AutoPlait: Automatic Mining of Co-evolving Time Sequences

Yasuko Matsubara (Kumamoto University)
Yasushi Sakurai (Kumamoto University)
Christos Faloutsos (CMU)
Motivation

Given: co-evolving time-series
– e.g., MoCap (leg/arm sensors)

“Chicken dance”
Motivation

Given: co-evolving time-series
– e.g., MoCap (leg/arm sensors)

“Chicken dance”

Q. Any distinct patterns?

Q. If yes, how many?

Q. What kind?
Motivation

Challenges: co-evolving sequences

- Unknown # of patterns (e.g., beaks)
- Different durations
Motivation

Challenges: co-evolving sequences

Q. Can we summarize it *automatically*?
Motivation

**Goal:** find patterns that agree with human intuition

![Diagram showing time series data with labels for left/right legs and arms](image_url)
Motivation

Goal: find patterns that agree with human intuition

AutoPlait: “fully-automatic” mining algorithm
Importance of “fully-automatic”

No magic numbers! … because,

Manual
- sensitive to the parameter tuning
- long tuning steps (hours, days, ...)

Automatic (no magic numbers)
- no expert tuning required

Big data mining:

-> we cannot afford human intervention!!
Outline

- Motivation
- Problem definition
- Compression & summarization
- Algorithms
- Experiments
- AutoPlait at work
- Conclusions
Problem definition

Key concepts

- **Bundle**: $X$ (given)
- **Segment**: $S$ (hidden)
- **Regime**: $\Theta$ (hidden)
- **Segment-membership**: $F$ (hidden)
Problem definition

- **Bundle**: set of \( d \) co-evolving sequences

\[
X = \{x_1, \ldots, x_n\}
\]

\( d \times n \) given
Problem definition

- **Segment**: convert $X \rightarrow m$ segments, $S$

\[ S = \{s_1, \ldots, s_m\} \]
• **Regime**: segment groups: \( \Theta = \{ \theta_1, \theta_2, \ldots, \theta_r, \Delta_{r \times r} \} \)

\( \theta_r \) : model of regime \( r \)

Regimes
(r=4)

\( \theta_1 \)
\( \theta_2 \)
\( \theta_3 \)
\( \theta_4 \)

beaks
wings

**Problem definition**
Problem definition

- Segment-membership: assignment

\[ F = \{ f_1, \ldots, f_m \} \]

Segment-membership (m=8)
Problem definition

• Given: bundle $X$

$$X = \{x_1, \ldots, x_n\}$$
Problem definition

• Given: bundle $X$

\[ X = \{x_1, \ldots, x_n\} \]

• Find: compact description $C$ of $X$

\[ C = \{m, r, S, \Theta, F\} \]
Problem definition

• Given: bundle $X$

\[ X = \{ x_1, \ldots, x_n \} \]

• Find: compact description $C$ of $X$

\[ C = \{ m, r, S, \Theta, F \} \]

$m$ segments

$r$ regimes

Segment-membership
Outline

- Motivation
- Problem definition
- Compression & summarization
- Algorithms
- Experiments
- AutoPlait at work
- Conclusions
Main ideas

Goal: compact description of $X$

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

without user intervention!!

Challenges:

Q1. How to generate ‘informative’ regimes?

Q2. How to decide \# of regimes/segments?
Main ideas

Goal: compact description of $X$

$$C = \{m, r, S, \Theta, F\}$$
without user intervention!!

Challenges:

Q1. How to generate ‘informative’ regimes?
   Idea (1): Multi-level chain model

Q2. How to decide # of regimes/segments?
   Idea (2): Model description cost

SIGMOD 2014 Y. Matsubara et al. 20
Idea (1): MLCM: multi-level chain model

Q1. How to generate ‘informative’ regimes?
Idea (1): MLCM: multi-level chain model

Q1. How to generate ‘informative’ regimes?

Idea (1): Multi-level chain model
- HMM-based probabilistic model
- with “across-regime” transitions
Idea (1): MLCM: multi-level chain model

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_r, \Delta_{r \times r}\}$$

$$(\theta_i = \{\pi, A, B\})$$

r regimes (HMMs) across-regime transition prob. Single HMM parameters
Idea (1): MLCM: multi-level chain model

\[ \Theta = \{\theta_1, \theta_2, \ldots, \theta_r, \Delta_{r \times r}\} \]

\[ (\theta_i = \{\pi, A, B\}) \]

- \( r \) regimes (HMMs)
- across-regime transition prob.
- Single HMM parameters

Regimes
r=2
Regime 1 (k=3)
Regime 2 (k=2)

Regime1 “beaks”
Regime2 “wings”
Idea (2): model description cost

Q2. How to decide \# of regimes/segments?

Idea (2): Model description cost
- Minimize coding cost
- find “optimal” \# of segments/regimes
Idea (2): model description cost

Idea: Minimize encoding cost!

\[
\min \left( Cost_M(M) + Cost_c(X|M) \right)
\]

Model cost
Coding cost

Good compression
Good description

CostM
CostC
CostT(# of r, m)
Idea (2): model description cost

Total cost of bundle $X$, given $C$

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

\[
\begin{align*}
Cost_T(X; C) &= Cost_T(X; m, r, S, \Theta, F) \\
&= \log^*(n) + \log^*(d) + \log^*(m) + \log^*(r) + m \log(r) \\
&\quad + \sum_{i=1}^{m-1} \log^*|s_i| + Cost_M(\Theta) + Cost_C(X|\Theta) \quad (6)
\end{align*}
\]
Idea (2): model description cost

Total cost of bundle X, given C

\[ C = \{m, r, S, \Theta, F\} \]

\[
\begin{align*}
\text{Cost}_T(X; C) &= \text{Cost}_T(X; m, r, S, \Theta, F) \\
&= \log^*(n) + \log^*(d) + \log^*(m) + \log^*(r) + m \log(r) \\
&+ \sum_{i=1}^{m-1} \log^*|s_i| + \text{Cost}_M(\Theta) + \text{Cost}_C(X|\Theta)
\end{align*}
\]

- Duration/dimensions
- # of segments/ regimes
- Segment-membership F
- Segment lengths
- Model description cost of \( \Theta \)
- Coding cost of X given \( \Theta \)
Outline

- Motivation
- Problem definition
- Compression & summarization
- Algorithms
- Experiments
- AutoPlait at work
- Conclusions
AutoPlait

Overview

Iteration 0
r=1, m=1

Start!

Cost

r=1

Iteration
Iteration 1
r=2, m=4

\begin{align*}
  f_1 &= 2 \\
  f_2 &= 1 \\
  f_3 &= 2 \\
  f_4 &= 1 \\
\end{align*}
AutoPlait

Overview

Iteration 1
r=2, m=4

\[ f_1 = 2 \]
\[ f_2 = 1 \]
\[ f_3 = 2 \]
\[ f_4 = 1 \]

\[ \theta_1 \]
\[ \theta_2 \]

Iteration 2
r=3, m=6

\[ f_1 = 2 \]
\[ f_2 = 3 \]
\[ f_3 = 1 \]
\[ f_4 = 2 \]
\[ f_5 = 3 \]
\[ f_6 = 1 \]

\[ \theta_1 \]
\[ \theta_3 \]
\[ \theta_2 \]

Split

r=1

r=2

r=3

Cost

Iteration

SIGMOD 2014

Y. Matsubara et al.

32
AutoPlait

Overview

Iteration 1
\( r=2, m=4 \)

Iteration 2
\( r=3, m=6 \)

Iteration 4
\( r=4, m=8 \)

\( f_1 = 2 \quad f_2 = 1 \quad f_3 = 2 \quad f_4 = 1 \)

\( f_1 = 2 \quad f_2 = 3 \quad f_3 = 1 \quad f_4 = 2 \quad f_5 = 3 \quad f_6 = 1 \)

\( f_1 = 2 \quad f_2 = 4 \quad f_3 = 3 \quad f_4 = 1 \quad f_5 = 2 \quad f_6 = 4 \quad f_7 = 3 \quad f_8 = 1 \)
AutoPlait

Algorithms

1. **CutPointSearch**
   - Inner-most loop
   - Find good cut-points/segments

2. **RegimeSplit**
   - Inner loop
   - Estimate good regime parameters $\Theta$

3. **AutoPlait**
   - Outer loop
   - Search for the best number of regimes ($r=2,3,4...$)
1. CutPointSearch

Given:
- bundle $X$
- parameters of two regimes $\Theta = \{\theta_1, \theta_2, \Delta\}$

Find: **cut-points** of segment sets $S_1, S_2$, 
\[
\{S_1, S_2\} = \text{argmax } P(X \mid S_1, S_2, \Theta)
\]

$S_1 = \{s_2, s_4\}$
$S_2 = \{s_1, s_3\}$
1. CutPointSearch

DP algorithm to compute likelihood:

\[ P(X | \Theta) \]

\[ X \]

\[ t = 1 \quad t = 2 \quad t = 3 \quad t = 4 \quad t = 5 \quad t = 6 \]

\[ \theta_1 \]

\[ \theta_2 \]
1. CutPointSearch

Theoretical analysis

Scalability
- It takes $O(ndk^2)$ time (only single scan)
  - n: length of X
  - d: dimension of X
  - k: # of hidden states in regime

Accuracy
- It guarantees the optimal cut points
  - (Details in paper)
2. Regime Split

**Given:**
- bundle $X$

**Find:** two regimes

1. find cut-points of segment sets: $S_1, S_2$
2. estimate parameters of two regimes:

$$\Theta = \{\theta_1, \theta_2, \Delta\}$$
Two-phase iterative approach

- **Phase 1**: (CutPointSearch)
  - Split segments into two groups: \( S_1, S_2 \)

- **Phase 2**: (BaumWelch)
  - Update model parameters: \( \Theta = \{\theta_1, \theta_2, \Delta\} \)

\[
\begin{align*}
S_1 &= \{s_2, s_4\} \\
S_2 &= \{s_1, s_3\} \\
\{\theta_1, \theta_2, \Delta\}
\end{align*}
\]
3. AutoPlait

Given:
- bundle $X$

Find:
- $r$ regimes ($r=2, 3, 4, \ldots$)
- i.e., find full parameter set

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

Split regimes $r=2,3,...$, as long as cost keeps decreasing
- Find appropriate # of regimes

\[ r = \min_r \text{Cost}_T(X; m, r, S, \Theta, F) \]
3. AutoPlait

Split regimes $r=2, 3, \ldots$, as long as cost keeps decreasing
- Find appropriate # of regimes

$$r = \min_{r} \text{Cost}_T(X; m, r, S, \Theta, F)$$

**r=2, m=4**

$$f_1 = 2 \quad f_2 = 1 \quad f_3 = 2 \quad f_4 = 1$$

**r=4, m=8**

$$f_1 = 2 \quad f_2 = 4 \quad f_3 = 3 \quad f_4 = 1 \quad f_5 = 2 \quad f_6 = 4 \quad f_7 = 3 \quad f_8 = 1$$

**r=3, m=6**

$$f_1 = 2 \quad f_2 = 3 \quad f_3 = 1 \quad f_4 = 2 \quad f_5 = 3 \quad f_6 = 1$$
Outline

- Motivation
- Problem definition
- Compression & summarization
- Algorithms

- Experiments
- AutoPlait at work
- Conclusions
Experiments

We answer the following questions...

**Q1. Sense-making**
Can it help us understand the given bundles?

**Q2. Accuracy**
How well does it find cut-points and regimes?

**Q3. Scalability**
How does it scale in terms of computational time?
Q1. Sense-making

MoCap data

AutoPlait (NO magic numbers)

DynaMMo (Li et al., KDD’09)

pHMM (Wang et al., SIGMOD’11)
Q1. Sense-making

MoCap data

AutoPlait (NO magic numbers)
Q2. Accuracy

(a) Segmentation

(b) Clustering

(a) Precision and recall (higher is better)  (b) CE score (lower is better)

AutoPlait needs “no magic numbers”
Q3. Scalability

Wall clock time vs. data size (length) : $n$

AutoPlait scales linearly, i.e., $O(n)$
Outline

- Motivation
- Problem definition
- Compression & summarization
- Algorithms
- Experiments
- AutoPlait at work
- Conclusions
AutoPlait at work

AutoPlait is capable of various applications, e.g.,

**App1. Model analysis**
- Web-click sequences

**App2. Event discovery**
- Google Trend data
AutoPlait at work

AutoPlait is capable of various applications, e.g.,

**App1. Model analysis**
- Web-click sequences

**App2. Event discovery**
- Google Trend data
Web-click sequences (1 month, 5 urls)

- 5 urls: blog, news, dictionary, Q&A, mail
- every 10 minutes
App1. Model analysis (WebClick)

Web-click sequences (1 month, 5 urls)

AutoPlait finds 2 patterns: weekday/weekend!
AutoPlait finds 2 patterns: weekday (but holiday)
Pattern of **weekday regime**

![Graph showing usage patterns over time](image)

**Observation:** Working hard every weekday (i.e., using dictionary, news sites)
Pattern of weekend regime

Observation: No more work on weekend (i.e., blog, mail, Q&A for non-business purposes)
AutoPlait at work

AutoPlait is capable of various applications, e.g.,

**App1. Model analysis**
- Web-click sequences

**App2. Event discovery**
- Google Trend data
App2. Event discovery (GoogleTrend)

Anomaly detection (flu-related topics, 10 years)

![Graph showing anomaly detection over time for flu-related topics over 10 years. The graph includes lines for different flu-related keywords such as "flu fever", "flu symptom", "flu headache", and "flu medicine". The x-axis represents time (per week) and the y-axis represents value.]
AutoPlait detects 1 unusual spike in 2009 (i.e., swine flu)
App2. Event discovery (GoogleTrend)

Turning point detection (seasonal sweets topics)

![Graph showing trends in search queries for seasonal sweets topics over time. The x-axis represents time (per week), and the y-axis represents value. The graph plots the trends for "ice cream", "milk shake", "hot cocoa", and "gingerbread".]
App2. Event discovery (GoogleTrend)

Turning point detection (seasonal sweets topics)

Trend suddenly changed in 2010 (release of android OS “Ginger bread”, “Ice Cream Sandwich”)
App2. Event discovery (GoogleTrend)

Trend discovery (game-related topics)

![Trend discovery graph](image-url)
App2. Event discovery (GoogleTrend)

Trend discovery (game-related topics)

It discovers 3 phases of “game console war”
(Xbox & PlayStation / Wii / Mobile social games)
Outline

- Motivation
- Problem definition
- Compression & summarization
- Algorithms
- Experiments
- AutoPlait at work
- Conclusions
Conclusions

AutoPlait has the following properties

• **Effective** ✓
  Find optimal segments/regimes

• **Sense-making** ✓
  Reasonable regimes

• **Fully-automatic** ✓
  No magic numbers

• **Scalable** ✓
  It scales linearly
Thank you!

Mail: yasuko@cs.kumamoto-u.ac.jp