Part 1

Similarity search, pattern discovery and summarization

Yasushi Sakurai (Kumamoto University)
Yasuko Matsubara (Kumamoto University)
Christos Faloutsos (Carnegie Mellon University)

Kumamoto U CMU CS

Outline

- Motivation
- Similarity Search and Indexing
- Feature extraction
- Linear forecasting
- ➡ Streaming pattern discovery
- Automatic mining

Kumamoto U CMU CS

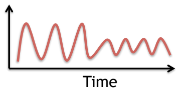
Stream mining

- Applications
  - Sensor monitoring
  - Network analysis
  - Financial and/or business transaction data
  - Web access and media service logs
  - Moving object tracking
  - Industrial manufacturing

Kumamoto U CMU CS

Stream mining

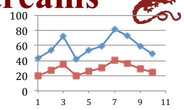
- Requirements



  - **Fast**
    high performance and quick response
  - **Nimble**
    low memory consumption, single scan
  - **Accurate**
    good approximation for pattern discovery and feature extraction

Kumamoto U CMU CS

Monitoring data streams



- Correlation coefficient

$$\rho = \frac{\sum_{t=1}^{n}(x_t - \bar{x})\cdot(y_t - \bar{y})}{\sigma(x)\cdot\sigma(y)} \qquad \sigma(x) = \sqrt{\sum_{t=1}^{n}(x_t - \bar{x})^2}$$

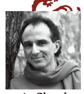
- Correlation coefficient and the (Euclidean) distance

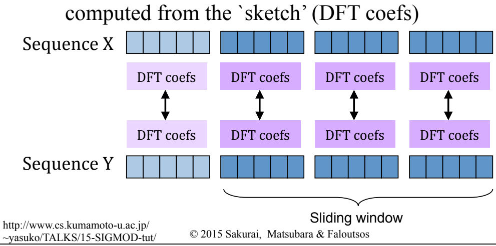
$$\rho = 1 - \frac{1}{2}\sum_{t=1}^{n}(\hat{x}_t - \hat{y}_t)^2 \qquad \hat{x}_t = (x_t - \bar{x})/\sigma(x)$$

Kumamoto U CMU CS

Monitoring data streams

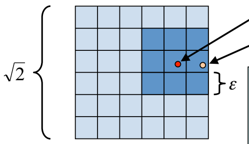
- Correlation monitoring [Zhu+, vldb02]
  - DFT coefficients for each basic window
  - Correlation coefficient of each sliding window computed from the `sketch' (DFT coefs)


Dennis Shasha



Monitoring data streams

- Grid structure (to avoid checking all pairs)
  - DFT coefficients yields a vector
  - High correlation -> closeness in the vector space

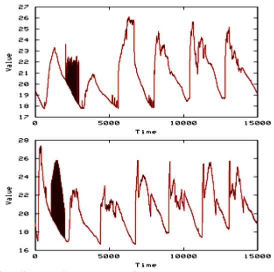


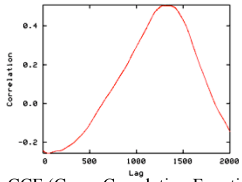
Correlation coefficients and the Euclidean distance

$$\rho = 1 - \frac{1}{2}\sum_{t=1}^{n}(\hat{x}_t - \hat{y}_t)^2$$

Monitoring data streams

- Lag correlation [Sakurai+, sigmod05]

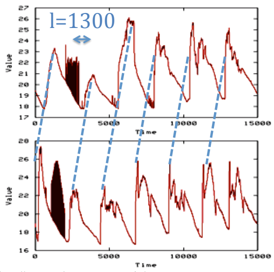




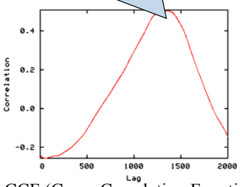
CCF (Cross-Correlation Function)

Monitoring data streams

- Lag correlation [Sakurai+, sigmod05]



correlated with lag l=1300



CCF (Cross-Correlation Function)

Lag correlation

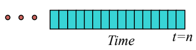
- Definition of 'score', absolute value of $R(l)$

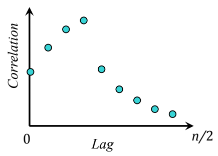
$$score(l) = |R(l)| \qquad R(l) = \frac{\sum_{t=l+1}^{n}(x_t - \bar{x})(y_{t-l} - \bar{y})}{\sqrt{\sum_{t=l+1}^{n}(x_t - \bar{x})^2}\sqrt{\sum_{t=1}^{n-l}(y_t - \bar{y})^2}}$$

- Lag correlation
  - Given a threshold γ, $score(l) > \gamma$
  - A local maximum
  - The earliest such maximum, if more maxima exist

Lag correlation

- Why not naïve?
  - Compute correlation coefficient for each lag l = {0, 1, 2, 3, …, n/2}
- But
  - $O(n)$ space
  - $O(n^2)$ time
  - or $O(n \log n)$ time w/ FFT

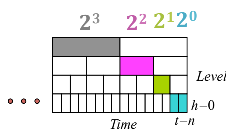


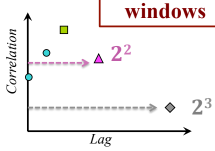


Lag correlation

- BRAID
  - Geometric lag probing + smoothing
  - Use colored windows
  - Keep track of only a geometric progression of the lag values: l = {0, 1, 2, 4, 8, …, 2^h, …}

Multi-scale windows

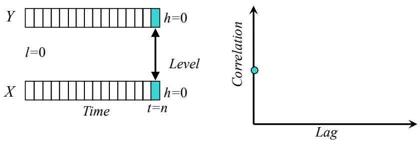




Kumamoto U CMU CS

Lag correlation

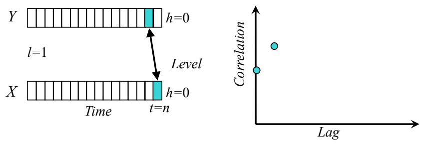
- BRAID
  - Geometric lag probing + smoothing
  - Keep track of only a geometric progression of the lag values: $l = \{0, 1, 2, 4, 8, \ldots, 2^h, \ldots\}$



Kumamoto U CMU CS

Lag correlation

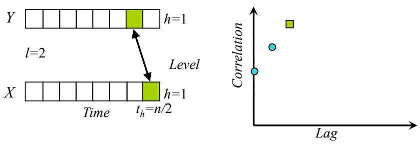
- BRAID
  - Geometric lag probing + smoothing
  - Keep track of only a geometric progression of the lag values: $l = \{0, 1, 2, 4, 8, \ldots, 2^h, \ldots\}$



Kumamoto U CMU CS

Lag correlation

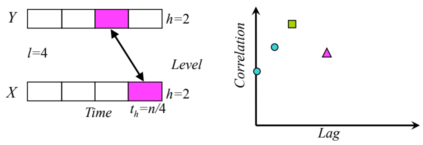
- BRAID
  - Geometric lag probing + smoothing
  - Keep track of only a geometric progression of the lag values: $l = \{0, 1, 2, 4, 8, \ldots, 2^h, \ldots\}$



Kumamoto U CMU CS

Lag correlation

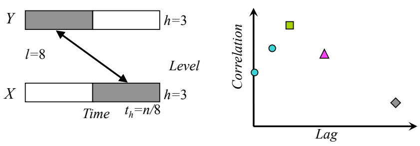
- BRAID
  - Geometric lag probing + smoothing
  - Keep track of only a geometric progression of the lag values: $l = \{0, 1, 2, 4, 8, \ldots, 2^h, \ldots\}$



Kumamoto U CMU CS

Lag correlation

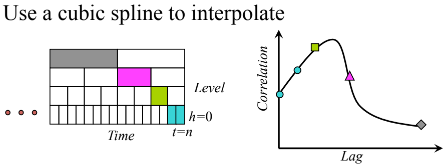
- BRAID
  - Geometric lag probing + smoothing
  - Keep track of only a geometric progression of the lag values: $l = \{0, 1, 2, 4, 8, \ldots, 2^h, \ldots\}$



Kumamoto U CMU CS

Lag correlation

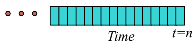
- BRAID
  - Geometric lag probing + smoothing
  - Keep track of only a geometric progression of the lag values: $l = \{0, 1, 2, 4, 8, \ldots, 2^h, \ldots\}$
  - Use a cubic spline to interpolate



Kumamoto U — CMU CS

Lag correlation

- Why not naïve?
  - Compute correlation coefficient for each lag $l = \{0, 1, 2, 3, \ldots, n/2\}$
- But
  - $O(n)$ space
  - $O(n^2)$ time
  - or $O(n \log n)$ time w/



BRAID
- $O(\log n)$ space
- $O(1)$ time

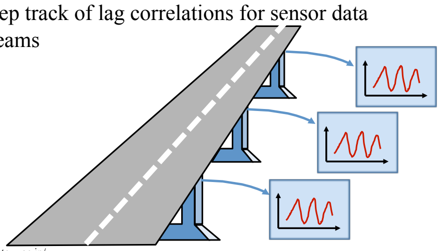
Multi-scale windows

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 19

Kumamoto U — CMU CS

BRAID in the real world

- Bridge structural health monitoring
  - Structural monitoring using vibration/shock sensors
  - Keep track of lag correlations for sensor data streams



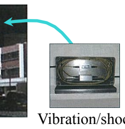
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 20

Kumamoto U — CMU CS

BRAID in the real world

- Bridge structural health monitoring
  - Goal: real-time anomaly detection for disaster prevention
  - Several thousands readings (per sec) from several hundreds sensor nodes

Structural health monitoring



Vibration/shock sensor

- Uses BRAID
- Metropolitan Expressway (Tokyo, Japan)

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 21

Kumamoto U — CMU CS

BRAID in the real world

- Bridge structural health monitoring with BRAID

Metropolitan Expressway (Tokyo, Japan)

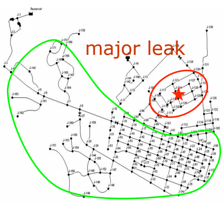
Tokyo Gate Bridge (Tokyo, Japan)

Can Tho Bridge (Vietnam)

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 22

Kumamoto U — CMU CS

Feature extraction from streams



water distribution network

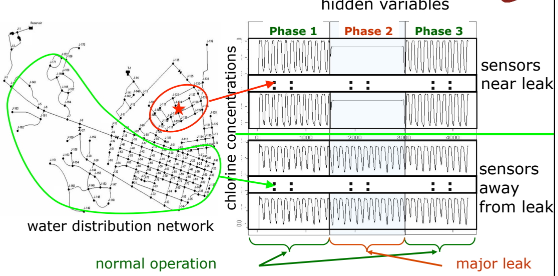
- Find hidden variables from streams [Papadimitriou+, vldb2005]

May have hundreds of measurements, but it is **unlikely they are completely unrelated**!

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 23

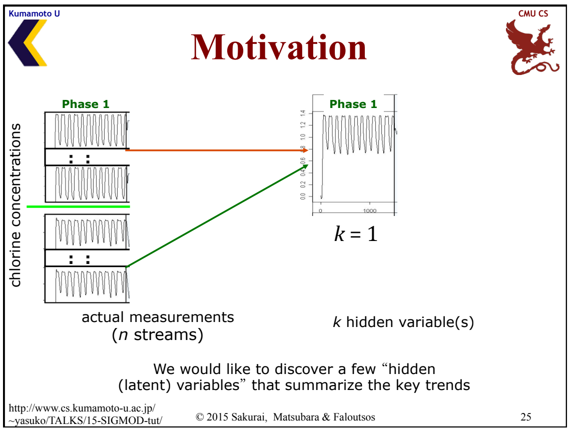
Kumamoto U — CMU CS

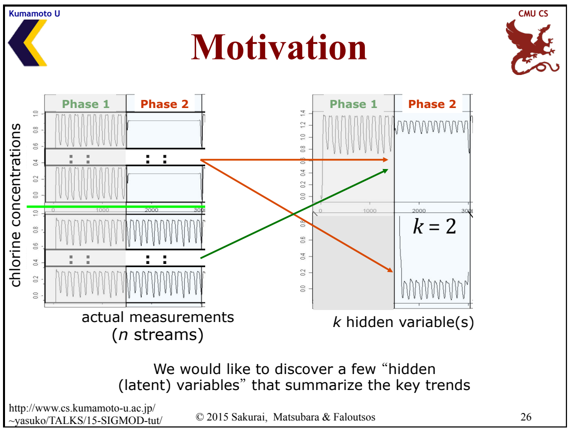
Feature extraction from streams

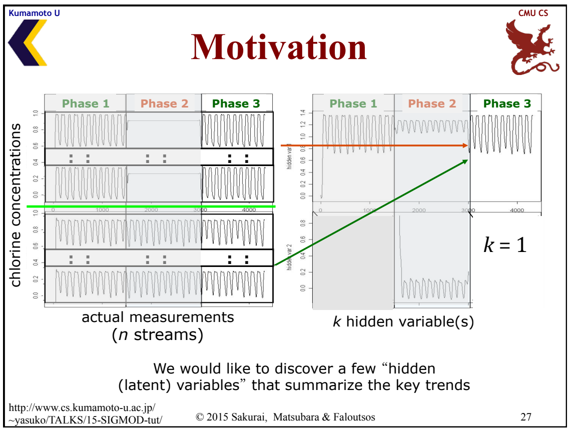


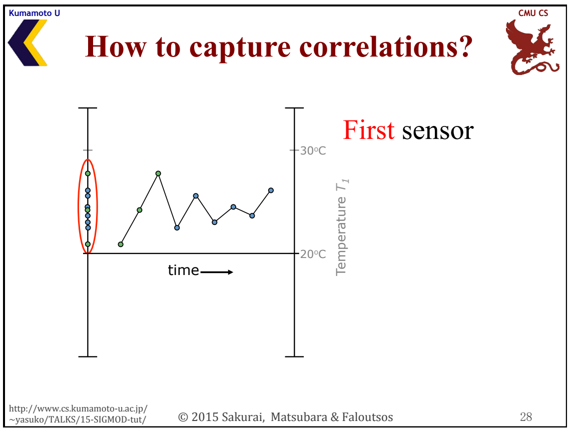
May have hundreds of measurements, but it is **unlikely they are completely unrelated**!

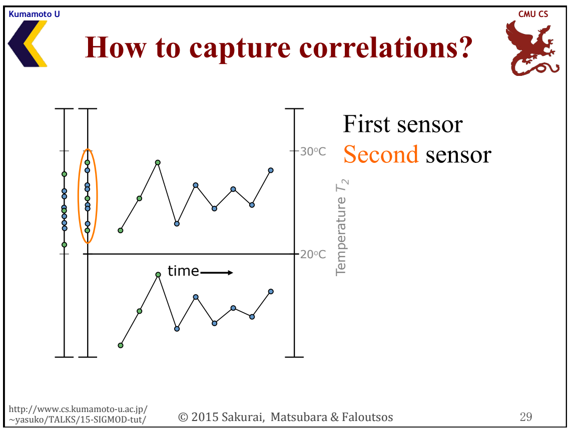
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 24

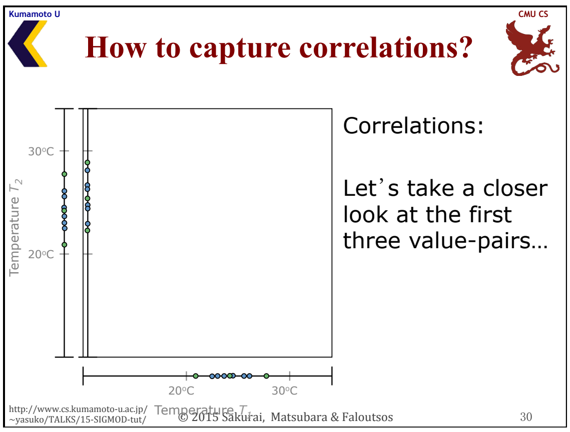

Kumamoto U
CMU CS
Motivation
Phase 1
Phase 1
chlorine concentrations
k = 1
actual measurements
(n streams)
k hidden variable(s)
We would like to discover a few "hidden
(latent) variables" that summarize the key trends
http://www.cs.kumamoto-u.ac.jp/
~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
25


Kumamoto U
CMU CS
Motivation
Phase 1
Phase 2
Phase 1
Phase 2
chlorine concentrations
k = 2
actual measurements
(n streams)
k hidden variable(s)
We would like to discover a few "hidden
(latent) variables" that summarize the key trends
http://www.cs.kumamoto-u.ac.jp/
~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
26

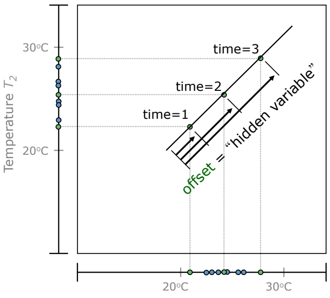

Kumamoto U
CMU CS
Motivation
Phase 1
Phase 2
Phase 3
Phase 1
Phase 2
Phase 3
chlorine concentrations
k = 1
actual measurements
(n streams)
k hidden variable(s)
We would like to discover a few "hidden
(latent) variables" that summarize the key trends
http://www.cs.kumamoto-u.ac.jp/
~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
27


Kumamoto U
CMU CS
How to capture correlations?
First sensor
30°C
20°C
Temperature T_1
time
http://www.cs.kumamoto-u.ac.jp/
~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
28


Kumamoto U
CMU CS
How to capture correlations?
First sensor
Second sensor
30°C
20°C
Temperature T_2
time
http://www.cs.kumamoto-u.ac.jp/
~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
29


Kumamoto U
CMU CS
How to capture correlations?
Correlations:
Let's take a closer look at the first three value-pairs…
30°C
20°C
Temperature T_2
20°C
30°C
Temperature T_1
http://www.cs.kumamoto-u.ac.jp/
~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
30

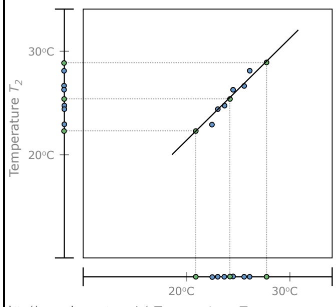
How to capture correlations?



First three lie (almost) on a line in the space of value-pairs…

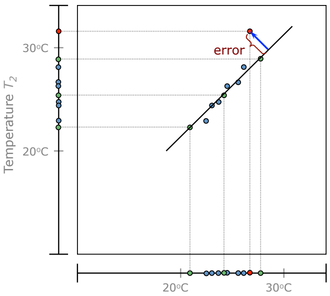
- O(n) numbers for the slope, and
- *One* number for each value-pair (offset on line)

How to capture correlations?



Other pairs also follow the same pattern: they lie (approximately) on this line

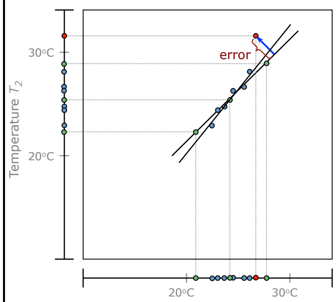
Incremental update



For each new point

- Project onto current line
- Estimate error

Incremental update

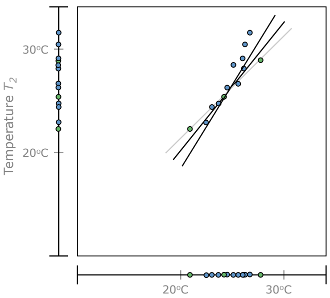


For each new point

- Project onto current line
- Estimate error
- Rotate line in the direction of the error and in proportion to its magnitude

➔O(n) time

Incremental update



For each new point

- Project onto current line
- Estimate error
- Rotate line in the direction of the error and in proportion to its magnitude

Related work

- Wavelet over streams [Gilbert+,vldb01] [Guha +,vldb04]
- Fourier representations [Gilbert+, stoc02]
- KNN [Koudas+, 04] [Korn+, vldb02]
- Histograms [Guha+, stoc01]
- Clustering [Guha+, focs00] [Aggarwal+, vldb03]
- Sketches [Indyk+, vldb00] [Cormode+, J. Algorithms 05]
- …
- …

Kumamoto U | CMU CS

Related work

Tutorial@PODS'15

Graham Cormode

- Heavy hitters [Cormode+, vldb03]
- Data embedding [Indyk+, focs00]
- Burst detection [Zhu+, kdd03]
- Segmentation [Keogh+, icdm01]
- Multiple scale analysis [Papadimitriou+, sigmod06]
- Fractal [Korn+, sigmod06]
- Time warping [Sakurai+, icde07]…
- …

Tutorial@SIGMOD'15

Divesh Srivastava

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 37

Kumamoto U | CMU CS

Outline

- Motivation
- Similarity Search and Indexing
- Feature extraction
- Streaming pattern discovery
- Linear forecasting
- ➡ Automatic mining

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 38

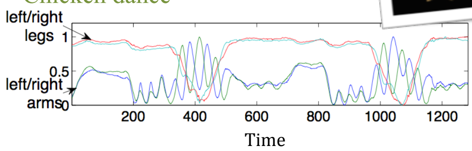
Kumamoto U | CMU CS

Motivation

Given: co-evolving time-series

– e.g., MoCap (leg/arm sensors)

"Chicken dance"



http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 39

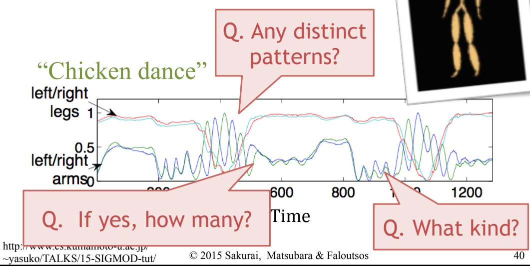
Kumamoto U | CMU CS

Motivation

Given: co-evolving time-series

– e.g., MoCap (leg/arm sensors)

"Chicken dance"



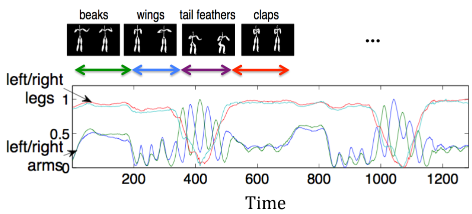
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 40

Kumamoto U | CMU CS

Motivation

Challenges: co-evolving sequences

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



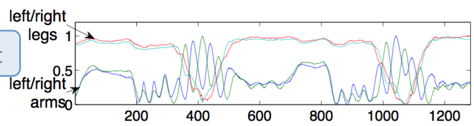
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 41

Kumamoto U | CMU CS

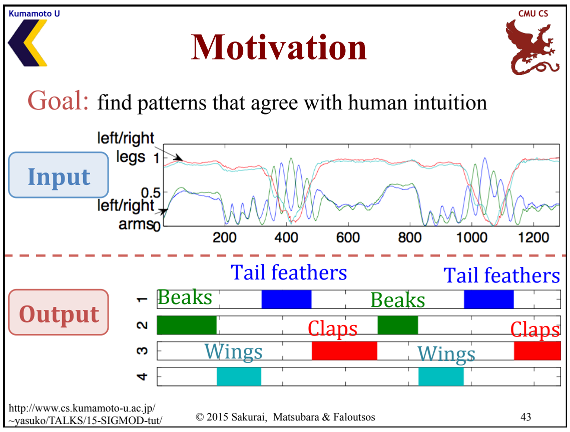
Motivation

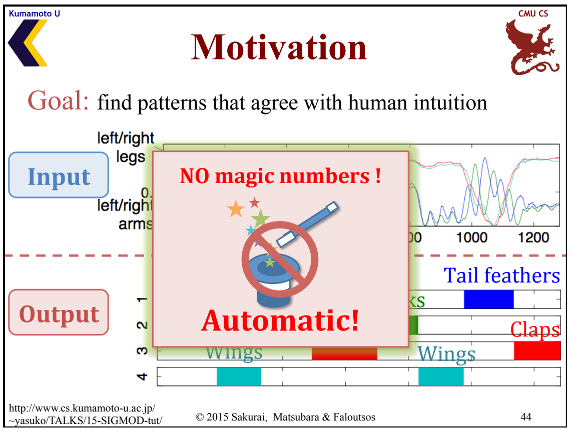
Goal: find patterns that agree with human intuition

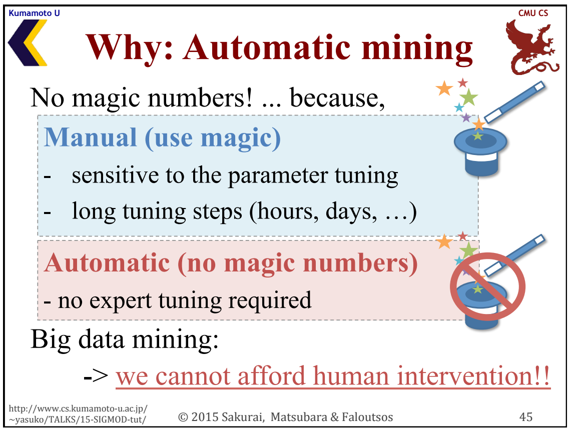
Input

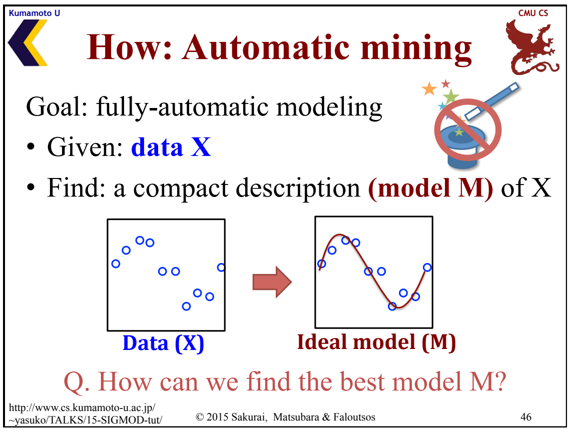


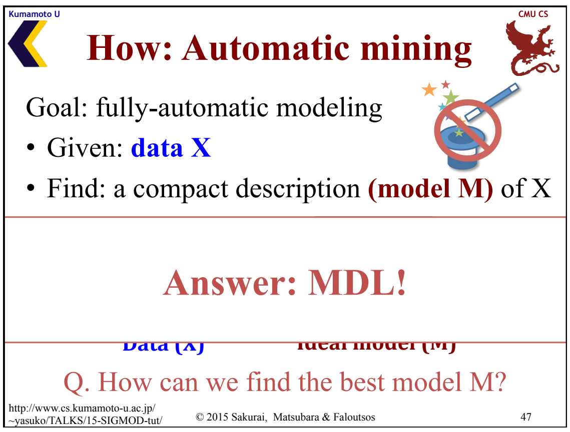
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 42

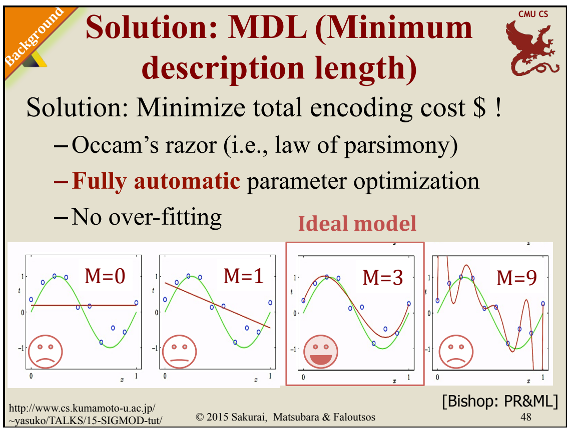

Kumamoto U
CMU CS
Motivation
Goal: find patterns that agree with human intuition
Input
left/right legs
left/right arms
1
0.5
0
200 400 600 800 1000 1200
Output
Tail feathers
Tail feathers
Beaks
Beaks
Claps
Claps
Wings
Wings
1 2 3 4
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
43

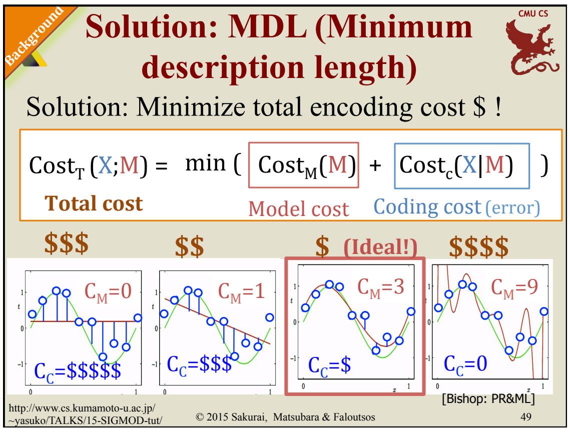

Kumamoto U
CMU CS
Motivation
Goal: find patterns that agree with human intuition
Input
Output
NO magic numbers !
Automatic!
Tail feathers
Claps
Wings
1000 1200
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
44

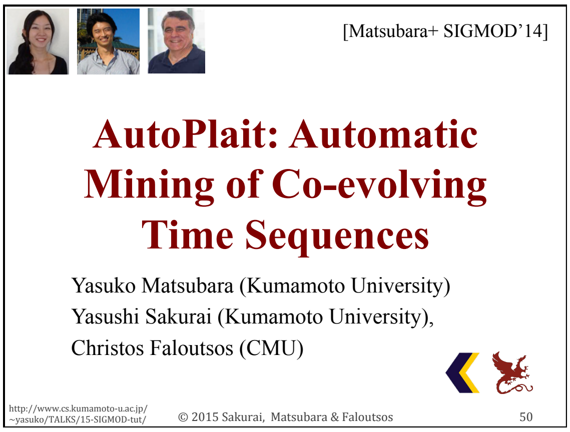

Kumamoto U
CMU CS
Why: Automatic mining
No magic numbers! ... because,
Manual (use magic)
- sensitive to the parameter tuning
- long tuning steps (hours, days, …)
Automatic (no magic numbers)
- no expert tuning required
Big data mining:
-> we cannot afford human intervention!!
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
45

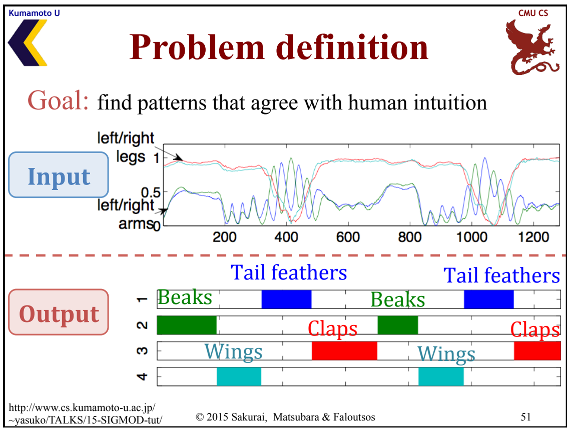

Kumamoto U
CMU CS
How: Automatic mining
Goal: fully-automatic modeling
• Given: data X
• Find: a compact description (model M) of X
Data (X)
Ideal model (M)
Q. How can we find the best model M?
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
46

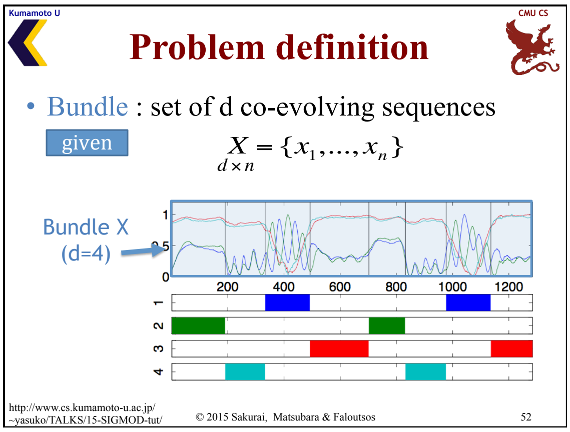

Kumamoto U
CMU CS
How: Automatic mining
Goal: fully-automatic modeling
• Given: data X
• Find: a compact description (model M) of X
Answer: MDL!
Q. How can we find the best model M?
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
47

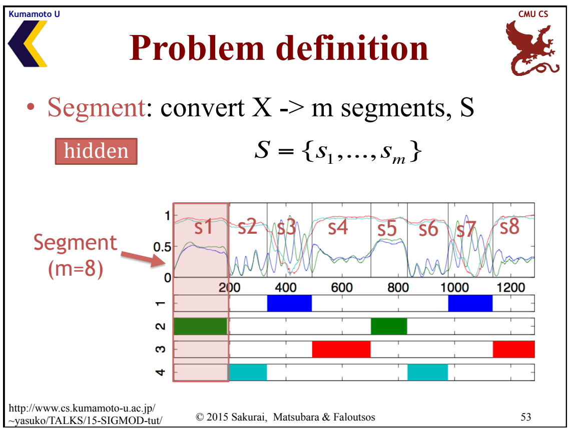

Background
CMU CS
Solution: MDL (Minimum description length)
Solution: Minimize total encoding cost $!
– Occam's razor (i.e., law of parsimony)
– Fully automatic parameter optimization
– No over-fitting
Ideal model
M=0
M=1
M=3
M=9
[Bishop: PR&ML]
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
48

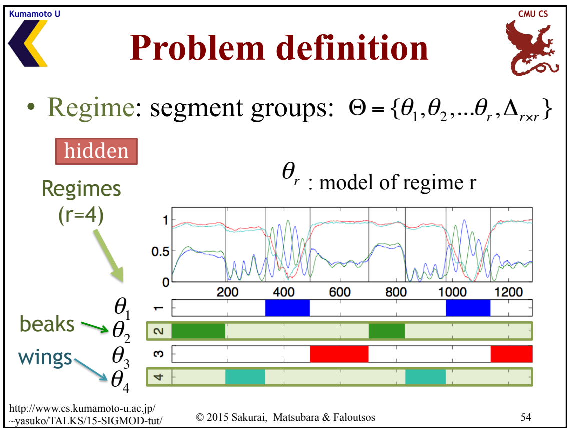

Background
Solution: MDL (Minimum description length)
CMU CS
Solution: Minimize total encoding cost $!
Cost_T (X;M) = min (Cost_M(M) + Cost_c(X|M))
Total cost
Model cost
Coding cost (error)
$$$
$$
$ (Ideal!)
$$$$
C_M=0
C_C=$$$$$
C_M=1
C_C=$$$
C_M=3
C_C=$
C_M=9
C_C=0
[Bishop: PR&ML]
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
49


[Matsubara+ SIGMOD'14]
AutoPlait: Automatic Mining of Co-evolving Time Sequences
Yasuko Matsubara (Kumamoto University)
Yasushi Sakurai (Kumamoto University),
Christos Faloutsos (CMU)
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
50


Kumamoto U
CMU CS
Problem definition
Goal: find patterns that agree with human intuition
Input
left/right legs
left/right arms
Output
Tail feathers
Tail feathers
Beaks
Beaks
Claps
Claps
Wings
Wings
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
51


Kumamoto U
CMU CS
Problem definition
• Bundle : set of d co-evolving sequences
given
$X_{d \times n} = \{x_1, \ldots, x_n\}$
Bundle X (d=4)
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
52


Kumamoto U
CMU CS
Problem definition
• Segment: convert X -> m segments, S
hidden
$S = \{s_1, \ldots, s_m\}$
Segment (m=8)
s1 s2 s3 s4 s5 s6 s7 s8
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
53

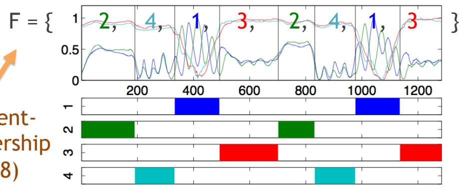

Kumamoto U
CMU CS
Problem definition
• Regime: segment groups: $\Theta = \{\theta_1, \theta_2, \ldots \theta_r, \Delta_{r \times r}\}$
hidden
θ_r : model of regime r
Regimes (r=4)
beaks
wings
θ_1 θ_2 θ_3 θ_4
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
54

Kumamoto U — CMU CS

Problem definition

- Segment-membership: assignment

hidden $F = \{f_1, \ldots, f_m\}$

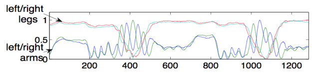


Kumamoto U — CMU CS

Problem definition

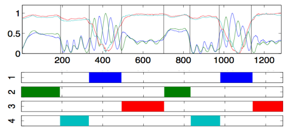
- Given: bundle X

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



- Find: compact description C of X

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

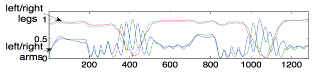


Kumamoto U — CMU CS

Problem definition

- Given: bundle X

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



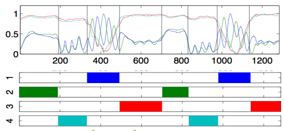
- Find: compact description C of X

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

m segments

r regimes

Segment-membership



Kumamoto U — CMU CS

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 ?

Kumamoto U — CMU CS

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

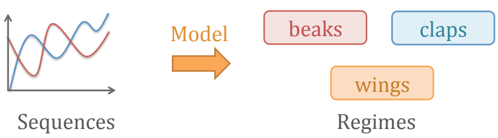
Q2. How to decide # of regimes/segments ?

Idea (2): Model description cost

Kumamoto U — CMU CS

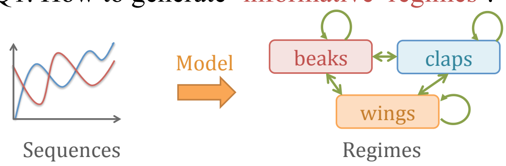
Idea (1): MLCM: multi-level chain model

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

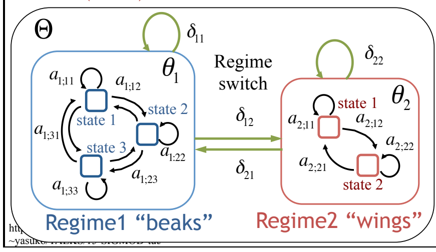
Details

$\Theta = \{\theta_1, \theta_2, \ldots \theta_r, \Delta_{r \times r}\}$ $\quad (\theta_i = \{\pi, A, B\})$

r regimes (HMMs)

across-regime transition prob.

Single HMM parameters



Regimes r=2

Regime 1 (k=3)

Regime 2 (k=2)

Idea (2): model description cost

Q2. How to decide # of regimes/segments ?

Idea (2): Model description cost

- Minimize encoding cost
- find "optimal" # of segments/regimes

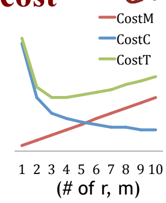
Idea (2): model description cost

Idea: Minimize encoding cost!

min ($Cost_M(M)$ + $Cost_c(X \mid M)$)

Model cost

Coding cost



Good compression

Good description

Idea (2): model description cost

Details

Total cost of bundle X, given C

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

$$Cost_T(\boldsymbol{X};\mathcal{C}) = Cost_T(\boldsymbol{X}; m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F})$$
$$= \log^*(n) + \log^*(d) + \log^*(m) + \log^*(r) + m\log(r)$$
$$+ \sum_{i=1}^{m-1} \log^* |s_i| + Cost_M(\boldsymbol{\Theta}) + Cost_C(\boldsymbol{X}|\boldsymbol{\Theta}) \qquad (6)$$

Idea (2): model description cost

Details

Total cost of bundle X, given C

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

duration/ dimensions

# of segments/ regimes

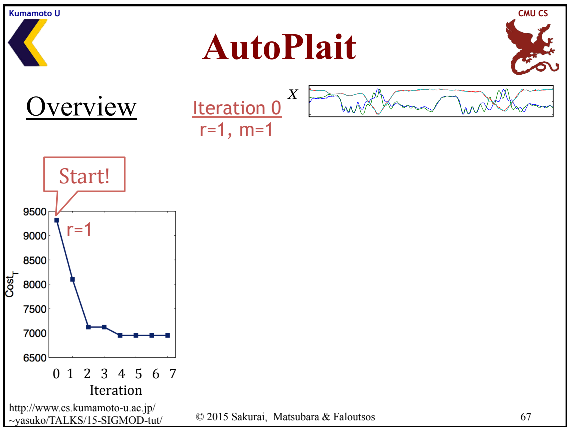
segment-membership F

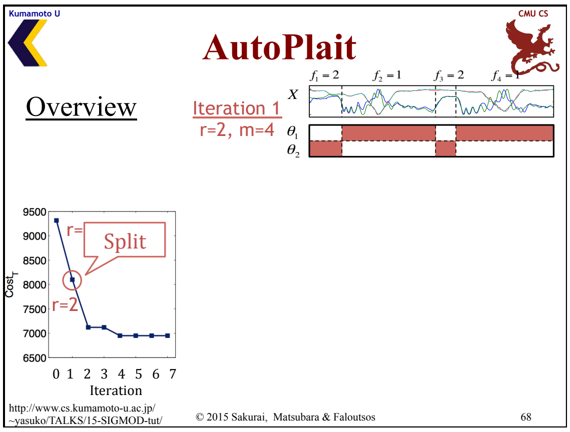
$$Cost_T(\boldsymbol{X};\mathcal{C}) = Cost_T(\boldsymbol{X}; m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F})$$
$$= \boxed{\log^*(n) + \log^*(d)} + \boxed{\log^*(m) + \log^*(r)} + \boxed{m\log(r)}$$
$$+ \sum_{i=1}^{m-1} \boxed{\log^* |s_i|} + \boxed{Cost_M(\boldsymbol{\Theta})} + \boxed{Cost_C(\boldsymbol{X}|\boldsymbol{\Theta})} \qquad (6)$$

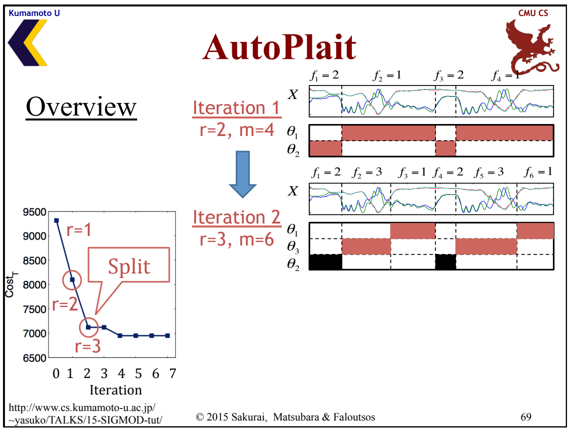
segment lengths

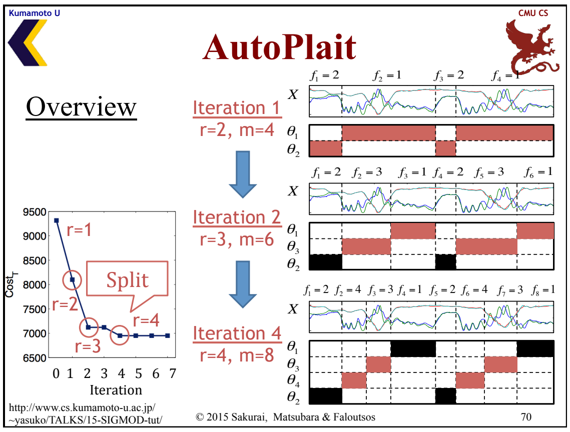
Model description cost of Θ

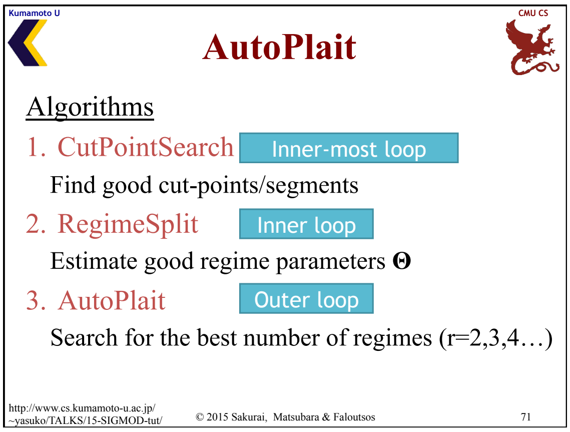
Coding cost of X given Θ

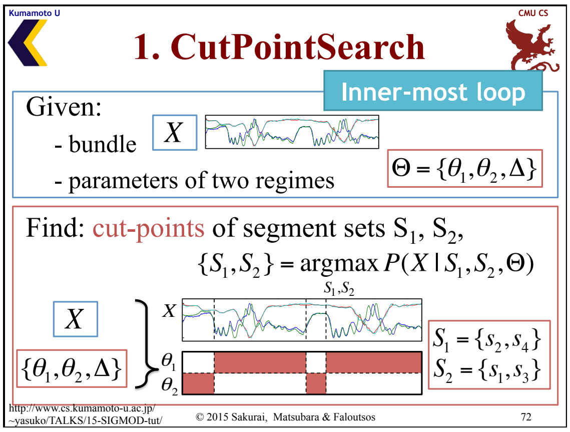

Kumamoto U
CMU CS
AutoPlait
Overview
Iteration 0
r=1, m=1
X
Start!
r=1
9500
9000
8500
8000
7500
7000
6500
Cost_T
0 1 2 3 4 5 6 7
Iteration
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
67


Kumamoto U
CMU CS
AutoPlait
Overview
Iteration 1
r=2, m=4
X
θ1
θ2
Split
r=2
Cost_T
Iteration
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
68


Kumamoto U
CMU CS
AutoPlait
Overview
Iteration 1
r=2, m=4
Iteration 2
r=3, m=6
X
θ1
θ2
θ3
Split
r=1
r=2
r=3
Cost_T
Iteration
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
69


Kumamoto U
CMU CS
AutoPlait
Overview
Iteration 1
r=2, m=4
Iteration 2
r=3, m=6
Iteration 4
r=4, m=8
X
θ1
θ2
θ3
θ4
Split
r=1
r=2
r=3
r=4
Cost_T
Iteration
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
70

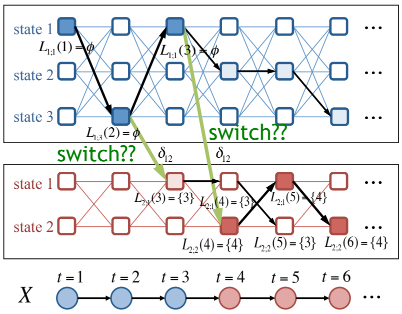

Kumamoto U
CMU CS
AutoPlait
Algorithms
1. CutPointSearch
Inner-most loop
Find good cut-points/segments
2. RegimeSplit
Inner loop
Estimate good regime parameters Θ
3. AutoPlait
Outer loop
Search for the best number of regimes (r=2,3,4…)
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
71


Kumamoto U
CMU CS
1. CutPointSearch
Inner-most loop
Given:
- bundle X
- parameters of two regimes
Θ = {θ1, θ2, Δ}
Find: cut-points of segment sets S1, S2,
{S1, S2} = argmax_{S1,S2} P(X | S1, S2, Θ)
X
{θ1, θ2, Δ}
S1 = {s2, s4}
S2 = {s1, s3}
http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/
© 2015 Sakurai, Matsubara & Faloutsos
72

1. CutPointSearch

DP algorithm to compute likelihood:

$P(X \mid \Theta)$



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

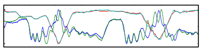
It guarantees the optimal cut points

- (Details in paper)

2. RegimeSplit

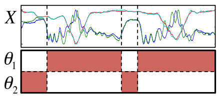
Inner loop

Given:
- bundle X



Find: two regimes

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

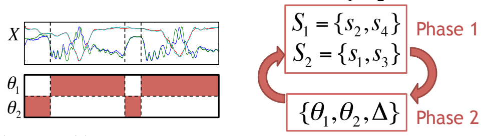


$\Theta = \{\theta_1, \theta_2, \Delta\}$

2. RegimeSplit

Two-phase iterative approach

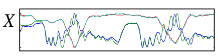
- Phase 1: (CutPointSearch)
  - Split segments into two groups : S_1, S_2
- Phase 2: (BaumWelch)
  - Update model parameters: $\Theta = \{\theta_1, \theta_2, \Delta\}$



3. AutoPlait

Outer loop

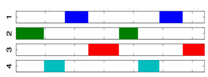
Given:
- bundle X



Find: r regimes (r=2, 3, 4, …)

- i.e., find full parameter set

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

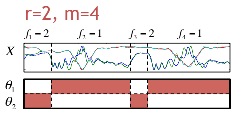


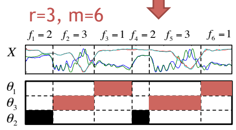
3. AutoPlait

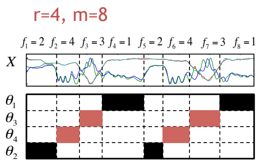
Split regimes r=2,3,…, as long as cost keeps decreasing

- Find appropriate # of regimes

$$r = \min_r Cost_T(X; m, r, S, \Theta, F)$$







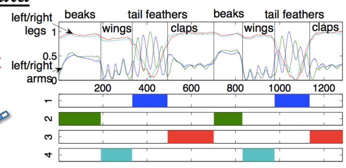
Results

- Mocap data
- WebClick data
- Google Trends

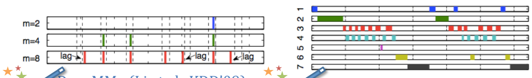
Q1. Sense-making

MoCap data

AutoPlait (NO magic numbers)



(a) AUTOPLAIT (no user defined parameters)

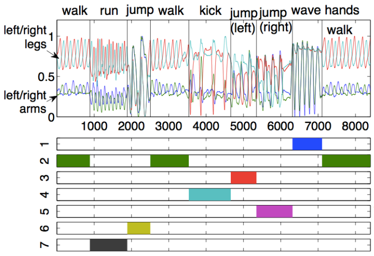


DynaMMo (Li et al., KDD'09)

pHMM (Wang et al., SIGMOD'11)

Q1. Sense-making

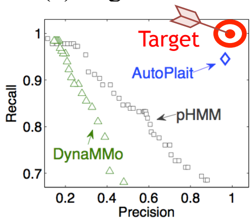
MoCap data



AutoPlait (NO magic numbers)

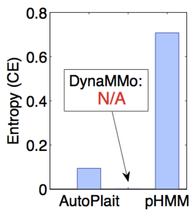
Q2. Accuracy

(a) Segmentation



(a) Precision and recall (higher is better)

(b) Clustering



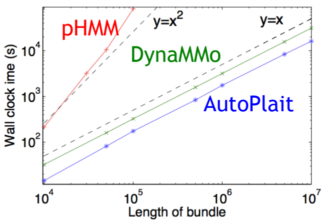
(b) CE score (lower is better)

AutoPlait needs "no magic numbers"

Q3. Scalability

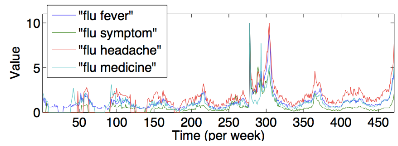
Wall clock time vs. data size (length) : n

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



App. Event discovery (GoogleTrend)

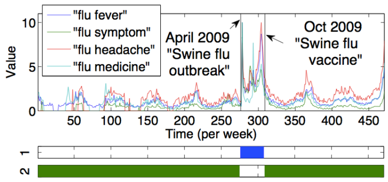
Anomaly detection (flu-related topics,10 years)



Kumamoto U | CMU CS

App. Event discovery (GoogleTrend)

Anomaly detection (flu-related topics,10 years)



(a) Flu-related topics (regimes $r = 2$)

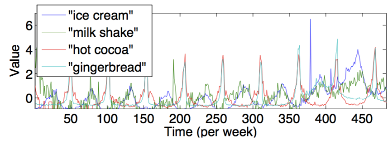
AutoPlait detects 1 unusual spike in 2009 (i.e., swine flu)

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 85

Kumamoto U | CMU CS

App. Event discovery (GoogleTrend)

Turning point detection (seasonal sweets topics)

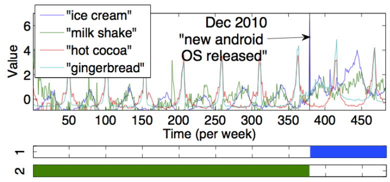


http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 86

Kumamoto U | CMU CS

App. Event discovery (GoogleTrend)

Turning point detection (seasonal sweets topics)



(b) Seasonal sweets topics (regimes $r = 2$)

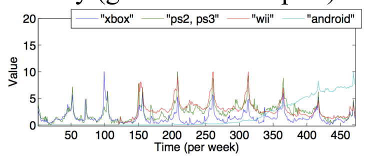
Trend suddenly changed in 2010 (release of android OS "Ginger bread", "Ice Cream Sandwich")

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 87

Kumamoto U | CMU CS

App. Event discovery (GoogleTrend)

Trend discovery (game-related topics)

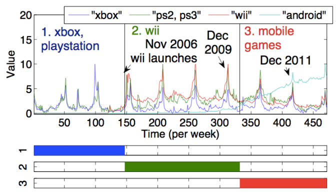


http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 88

Kumamoto U | CMU CS

App. Event discovery (GoogleTrend)

Trend discovery (game-related topics)



(c) Game-related topics (regimes $r = 3$)

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

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 89

Kumamoto U | CMU CS

Industrial contribution

- Automobile sensor data
  - location, velocity, longitudinal/lateral acceleration

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 90

Kumamoto U CMU CS

Code at

- http://www.cs.kumamoto-u.ac.jp/~yasuko/software.html

Kumamoto U CMU CS

Part 1 – Conclusions

- Motivation
- Similarity Search and Indexing
- Feature extraction
- Linear forecasting
- Streaming pattern discovery
- Automatic mining

Kumamoto U CMU CS

Part 1 – Conclusions

- Motivation
- Similarity Search and Indexing
  - Euclidean/time-warping
  - extract features
  - index (SAM, R-tree)
- Feature extraction
  - SVD, ICA, DFT, DWT (multi-scale windows)

Kumamoto U CMU CS

Part 1 – Conclusions

- Linear forecasting
  - AR, RLS
- Streaming pattern discovery
  - RLS, "incremental" wavelet transform
  - Multi-scale windows
- Automatic mining
  - MDL

Kumamoto U CMU CS

References

- Yunyue Zhu, Dennis Shasha ``*StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time*' ' VLDB, August, 2002. pp. 358-369.
- Spiros Papadimitriou, Jimeng Sun, Christos Faloutsos. *Streaming Pattern Discovery in Multiple Time-Series*. VLDB 2005.
- Yasushi Sakurai, Spiros Papadimitriou, Christos Faloutsos. *BRAID: Stream Mining through Group Lag Correlations*. SIGMOD 2005.
- Anna C. Gilbert, Yannis Kotidis, S. Muthukrishnan, Martin Strauss. *Surfing Wavelets on Streams: One-Pass Summaries for Approximate Aggregate Queries*. VLDB 2001.
- Sudipto Guha, Nick Koudas, Amit Marathe, Divesh Srivastava. *Merging the Results of Approximate Match Operations*. VLDB 2004.
- Anna C. Gilbert, Sudipto Guha, Piotr Indyk, S. Muthukrishnan, Martin Strauss. *Near-optimal sparse fourier representations via sampling*. STOC 2002.

Kumamoto U CMU CS

References

- Nick Koudas, Beng Chin Ooi, Kian-Lee Tan, Rui Zhang. *Approximate NN queries on Streams with Guaranteed Error/performance Bounds*. VLDB 2004.
- Flip Korn, S. Muthukrishnan, Divesh Srivastava. *Reverse Nearest Neighbor Aggregates Over Data Streams*. VLDB 2002.
- Sudipto Guha, Nick Koudas, Kyuseok Shim. *Data-streams and histograms*. STOC 2001.
- Sudipto Guha, Nina Mishra, Rajeev Motwani, Liadan O'Callaghan. *Clustering Data Streams*. FOCS 2000.
- Charu C. Aggarwal, Jiawei Han, Jianyong Wang, Philip S. Yu. *A Framework for Clustering Evolving Data Streams*. VLDB 2003.

Kumamoto U CMU CS

References

- Piotr Indyk, Nick Koudas, S. Muthukrishnan. *Identifying Representative Trends in Massive Time Series Data Sets Using Sketches*. VLDB 2000.
- Graham Cormode, S. Muthukrishnan. *An improved data stream summary: the count-min sketch and its applications*. J. Algorithms 55 (1), 2005.
- Graham Cormode, Flip Korn, S. Muthukrishnan, Divesh Srivastava. *Finding Hierarchical Heavy Hitters in Data Streams*. VLDB 2003.
- Piotr Indyk. *Stable Distributions, Pseudorandom Generators, Embeddings and Data Stream Computation*. FOCS 2000.
- Yunyue Zhu, Dennis Shasha. *Efficient elastic burst detection in data streams*. KDD 2003.

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 97

Kumamoto U CMU CS

References

- Eamonn J. Keogh, Selina Chu, David M. Hart, Michael J. Pazzani. *An Online Algorithm for Segmenting Time Series*. ICDM 2001.
- Spiros Papadimitriou, Philip S. Yu. *Optimal multi-scale patterns in time series streams*. SIGMOD 2006.
- Flip Korn, S. Muthukrishnan, Yihua Wu. *Modeling skew in data streams*. SIGMOD 2006.
- Yasushi Sakurai, Christos Faloutsos, Masashi Yamamuro. *Stream Monitoring under the Time Warping Distance*. ICDE 2007.

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 98

Part 1

Similarity search, pattern discovery and summarization

Yasushi Sakurai (Kumamoto University)
Yasuko Matsubara (Kumamoto University)
Christos Faloutsos (Carnegie Mellon University)

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/ © 2015 Sakurai, Matsubara & Faloutsos 99