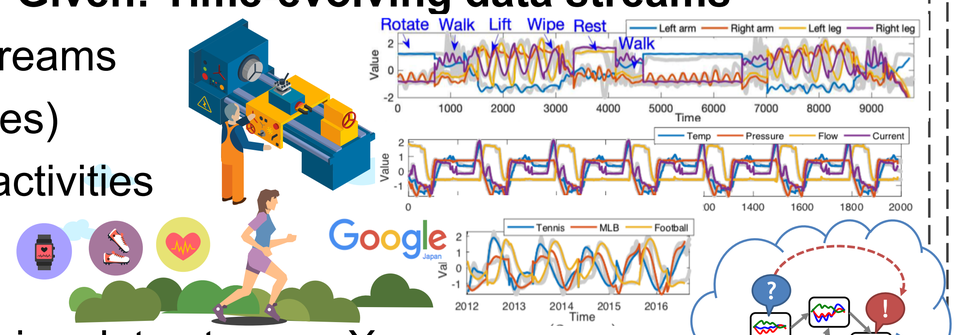
ACM SIGKDD 2019
25nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining
August 4 - 8, 2019, Anchorage, Alaska USA

Dynamic Modeling and Forecasting of Time-evolving Data Streams

Yasuko Matsubara, Yasushi Sakurai
Artificial Intelligence Research Center, ISIR, Osaka University
{yasuko, yasushi}@sanken.osaka-u.ac.jp

Motivation - Given: Time-evolving data streams

- IoT/sensor streams (smart factories)
- Web, online activities



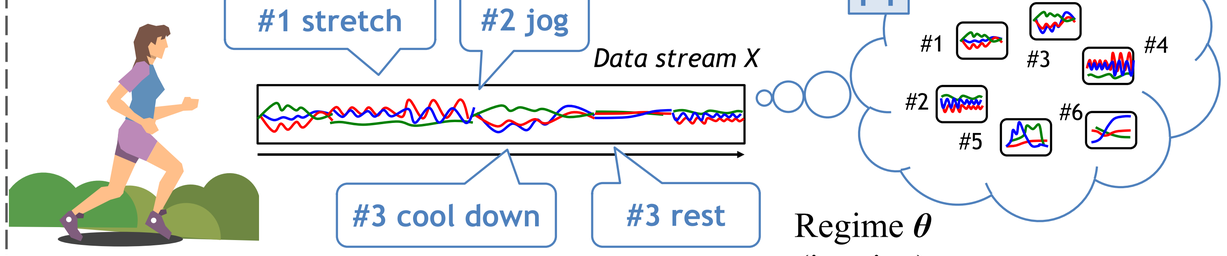
Given: Co-evolving data streams X
Forecast ls-steps-ahead future value

- **Find** major patterns / **regimes**
- **Find dynamic space transitions** between regimes
- **Report ls-steps-ahead future** value (i.e., $\boldsymbol{e}(t_c+t_s)$)

Proposed model - OrbitMap

Main idea P1 : **Regimes** (i.e., time-evolving patterns)

Various patterns ("**regimes**") in streams

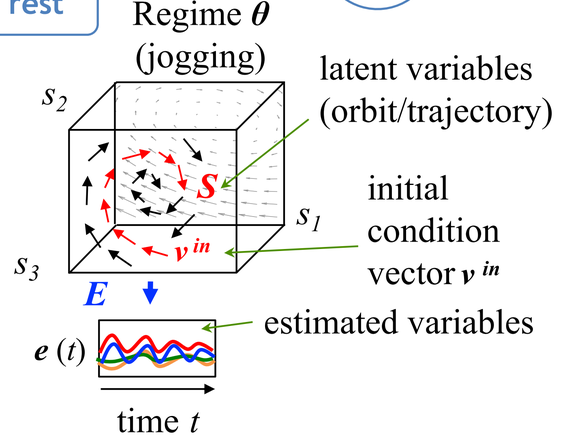


Exponential / Linear / Non-linear

$$\frac{d\boldsymbol{s}(t)}{dt} = \boldsymbol{p} + \mathrm{Q}\boldsymbol{s}(t) + \mathcal{A}\mathrm{S}(t)$$
$$\boldsymbol{e}(t) = \boldsymbol{u} + \mathrm{V}\boldsymbol{s}(t)$$

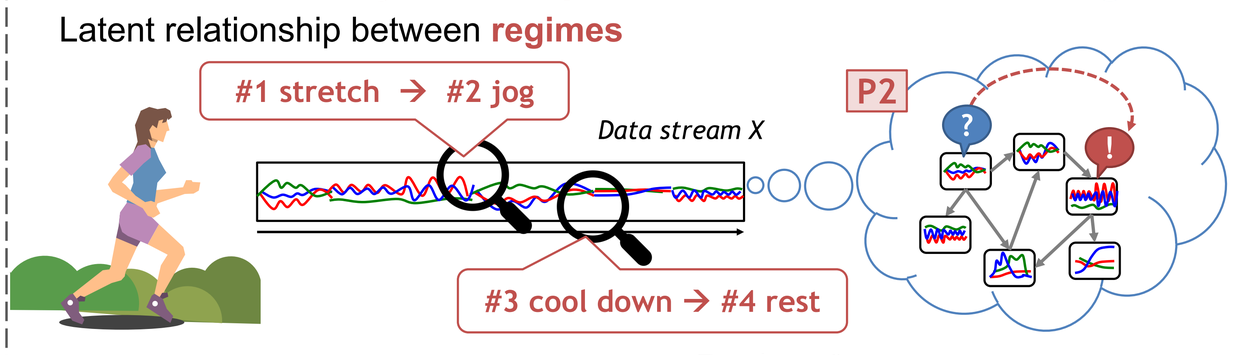
Projection

$$\boldsymbol{s}(1) = \boldsymbol{v}^{in} \quad \theta = \{\boldsymbol{p}, \mathrm{Q}, \mathcal{A}, \boldsymbol{u}, \mathrm{V}\}$$



Main idea P2 : **Dynamic space transition** between **regimes**

Latent relationship between **regimes**



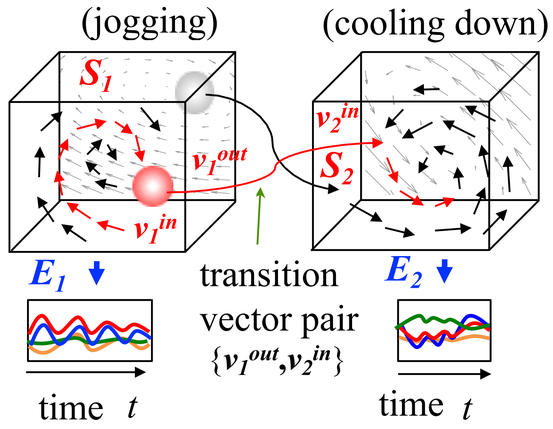
Staying in Regime i

$$\boldsymbol{s}(t) = \begin{cases} \boldsymbol{s}_i(t) & (1 \le t < t_s) \\ \boldsymbol{v}_j^{in} & (t = t_s) \\ \boldsymbol{s}_j(t) & (t_s < t) \end{cases}$$

Transition from Regime i to j

Staying in Regime j

$$t_s = \underset{t \mid D(t) \le \rho}{\arg\min}\ D(t),$$
$$D(t) = ||\boldsymbol{s}_i(t) - \boldsymbol{v}_i^{out}||$$

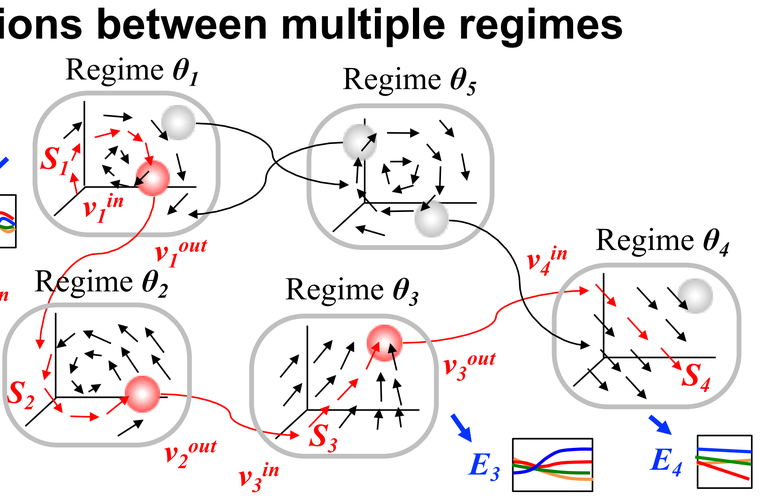


Dynamic space transitions between multiple regimes

regimes $\theta_1 \rightarrow \theta_2 \rightarrow \theta_3 \rightarrow \theta_4$

$$\mathcal{M} = \{\Theta, \mathcal{V}\}$$
$$\Theta = \{\theta_1, \cdots, \theta_r\}$$
$$\{\boldsymbol{v}_i^{out}, \boldsymbol{v}_j^{in}\} \in \mathcal{V}$$

Estimated variables **E**



URL : http://www.dm.sanken.osaka-u.ac.jp/~yasuko/

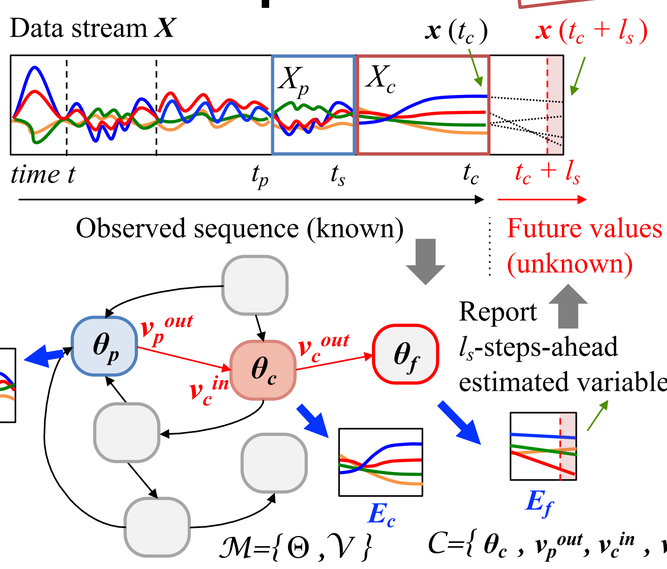
Streaming algorithm – OrbitMap-F

Details in paper

Given:
- data stream
  $X = \{\boldsymbol{x}(1), \ldots, \boldsymbol{x}(t_c)\}$

Estimate:
- Model parameter set
  $\mathcal{M} = \{\Theta, \mathcal{V}\}$
- Model candidate
  $C = \{\theta_c, \boldsymbol{v}_p^{out}, \boldsymbol{v}_c^{in}, \boldsymbol{v}_c^{out}\}$

Report:
- l_s-steps-ahead future value
  $\boldsymbol{e}(t_c + l_s)$



OrbitMap-F (Algorithms)

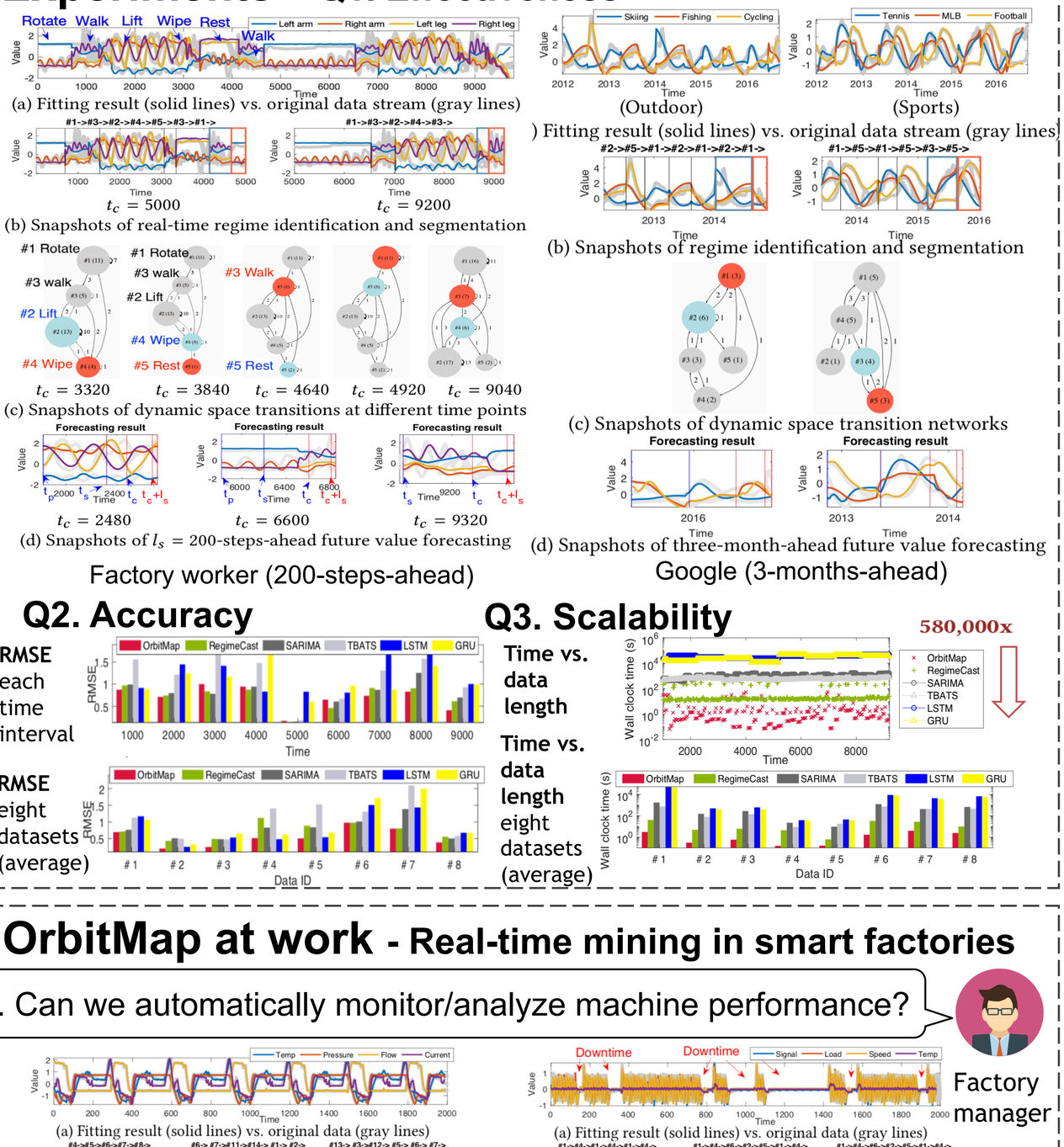
O-Estimator - Estimates model parameters $\mathcal{M} = \{\Theta, \mathcal{V}\}$ and model candidate $C = \{\theta_c, \boldsymbol{v}_p^{out}, \boldsymbol{v}_c^{in}, \boldsymbol{v}_c^{out}\}$

O-Generator - Generates ls-steps-ahead future values $\boldsymbol{e}(t_c + l_s)$

O-Feedback - Cleans up useless models in $\mathcal{M}$

Scalability (OrbitMap) : at least O(1), at most (r), r : # or regimes in *M*

Experiments – Q1. Effectiveness



Q2. Accuracy

RMSE each time interval

RMSE eight datasets (average)

Q3. Scalability

Time vs. data length

Time vs. data length eight datasets (average)

580,000x

OrbitMap at work - Real-time mining in smart factories

Q. Can we automatically monitor/analyze machine performance?

Conclusions - OrbitMap has following advantages:

Effective : long-term forecasting / no prior training
General & Scalable: it matches diverse real data / any-time processing