Regime Shifts in Streams: Real-time Forecasting of Co-evolving Time Sequences

Yasuko Matsubara, Yasushi Sakurai (Kumamoto University)
Big time-series data streams

Social/natural phenomena
- Climate
- Economy
- Web
- Epidemic

Physical sensors

Big Data

KDD 2016 Y. Matsubara & Y. Sakurai
Big time-series data streams

Social/natural phenomena

climate

Economy

Epidemic

Motion sensors

L/R legs

L/R arms

Physical sensors

(walking)
Big time-series data streams

Social/natural phenomena

climate

Online activities

Amazon P
Netflix
YouTube
Hulu

Physical sensors

Economy

Epidemic

KDD 2016
Y. Matsubara & Y. Sakurai
Big time-series data streams

Social/natural phenomena

Q. Can we forecast future events?
Big time-series data streams

- **Given:**
  Co-evolving event stream
  \[ X = \{ x(1), x(2), \ldots, x(t_c), \ldots \} \]

- **Goal:**
  Forecast \( l_s \)-steps-ahead future events, at any point in time
Roadmap

✔ Motivation
- Forecasting power of RegimeCast
- Overview
- Proposed model
- Streaming algorithm
- Experiments
- Conclusions
Forecasting power of RegimeCast

Real-time forecasting over data streams
Forecasting power of RegimeCast

Real-time forecasting over data streams

Original
Forecasting power of RegimeCast

Real-time forecasting over data streams

Original

Forecast
(100-steps-ahead)
Forecasting power of RegimeCast

Real-time forecasting over data streams

Original

Forecast
(100-steps-ahead)

Snap-Shot
(Current window)
Forecasting power of RegimeCast

Real-time forecasting over data streams

Original

Forecast
(100-steps ahead)

Snap-Shot
(Current window)

Arrived events

Future events

?
Forecasting power of RegimeCast

Real-time forecasting over data streams

Forecast (100-steps ahead)

Snap-Shot (Current window)

Original

Arrived events

Future events

KDD 2016 Y. Matsubara & Y. Sakurai
Forecasting power of RegimeCast

Real-time forecasting over data streams

1. Original
2. Forecast (100-steps ahead)
3. Snap-Shot (Current window)
Roadmap

✔ Motivation
✔ Forecasting power of RegimeCast
  - Overview
  - Proposed model
  - Streaming algorithm
  - Experiments
  - Conclusions
Overview

What is “Real-time forecasting”?

(a) $l_s$-steps-ahead forecasting

- Long-term
- Continuous

(b) Adaptive non-linear modeling

- Non-linear
- Adaptive
(a) $l_s$-steps-ahead forecasting

**Long-term**: Predict $l_s$-steps ahead events

**Continuous**: Capture dynamic patterns

```
<table>
<thead>
<tr>
<th>X</th>
<th>Arrived events</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Future events</td>
</tr>
<tr>
<td></td>
<td>$l_s$-steps ahead events</td>
</tr>
</tbody>
</table>
```

KDD 2016
Y. Matsubara & Y. Sakurai
(b) Adaptive non-linear modeling

Non-linear: Non-linear dynamical systems

Adaptive: Regime shifts (ecosystems)

\[
\frac{ds(t)}{dt}
\]

NLDSs

Woodlands

Grasslands

Image courtesy of dan at FreeDigitalPhotos.net.
Motivation

Forecasting power of RegimeCast

Overview

- Proposed model
- Streaming algorithm
- Experiments
- Conclusions
Proposed model

Main ideas

P1 Latent non-linear dynamics

P2 Regime shifts in streams

P3 Nested structure
Proposed model

Main ideas

P1  Latent non-linear dynamics

P2  Regime shifts in streams

P3  Nested structure
Latent non-linear dynamics

Various patterns ("regimes") in streams

- Walking
- Stretching
- (Right)

KDD 2016 Y. Matsubara & Y. Sakurai
Various patterns ("regimes") in streams

Q. How can we effectively capture dynamics of "regimes"?
A. Latent NLDS

\[ \frac{ds(t)}{dt} = p + Qs(t) + AS(t) \]

\[ v(t) = u + Vs(t) \]
A. Latent NLDS

\[ \frac{ds(t)}{dt} = p + Qs(t) + As(t) \]
\[ v(t) = u + Vs(t) \]

* S(0) = s_0

\[ \theta = \{ s_0, p, Q, A, u, V \} \]
Proposed model

Main ideas

P1: Latent non-linear dynamics

P2: Regime shifts in streams

P3: Nested structure
Various patterns ("regimes") in streams

walking

stretching (left) (both)
Various patterns ("regimes") in streams

Walking

Stretching (left) (both)

Regime #1 "Walk"
Regime shifts in streams

Various patterns ("regimes") in streams

walking

stretching (left) (both)

Regime #1
“Walk”

change

Regime #2
“Stretch”
Q. How can we identify sudden discontinuities?

Regime #1
“Walk”

Regime #2
“Stretch”

Various patterns (“regimes”) in streams
walking (left) stretching (both)

KDD 2016
Y. Matsubara & Y. Sakurai
Q. How can we identify sudden discontinuities?

A: “Regime Shifts in Streams”!
Regime shifts in natural systems

Abrupt changes in the structure of complex systems

Examples:
• Woodland vs. grassland
• Coral vs. macro algae
• Desert vs. vegetation

Ecological system

Image courtesy of dan at FreeDigitalPhotos.net.
Abrupt changes in the structure of complex systems

\[ \frac{ds(t)}{dt} = a_0 + a_1 s(t) + a_2 f(s(t)) \]

**Ecological system**

- **Woodlands**
- **Grasslands**

**Time-evolving ecosystem property** (nutrients/soils)

- \( a_0 \): environmental factor
- \( a_1 \): growth/decay rate
- \( a_2 \): recover rate

Image courtesy of dan at FreeDigitalPhotos.net.
Regime shifts in event streams

Abrupt changes in the structure of complex systems

Ecological system

Motion sensors

Image courtesy of dan at FreeDigitalPhotos.net.
Regime shifts in event streams

L-NLDS + regime activity

\[
\frac{ds_i(t)}{dt} = p_i + Q_i s_i(t) + A_i S_i(t) \quad (i = 1, \ldots, c)
\]

\[
\frac{dw(t)}{dt} = r(t), \quad R = \{ r(t) \}_{t=1}^{tc}
\]

\[
v(t) = \sum_{i=1}^{c} w_i(t) [u_i + V_i s_i(t)]
\]

\[
\Theta = \{ \theta_1, \ldots, \theta_c, R \}
\]

R: Regime shift dynamics

c: # of regimes
Proposed model

Main ideas

P1 Latent non-linear dynamics

P2 Regime shifts in streams

P3 Nested structure
Nested structure

Nested, multi-scale dynamical activities

Chicken dance
Nested structure

Nested, multi-scale dynamical activities

beaks  wings  tail feathers  claps

Chicken dance

\( X_{\text{org}} \)
Nested structure

Nested, multi-scale dynamical activities

Original events:

\[ X^{(1)} : \text{Long-term} \]
\[ X^{(2)} : \text{Short-term} \]

Chicken dance
Nested, multi-scale dynamical activities

$X_{org} = X^{(1)} + X^{(2)}$

Tail feathers = bending knees, once + moving arms, quickly
Nested structure

Multi-level modeling structure

\[ \Theta^{(1)} \]
\[ \Theta^{(2)} \]

Local events

\[ V^{(1)}_E \]
\[ V^{(2)}_E \]

Estimated events \( V_E \)

\[ \approx \]

Full parameter set \( \mathcal{M} \)

\[ \mathcal{M} = \{ \Theta^{(1)}, \ldots, \Theta^{(h)} \} \]
Roadmap

- Motivation
- Forecasting power of RegimeCast
- Overview
- Proposed model
  - Streaming algorithm
- Experiments
- Conclusions
Problem definition

- **RegimeSnap**

![Diagram showing current and forecast windows with event streams and future events](image)

- Event stream $X$
- Estimated events $V_E$
- Forecast window $V_F$
- Current window $X_C$

Time: $t_m$, $t_c$, $t_s$, $t_e$

- Arrived events
- Future (unknown) events
Problem definition

- **RegimeSnap**

Given:
- Current window $X_C$
- (original events)

Future (unknown) events
Problem definition

- **RegimeSnap**

Find:

Estimated events $V_E$

Current window

Event stream $X$

Estimated events $V_E$

Arrived events

Future (unknown) events
Problem definition

- **RegimeSnap**

**Report:**

Forecast window \( V_F \)

(\( l_s \)-steps-ahead)

- Event stream \( X \)
- Estimated events \( V_E \)

Current window

Arrived events

Future (unknown) events
Streaming algorithm

- Proposed algorithms

1. **RegimeCast**
   - Report $l_s$-steps-ahead future events

2. **RegimeReader**
   - Identify current regime dynamics

3. **RegimeEstimator**
   - Estimates regime parameter set $\theta$
Event stream $X$

Model

Regime

Estimator

Regime

Reader

Forecast

window

Report

$V_E^{(1)} + V_E^{(2)} \approx \cdots$
RegimeCast

Event stream $X$

Time $t_c$

$X_c$

Step 1: Extract current window

Model

$\theta_1^{(1)}$ $\theta_2^{(1)}$ $\theta_1^{(2)}$ $\theta_2^{(2)}$

$\Theta^{(1)}$ $\Theta^{(2)}$

$V_E$ $V_F$

Report

Forecast window
Step 2: Find optimal regimes

Model DB

Regime Reader

Regime Estimator

Report window

VE

VF

VE

θ1(1)

θ2(1)

θ1(2)

θ2(2)

RegimeCast

KDD 2016

Y. Matsubara & Y. Sakurai
Step 2: Find optimal regimes

- Update model parameters $\theta_1^{(1)}, \ldots$
- Identify regime shift dynamics $r(t_c)$
Step3: (optional) Estimate/insert new regime $\theta$

RegimeEstimator

Model DB

Event stream $X_C$

$\Theta^{(1)}$

$\Theta^{(2)}$

$\theta^{(1)}_1$

$\theta^{(1)}_2$

$\theta^{(2)}_1$

$\theta^{(2)}_2$
RegimeCast

Step 4: Estimate future events

Estimated local events:

\[ V_E^{(1)} + V_E^{(2)} \approx \ldots \]

Model DB

Regime Reader

Forecast window

Report

Event stream

Time

\( t_c \)

\( X_C \)

\( V_E \)

KDD 2016 Y. Matsubara & Y. Sakurai
RegimeCast

Step 5: Report future events

Report

Forecast window $V_F$

$V_{F}$
RegimeCast

Event stream $X$

Time $t_c$

Report

Forecast window

Model DB

Regime Estimator

$\theta_1^{(1)}$ $\theta_2^{(1)}$

$\theta_1^{(2)}$ $\theta_2^{(2)}$

$\Theta^{(1)}$ $\Theta^{(2)}$

$X_c$

$V_F$

$V_E$

$V_E^{(1)}$ $V_E^{(2)}$

$\approx$

$\theta_1$ $(1)$ $\theta_2$ $(1)$

$\theta_1$ $(2)$ $\theta_2$ $(2)$
Efficient event generation

Q. How can we efficiently generate events?

A. Dynamic point set (DPS)

Scalability (RegimeCast)

at least $O(c \cdot l_e / \delta)$ at most $O(c \cdot l_e / \delta + l_c)$

- $c$: # of regimes
- $l_e$: Length of $V_E$
- $\delta$: DPS interval
- $l_c$: Length of $X_C$

* It does not depend on data length $t_c$
Roadmap

- Motivation
- Forecasting power of RegimeCast
- Overview
- Proposed model
- Streaming algorithm
  - Experiments
  - Conclusions
Experiments

We answer the following questions...

**Q1. Effectiveness**
How successful is it in forecasting events?

**Q2. Accuracy**
How well does it forecast future events?

**Q3. Scalability**
How does it scale in terms of computational time?
Q1. Effective - MoCap #1

“Exercise”

(a) Original event stream (top) and (100:120)-steps-ahead forecasting results (bottom)

(1) Forecasted variables
(2) Forecasted result
(3) Original data
(4) Walking

(b-1) Stretch/left ($t_c = 2240$)
(b-2) Stretch/right ($t_c = 3100$)
(b-3) Stretch/both ($t_c = 4200$)
(b-4) Walking ($t_c = 4980$)

KDD 2016 Y. Matsubara & Y. Sakurai 59
“House cleaning”

(a) Original data stream (top) and our real-time forecasted result (bottom)

(b-1) Walking ($t_c = 560$)
(b-2) Dustpan ($t_c = 4020$)
(b-3) Wipe a window ($t_c = 6760$)
(b-4) Walking ($t_c = 8680$)

(100-120)-steps-ahead forecasted variables
Q1. Effective - MoCap #3

"Chicken dance"

Effective steps ahead

(c-1) $t_c = 540$
(c-2) $t_c = 1044$
(c-3) $t_c = 1116$
Q1. Effective - Google Trend

3-months ahead

(a) Online TV

(b) Beers
Q1. Effective - Google Trend

3-months ahead
Q1. Effective - others

Atmospheric pressure & temperature

3-months ahead
Q1. Effective - others

Yen vs. dollar & AU vs. PT

6-weeks ahead

3-months ahead
Q2. Accuracy

Forecasting results of RegimeCast vs. others

Original stream

(a) Original event stream (top) and (100:120)-steps-ahead forecasting results (bottom)

Regime Cast

ARIMA

TBATS
Q2. Accuracy

Forecasting results of RegimeCast vs. others

RegimeCast can identify regime-shift dynamics, immediately.
Q2. Accuracy

Forecasting error (RMSE), lower is better

(a) Forecasting error for each time tick (left) and average (right)
Q3. Scalability

(a) “Exercise”

(b) “House-cleaning”

RegimeEstimator

ARIMA

TBATS

Regime Cast
Q3. Scalability

RegimeCast

ARIMA

TBATS

Up to 270x faster than ARIMA/TBATS

(a) “Exercise”

(b) “House-cleaning”
Q. How long ahead can it forecast?
Q. Discussion

Q. How long ahead can it forecast?

$l_s$-steps vs. error

$l_s$-steps vs. speed
Q. How long ahead can it forecast?

A. It can forecast future events for every step $l_s$.
Roadmap

- Motivation
- Forecasting power of RegimeCast
- Overview
- Proposed model
- Streaming algorithm
- Experiments
- Conclusions
Conclusions

RegimeCast has the following advantages

- **Effective**
  - Long-term forecasting

- **Adaptive**
  - No prior training

- **Scalable**
  - It does not depend on data length

- **Anytime**
  - It forecasts future events, immediately, any time
Thank you!

Data & Code: http://www.cs.kumamoto-u.ac.jp/~yasuko
Regime Shifts in Streams: Real-time Forecasting of Co-evolving Time Sequences

Yasuko Matsubara, Yasushi Sakurai
(Kumamoto University)