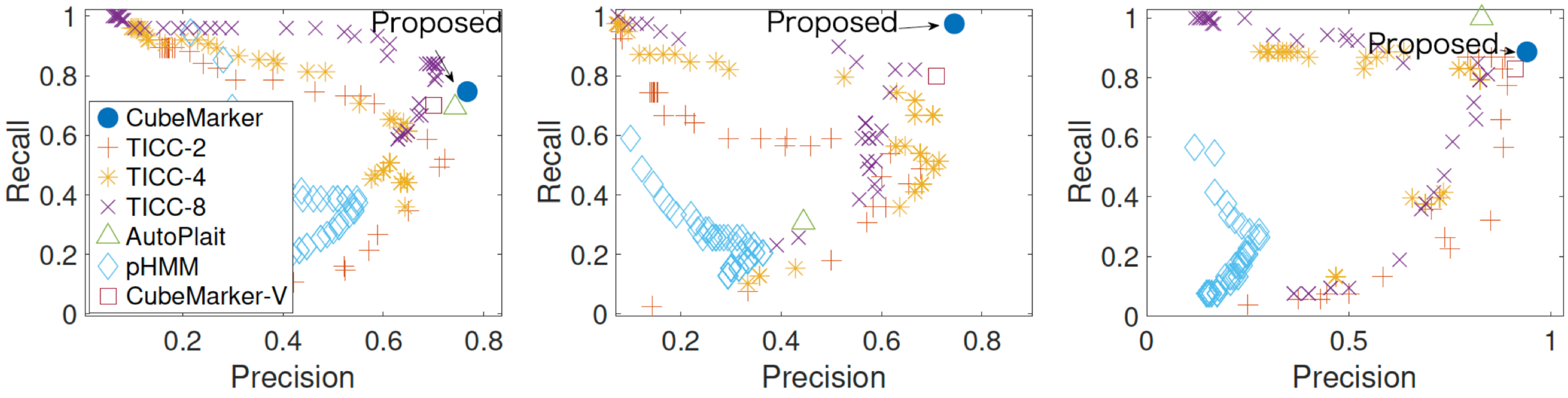


Wall clock time v.s. dataset size for (#1) Workout (*O(dwn)*)



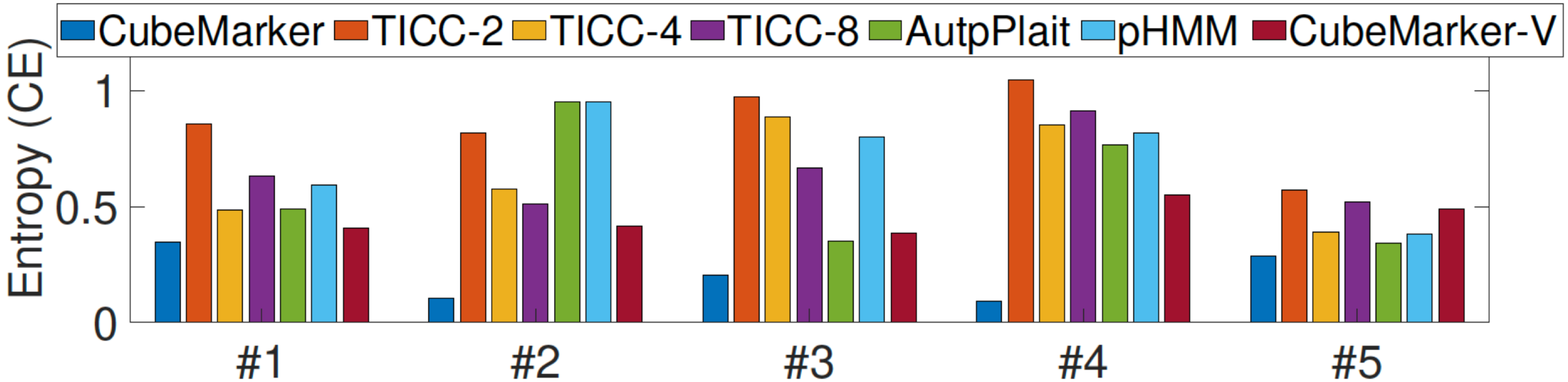
Wall clock time for each dataset (1700x faster than pHMM)

Q3. Accuracy (segment/regime)



(a) *(#1) Workout* (b) *(#2) Tennis* (c) *(#3) Factory*

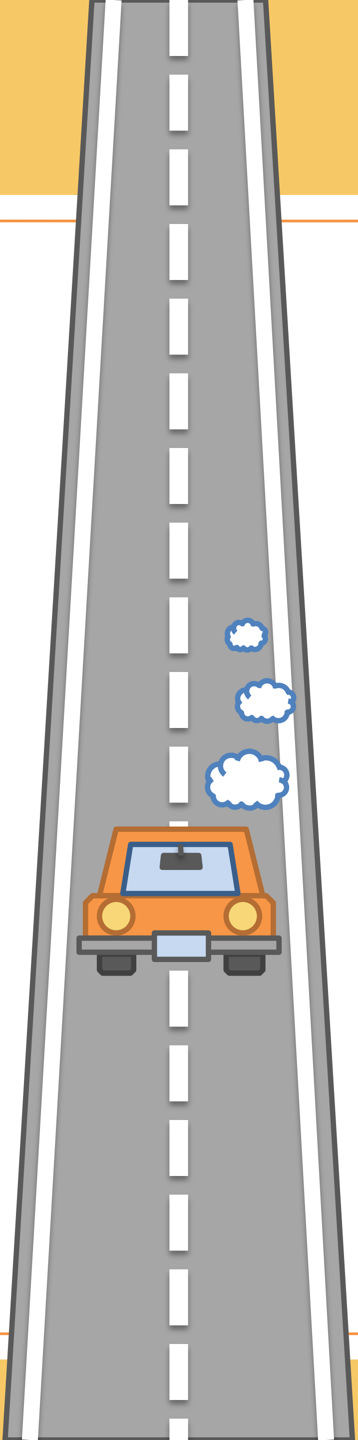
Segmentation accuracy (top righter is better)



Regime clustering accuracy (lower is better)

Outline

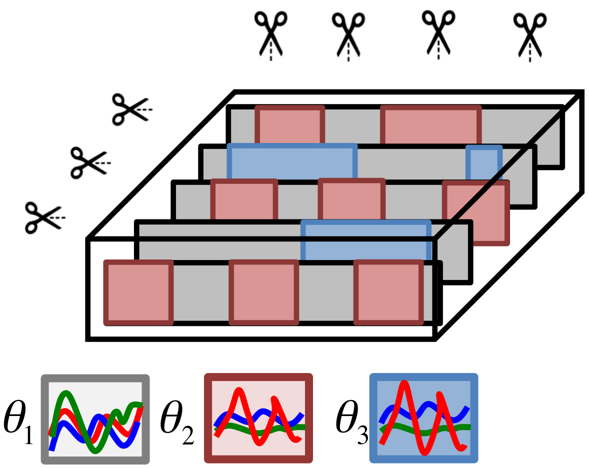
- Motivation
- Problem definition
- Main ideas
- Algorithms
- Experiments
- **Conclusions**



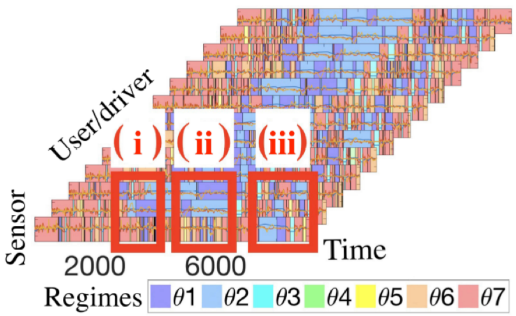
Conclusions

Our method has the following properties:

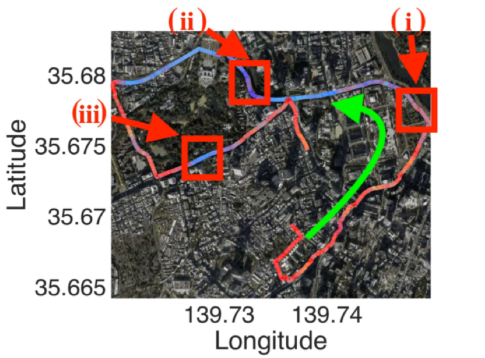
- **Effective**
  Find multi-aspect segments/regimes
- **Automatic**
  No magic numbers
- **Scalable**
  It scales linearly to the data size



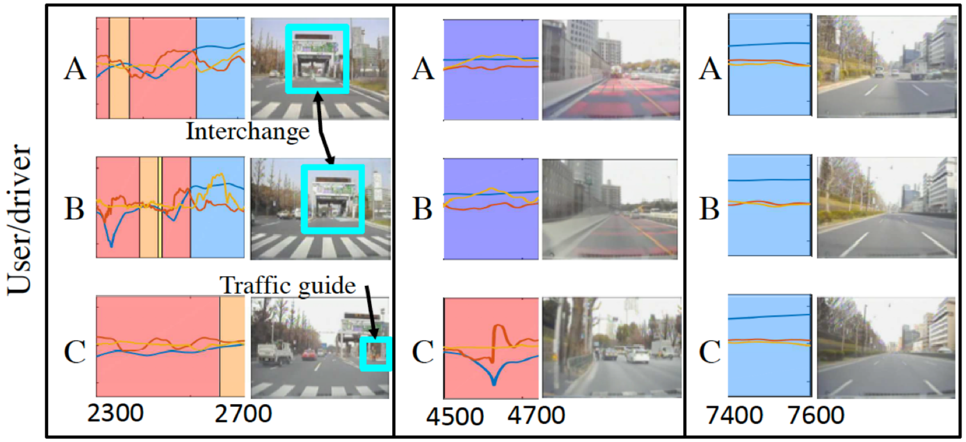
Thank you!



(a) Multi-aspect segmentation and summarization



(b) Representative driving behavior on a map



(c-i) Interchange (c-ii) Expressway (c-iii) Wide road

(c) User/driver-specific behavior at three different locations