Part 1

Similarity search, pattern discovery and summarization

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

Outline

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

Stream mining

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

Monitoring data streams

• Correlation coefficient
  \[ \rho = \frac{\sum_{i=1}^{n} (x_i - \bar{x}) (y_i - \bar{y})}{\sigma(x) \cdot \sigma(y)} \]
  \[ \bar{x} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2 \]
  \[ \hat{\rho} = \frac{1}{n} \sum_{i=1}^{n} (\hat{x}_i - \bar{x}) / \sigma(x) \]

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

Correlation monitoring [Zhu+, vldb02]
  – DFT coefficients for each basic window
  – Correlation coefficient of each sliding window
    computed from the ‘sketch’ (DFT coefs)
Monitoring data streams

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

\[
\rho = 1 - \frac{1}{2} \sum_{t=1}^{n} (x_t - \hat{x}_t)(y_t - \hat{y}_t)
\]

Lag correlation [Sakurai+, sigmod05]

- Definition of ‘score’, absolute value of \( R(l) \)
  \[
  \text{score}(l) = |R(l)| = \frac{\sum_{t} (x_t - \bar{x})(y_{t-l} - \bar{y})}{\sqrt{\sum_{t} (x_t - \bar{x})^2} \sqrt{\sum_{t} (y_{t-l} - \bar{y})^2}}
  \]

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

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/ FFT

Lag correlation

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

BRAID in the real world

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

Feature extraction from streams

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

May have hundreds of measurements, but it is unlikely they are completely unrelated!
We would like to discover a few “hidden (latent) variables” that summarize the key trends.

**Phase 1**

- $k = 1$

**Phase 2**

- $k = 2$

**Phase 3**

We would like to discover a few “hidden (latent) variables” that summarize the key trends.
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)

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

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] [Kom+, vldb02]
- Histograms [Guha+, stoc01]
- Clustering [Guha+, focs00] [Aggarwal+, vldb03]
- Sketches [Indyk+, vldb00] [Cormode+, J. Algorithms 05]
- …
Related work

- 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]...

Outline

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

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

Input

Goal: find patterns that agree with human intuition
**Tutorial@SIGMOD’15**

---

**Motivation**

Goal: find patterns that agree with human intuition

- Input
- Output

---

**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!!

---

**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?

---

**Solution: MDL (Minimum description length)**

Solution: Minimize total encoding cost $S$!

- Occam’s razor (i.e., law of parsimony)
- Fully automatic parameter optimization
- No over-fitting

Ideal model

---
Solution: MDL (Minimum description length)
Solution: Minimize total encoding cost $!$

Validation

Kumamoto U
CMU CS

Solution: MDL (Minimum description length)
Solution: Minimize total encoding cost $!$

AutoPlait: Automatic Mining of Co-evolving Time Sequences
Yasuko Matsubara (Kumamoto University)
Yasushi Sakurai (Kumamoto University),
Christos Faloutsos (CMU)

Problem definition
Goal: find patterns that agree with human intuition

Input
Output

Problem definition
• Bundle: set of $d$ co-evolving sequences

Problem definition
• Segment: convert $X \rightarrow m$ segments, $S$

Problem definition
• Regime: segment groups: $\Theta = \{ \theta_1, \theta_2, \ldots, \theta_r, \Lambda_{\text{reg}} \}$

© 2015 Sakurai, Matsubara & Faloutsos
Problem definition

- Segment-membership: assignment
  \[ F = \{ f_1, \ldots, f_n \} \]

\[ F = \{ 2, 4, 1, 2, 4, 1, 3 \} \]

Main ideas

Goal: compact description of \( X \)
\[ C = \{ m, r, S, \Theta, F \} \]

Challenges:
- Q1. How to generate ‘informative’ regimes?
- Q2. How to decide # of regimes/segments?

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

Idea (2): Model description cost

© 2015 Sakurai, Matsubara & Faloutsos
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 (2): Model description cost

Q2. How to decide # of regimes/segments?

Idea (2): Model description cost
• Minimize encoding cost
• find “optimal” # of segments/regimes

Detailed equations and calculations for model description cost
AutoPlait

Overview

Iteration 0

r=1, m=1

Start!

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

AutoPlait

Overview

Iteration 1

r=2, m=4

Split

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

AutoPlait

Overview

Iteration 1

r=2, m=4

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

AutoPlait

Overview

Iteration 2

r=3, m=6

Split

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

AutoPlait

Overview

Iteration 2

r=3, m=6

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

AutoPlait

Overview

Iteration 4

r=4, m=8

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

Algorithms

1. CutPointSearch
   - Given: bundle \( \mathbf{X} \)
   - Parameters of two regimes \( \Theta = \{\theta_1, \theta_2, \Delta\} \)
   - Find: cut-points of segment sets \( S_1, S_2 \)
   \[ \{S_1, S_2\} = \text{argmax}_S \mathcal{P}(X | S_1, S_2, \Theta) \]
   \[ S_i = \{s_{i1}, s_{i2}\} \]

2. RegimeSplit
   - Estimate good regime parameters \( \Theta \)
   - Inner loop

3. AutoPlait
   - Search for the best number of regimes \( r=2, 3, 4 \ldots \)
   - Outer loop

http://www.cs.kumamoto-u.ac.jp/~yasuko/TALKS/15-SIGMOD-tut/

© 2015 Sakurai, Matsubara & Faloutsos
1. CutPointSearch

- Find: \( r \) regimes (\( r = 2, 3, 4, \ldots \) )

- Given: \( P(X|\Theta) \)

- DP algorithm to compute likelihood:

\[
P(X|\Theta) = \sum_{\Theta'} P(X|\Theta',\Theta) P(\Theta')
\]

- Theoretical analysis

- Scalability
  - It takes \( O(nd^2) \) 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

- 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: \( \Theta = \{\theta_1, \theta_2, \Delta\} \)

3. AutoPlait

- Given: bundle \( X \)

- Find: \( r \) regimes (\( r = 2, 3, 4, \ldots \) )

- i.e., find full parameter set

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

- Mocap data
- WebClick data
- Google Trends

Q1. Sense-making

MoCap data

AutoPlait (NO magic numbers)

DynaMMo (Li et al., KDD’09)

pHMM (Wang et al., SIGMOD’11)

Q2. Accuracy

(a) Segmentation

(b) Clustering

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)
App. Event discovery
(GoogleTrend)

Anomaly detection (flu-related topics, 10 years)

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

Turning point detection (seasonal sweets topics)

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

Trend discovery (game-related topics)

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

Industrial contribution
- Automobile sensor data
  - location, velocity, longitudinal/lateral acceleration
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)
• Linear forecasting
  – AR, RLS
• Streaming pattern discovery
  – RLS, “incremental” wavelet transform
  – Multi-scale windows
• Automatic mining
  – MDL

References

• 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.
References


Similarity search, pattern discovery and summarization

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