WindMine: Fast and Effective Mining of Web-click Sequences

Yasushi Sakurai (NTT)
Lei Li (Carnegie Mellon Univ.)
Yasuko Matsubara (Kyoto Univ.)
Christos Faloutsos (Carnegie Mellon Univ.)

Introduction

Web-click sequence applications

Web masters and web-site owners

- Capacity planning
- Intrusion detection
- Advertisement design

Goal

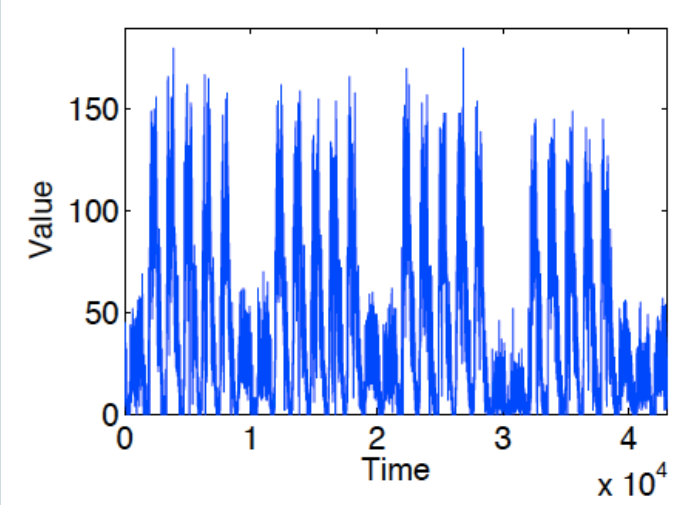
- Find meaningful patterns for web-click data
  (e.g., the lunch-break trend, huge spike, anomalies)
- Find periodicity (daily and/or weekly, etc)
- Determine suitable window sizes automatically

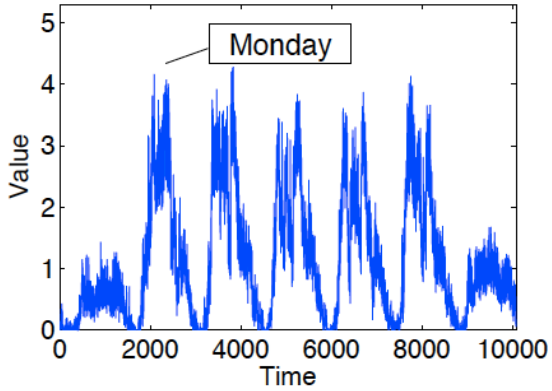
Introduction

Examples

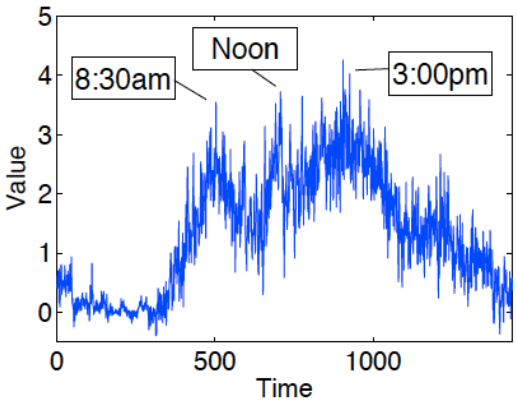
access count from a business news site

Original web-click sequence

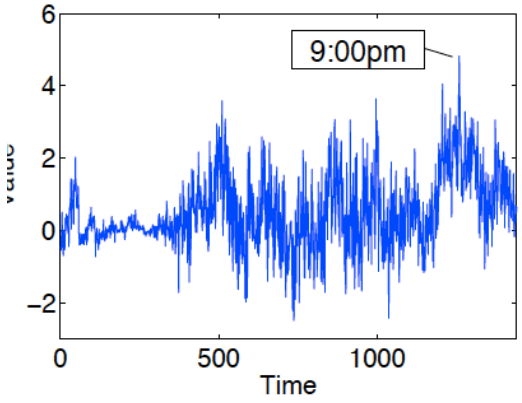




Weekly component



Weekday component



Weekend component

Problem definition

Web-click sequences of m URLs:

$$(X_1, \ldots, X_m)$$

Web-click sequence X of duration n :

$$X = (x_1, \ldots, x_t, \ldots, x_n)$$

Local Component Analysis:

Given m sequences of duration n, $(X_1, \ldots, X_m)$

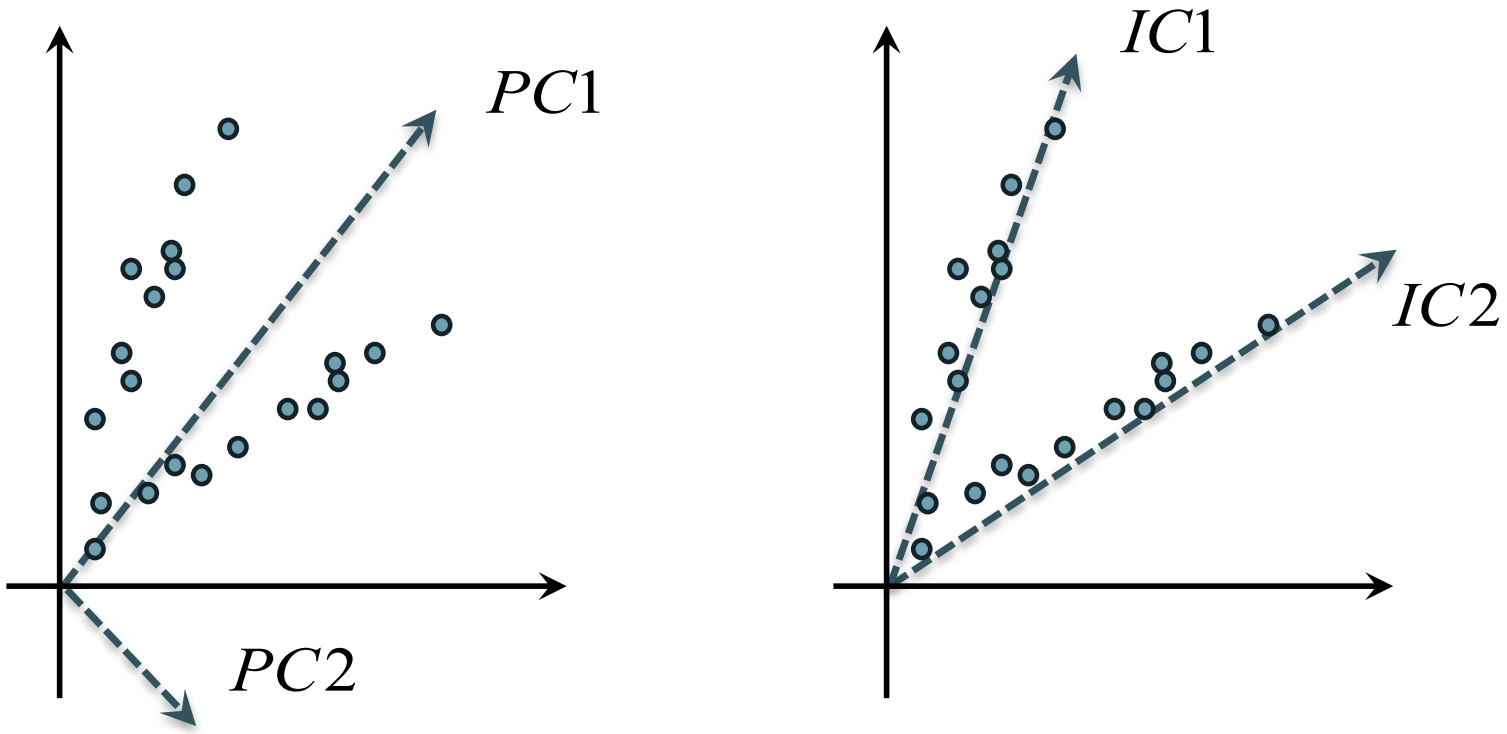
- Find patterns, main components of the sequences
- Find the 'best window' size w for the analysis

Final challenge: scalable algorithm for the local component analysis

Background

Independent component analysis (ICA)

- PCA vs. ICA

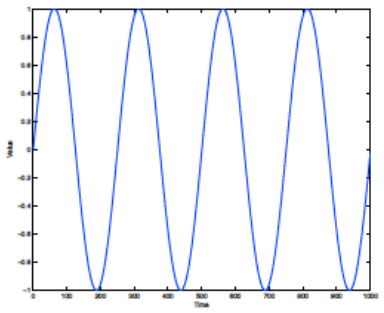


Why not ‘PCA’?

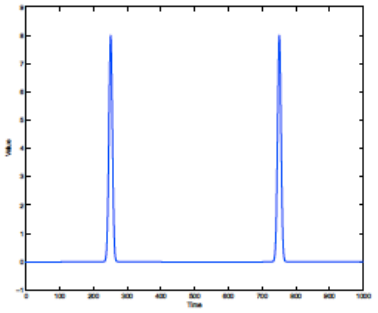
Example of component analysis

Source

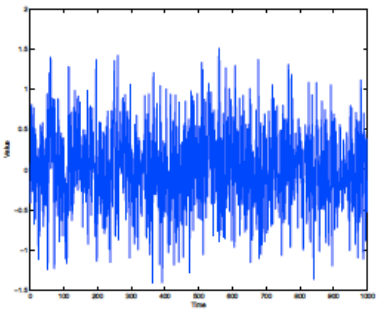
Mix



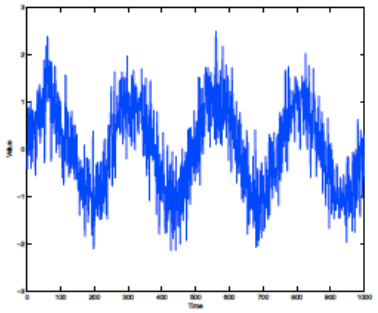
Source #1



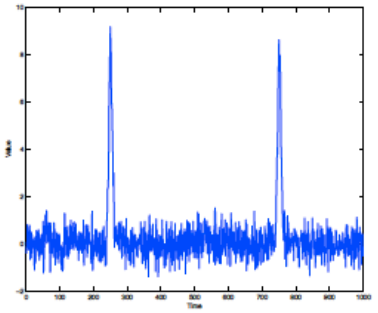
Source #2



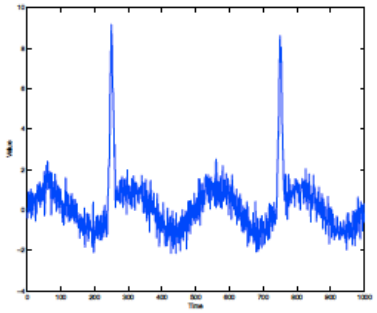
Source #3



Sequence #1
(Sources #1 & #3)



Sequence #2
(Sources #2 & #3)

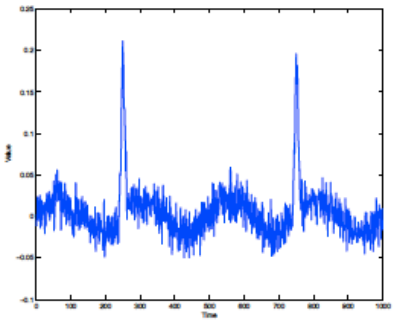


Sequence #3
(Mix of all 3 sources)

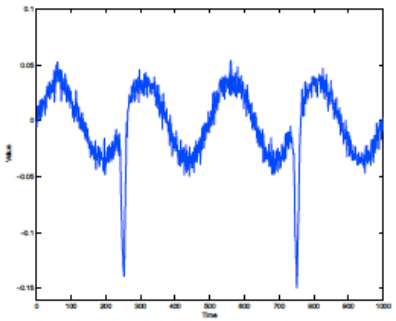
Why not ‘PCA’?

Example of component analysis

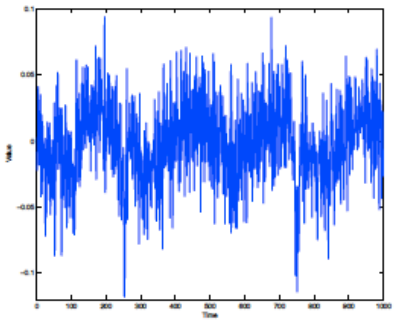
PCA



PC1

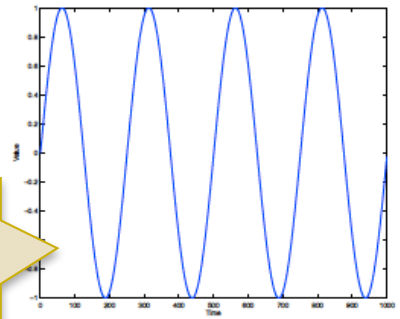


PC2

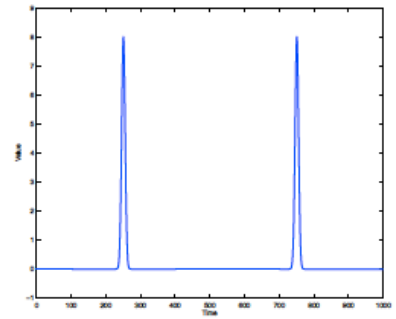


PC3

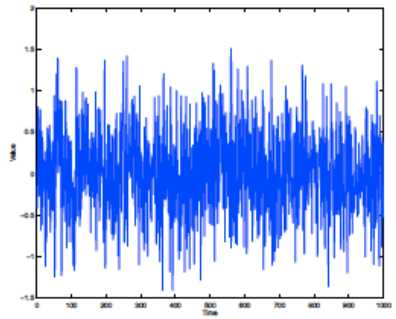
ICA



IC1



IC2

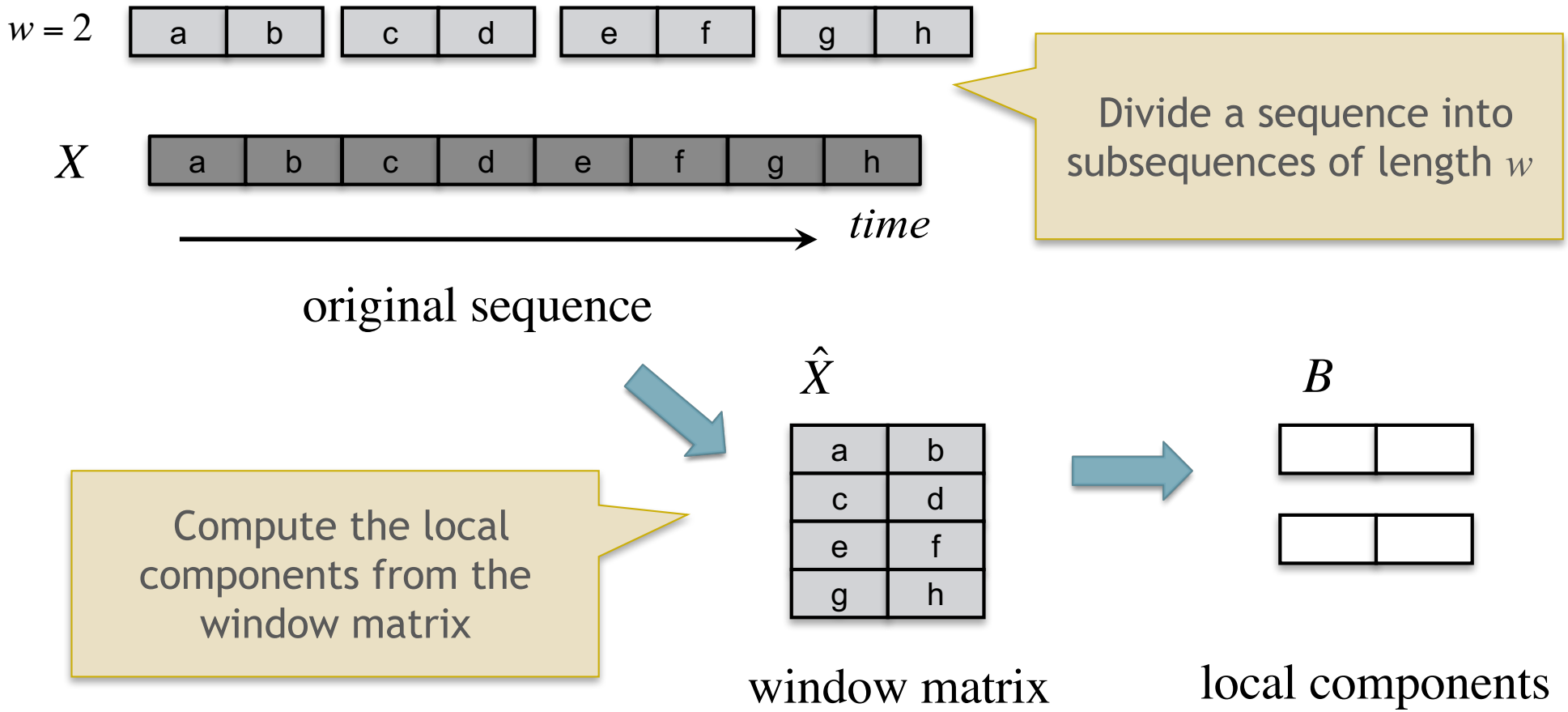


IC3

ICA recognizes the components successfully and separately

Main idea (1)

Multi-scale local component analysis



Main idea (2)

Best window size selection

Q : How to estimate a 'good window size' automatically when we have multiple sequences?

Proposed criterion:

CEM (Component Entropy Maximization)

- Estimate the optimal number of w for the sequence set
- Compute the entropy of the weight values of the mixing matrix A
- 'popular' (widely-used) components show high CEM scores

Main idea (2)

CEM criterion:

- CEM score of the j-th component for the window size w

$$C_{w,j} = -\frac{1}{\sqrt{w}} \sum_i p_{i,j} \log p_{i,j}$$

k: # of components

M: # of subsequences

$$(i = 1, \ldots, M; j = 1, \ldots, k)$$

- Probability for the j-th component (size of the j-th component's contribution to each subsequence)

$$p_{i,j} = \left\| a'_{i,j} \right\| \Big/ \sum_i \left\| a'_{i,j} \right\|$$

- Normalized weight values for each subsequences

$$a'_{i,j} = a_{i,j} \Big/ \sum_j a_{i,j}^2$$

- Mixing matrix $A_w = [a_{i,j}]$

WindMine-part

Efficient solution

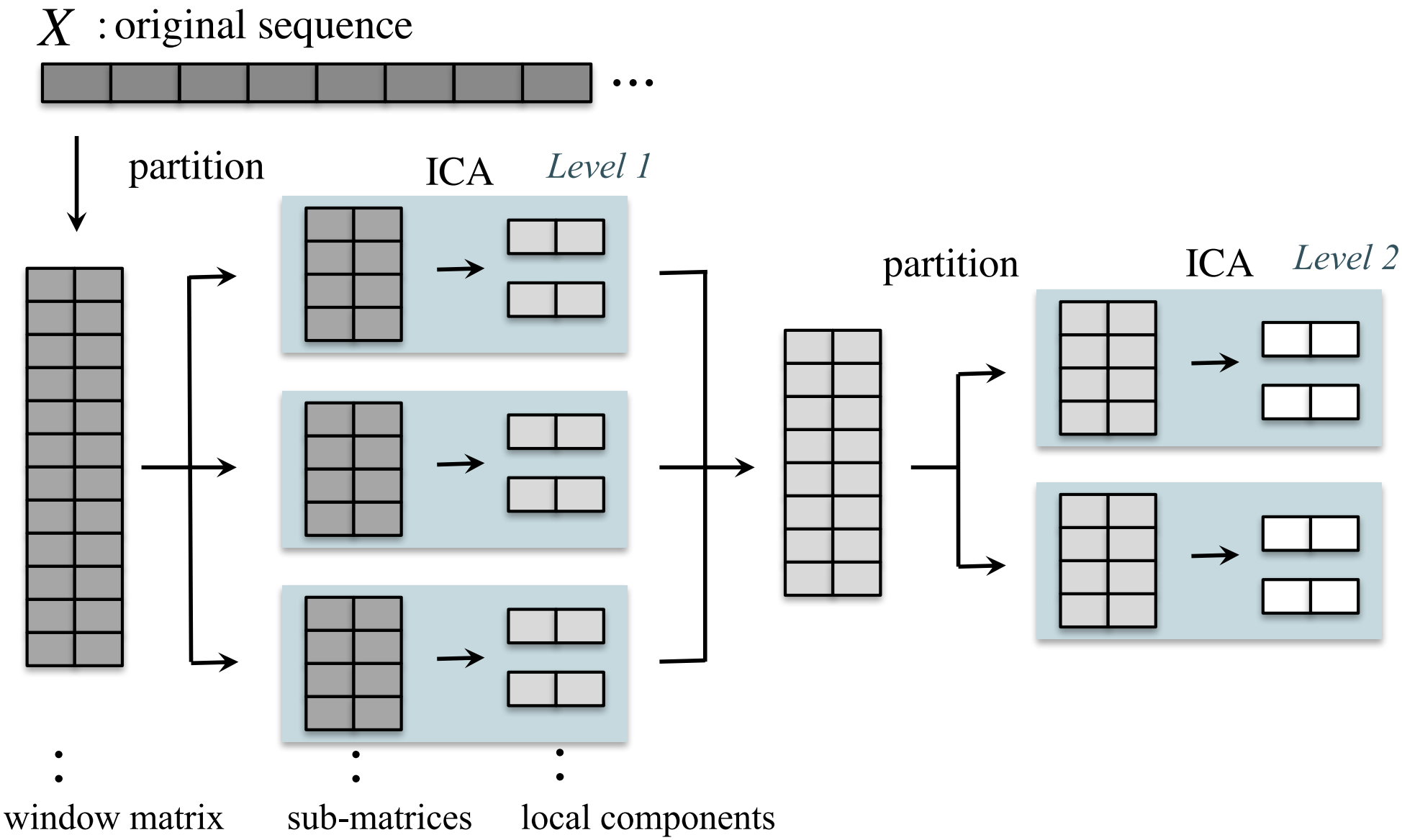
Q : How do we efficiently extract the best local component from large sequence sets?

Hierarchical partitioning approach:

WindMine-part

- Partition the original window matrix into sub-matrices
- Extract local components each from the sub-matrices
- Reuse the local components for the component analysis on the higher level

WindMine-part



Experimental Results

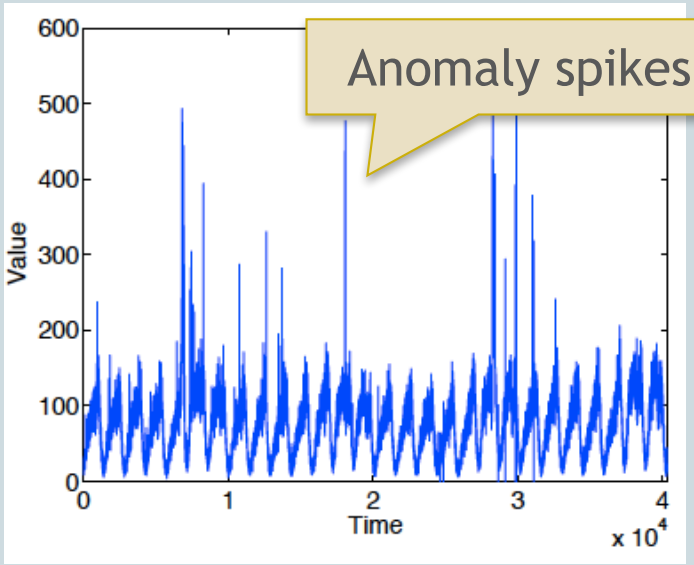
- Experiments with real and datasets
  - *Ondemand TV, WebClick,*
  - *Automobile, Temperature, Sunspots*
- Evaluation
  - Accuracy for pattern discovery
  - Accuracy for the best window size
  - Computation time

Pattern discovery

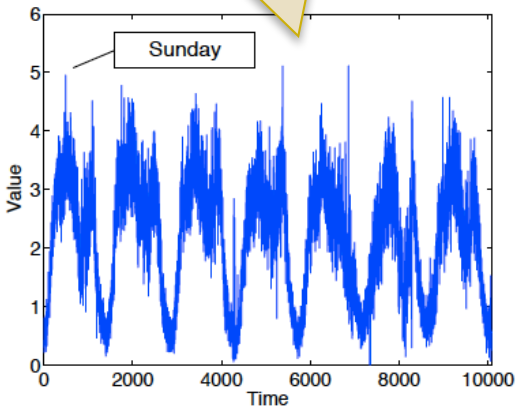
Ondemand TV

access count of users

Original sequence

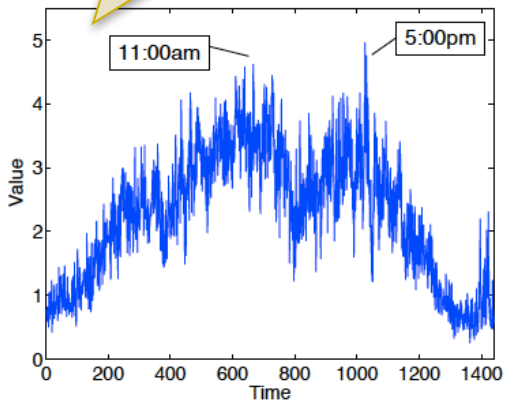


Weekly pattern

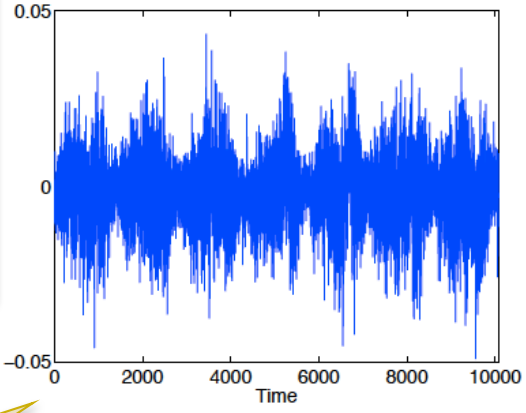


(b) Weekly pattern (*WindMine*)

Daily pattern

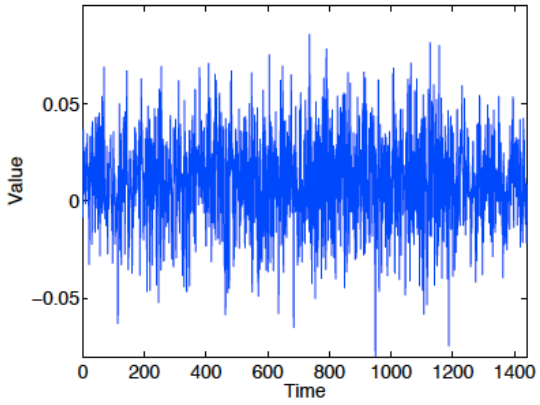


(c) Daily pattern (*WindMine*)



PCA: failed

(d) Weekly pattern (PCA)



(e) Daily pattern (PCA)

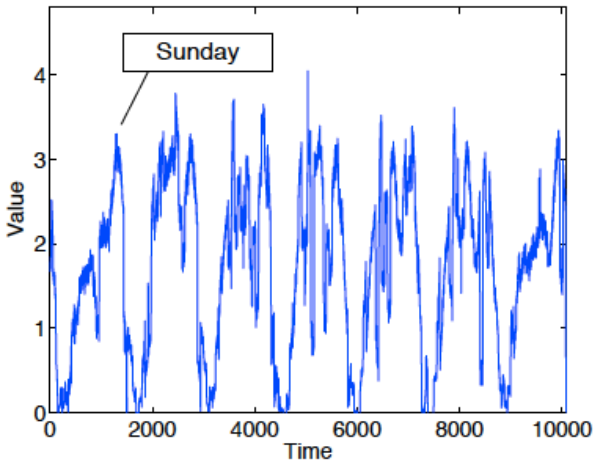
Pattern discovery

WebClick

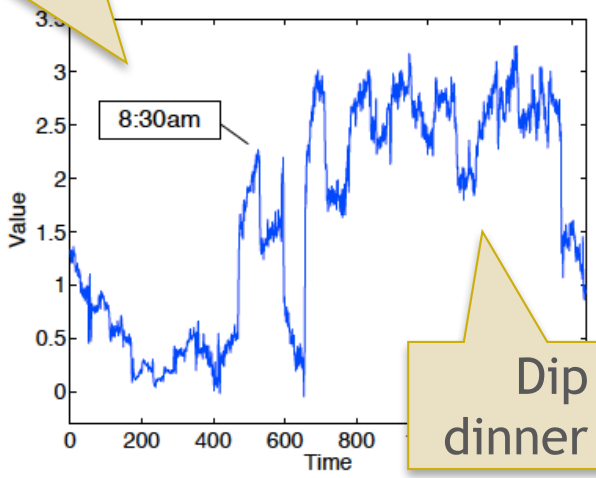
Q & A site

Low activity during sleeping time

Increase from morning to night and reach a peak

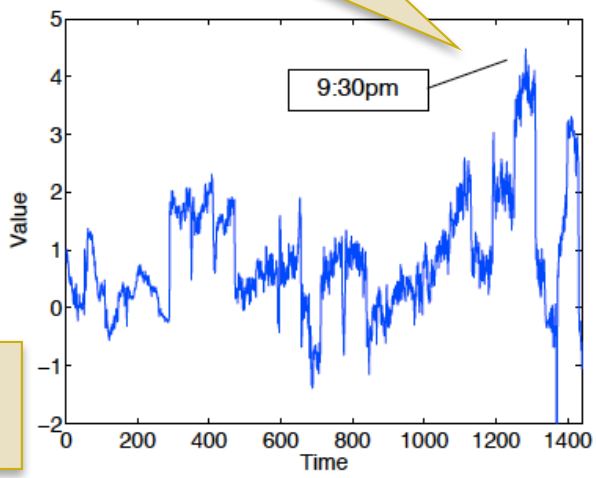


(a) Weekly pattern



(b) Daily pattern

Dip at dinner time



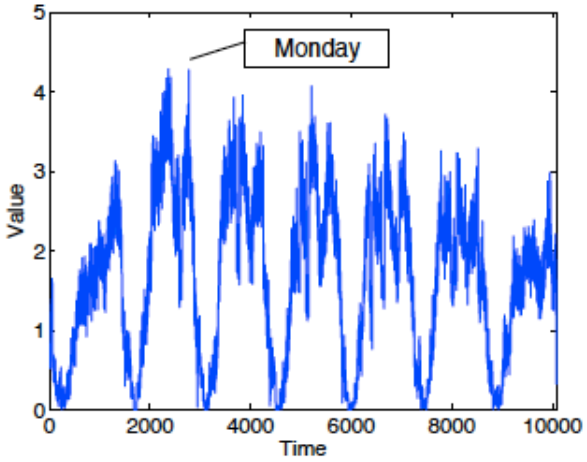
(c) Weekend pattern

Weekly pattern

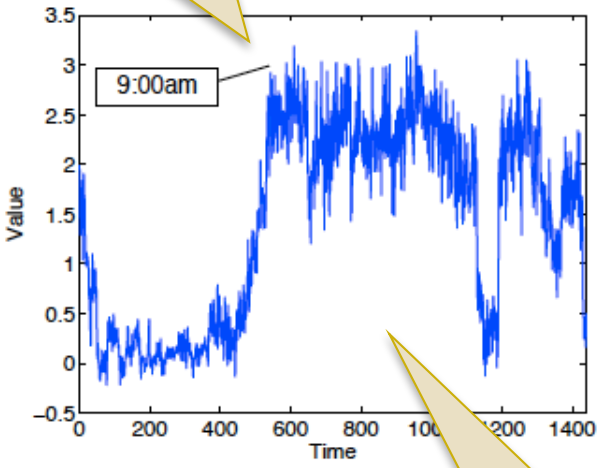
Pattern discovery

WebClick

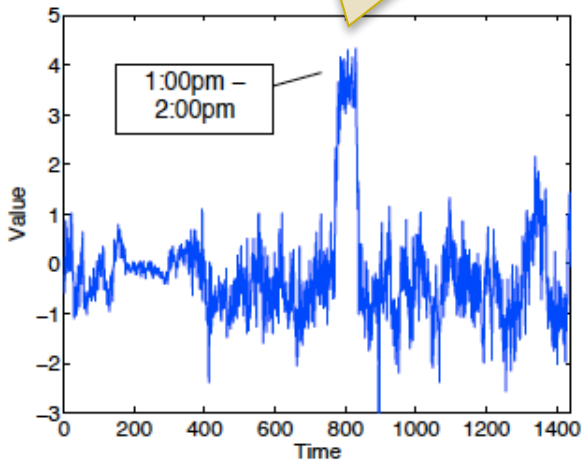
job-seeking site



(a) Weekly pattern



(b) Daily pattern



(c) Weekday additional pattern

Workers arrive at their office

Large spike during the lunch break

High activity on week days (daily access decreases as the weekend approaches)

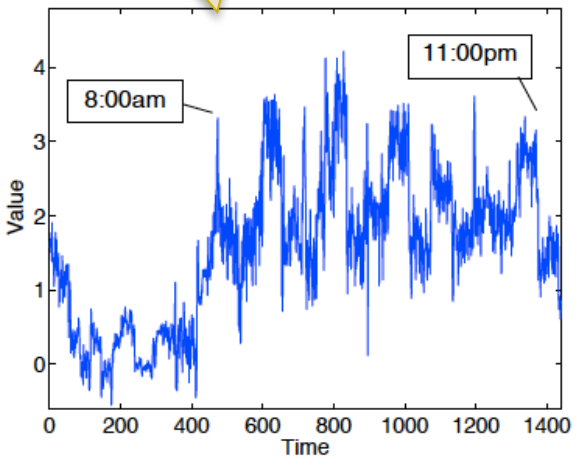
Job seeking during a short break

Pattern discovery

WebClick

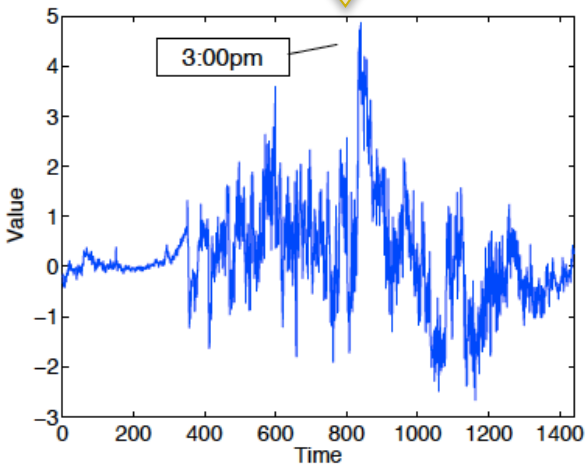
other websites

High activity
8am-11pm, weekday
(business purposes)



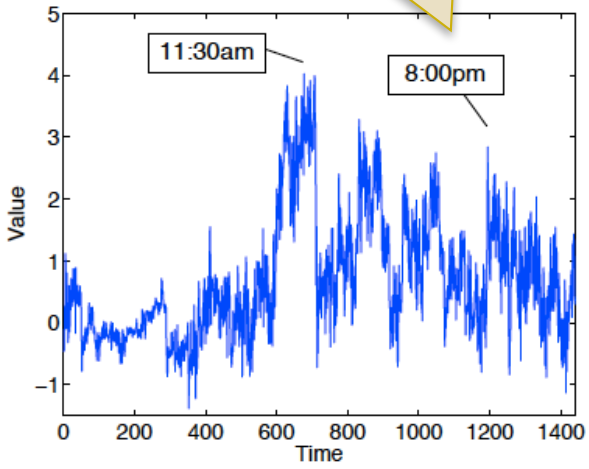
(a) Dictionary

Educational site for kids
(they visit here after
school, 3pm)



(b) Kids

Website for baby nursery
(the main users will be their
parents, rather than babies!)



(c) Baby

Pattern discovery

WebClick

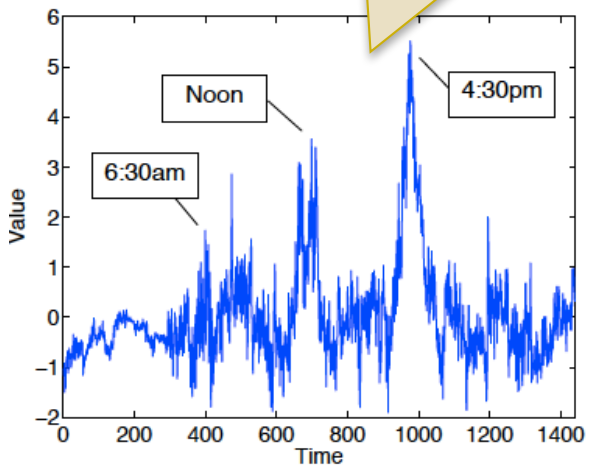
other websites

Access count increases after meal times

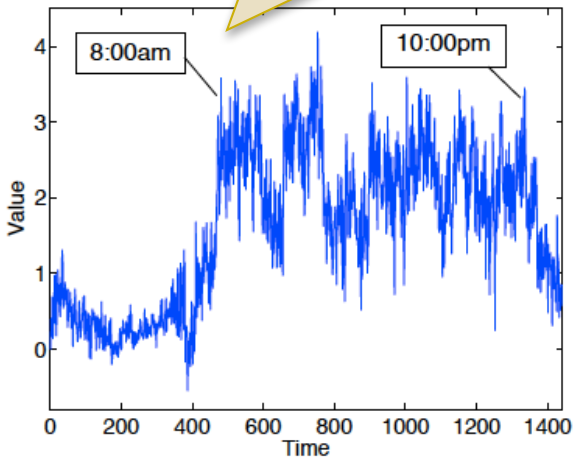
The users visit three times a day (early morning, noon, early evening)

The users rarely visit here late in the evening (which is indeed good for their health!)

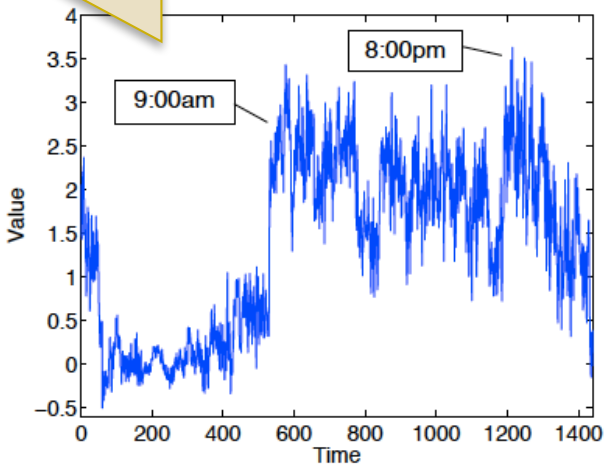
Access count is still high in the night, 0am-1am (healthy diet should include an earlier bed time!)



(d) Weather news



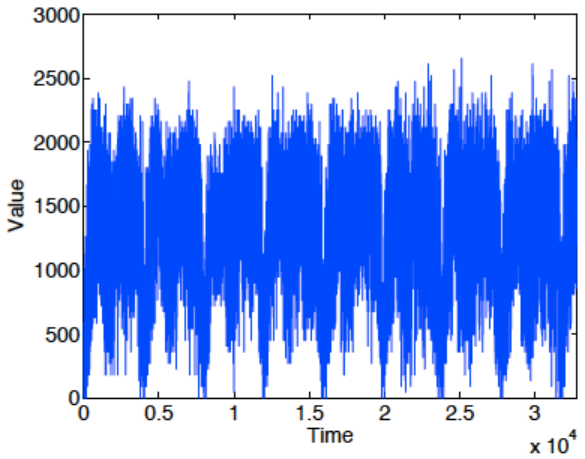
(e) Health

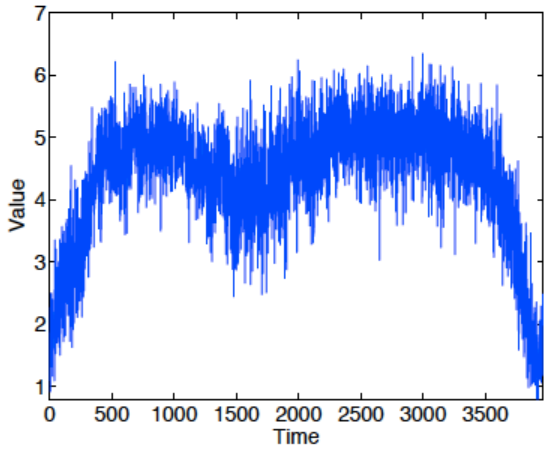


(f) Diet

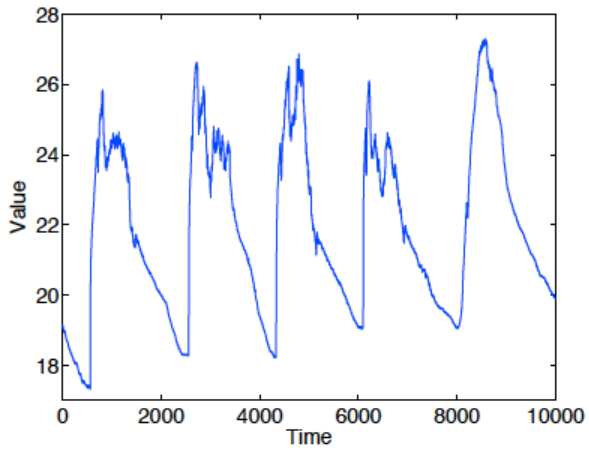
Pattern discovery

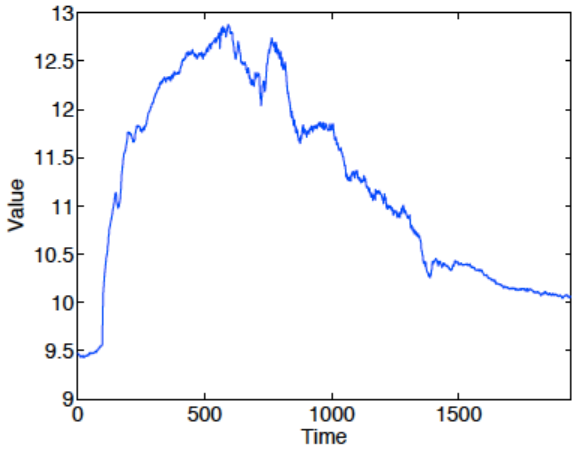
Generalization of WindMine



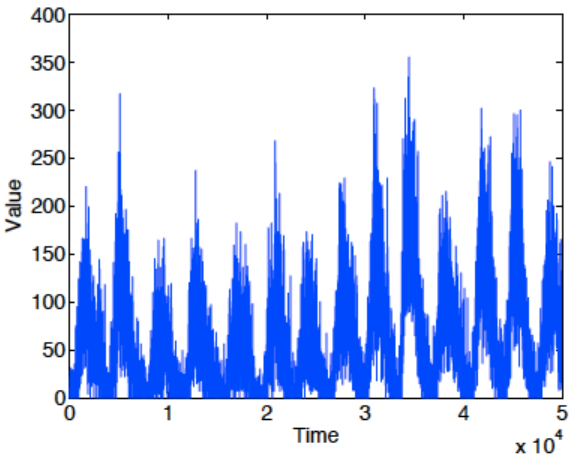


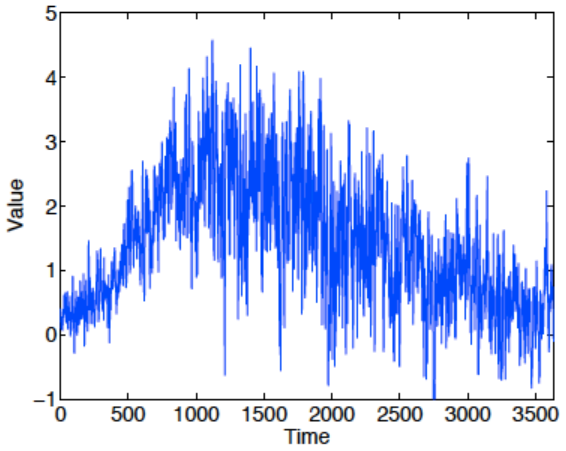
(a) *Automobile*





(b) *Temperature*

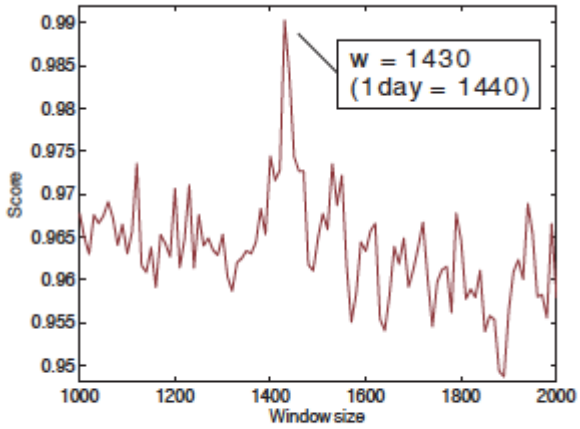




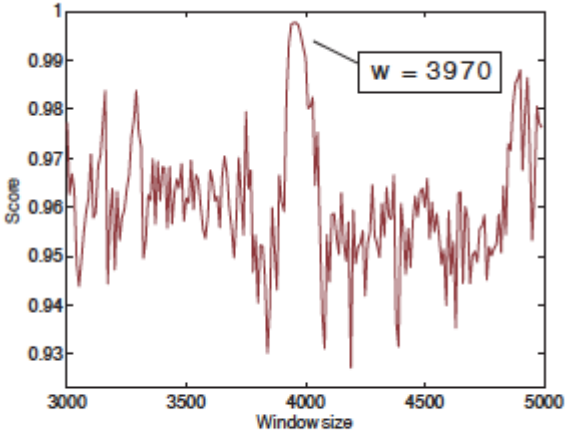
(c) *Sunspot*

Choice of best window size

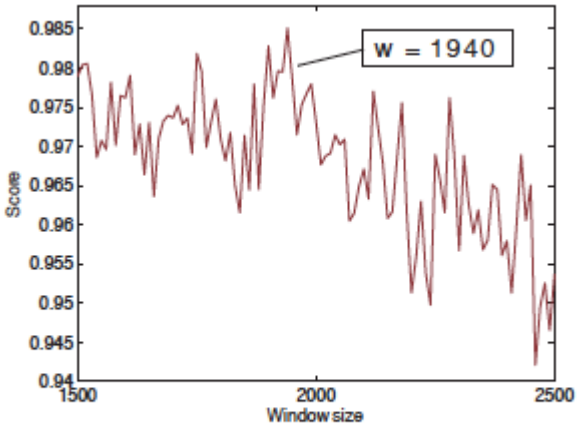
CEM score for various window sizes



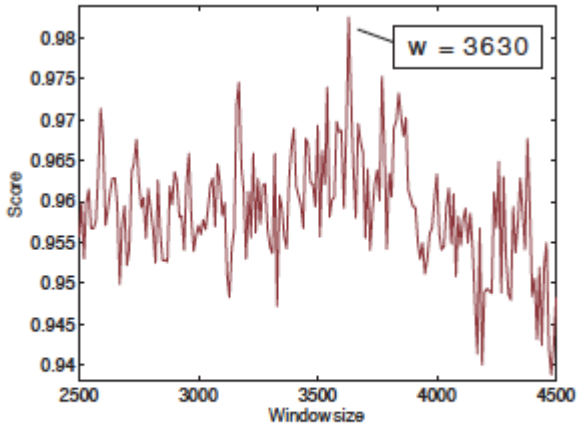
(a) *Ondemand TV*



(b) *Automobile*

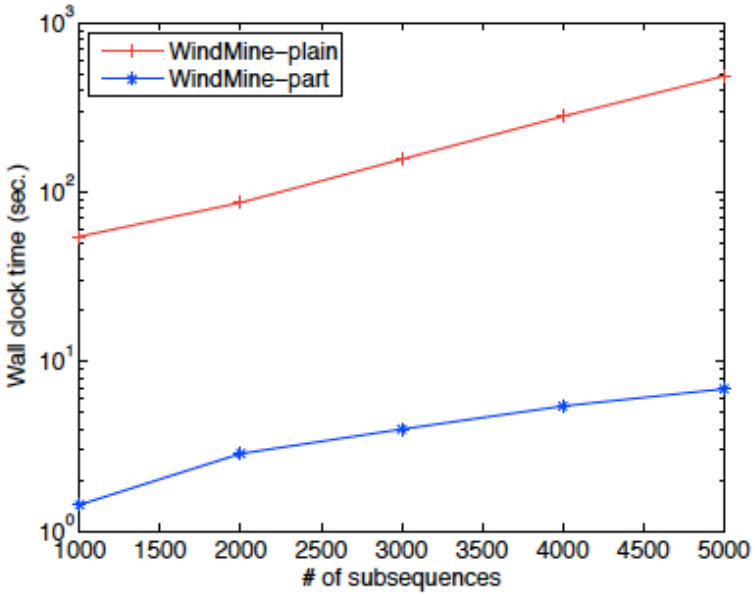


(c) *Temperature*

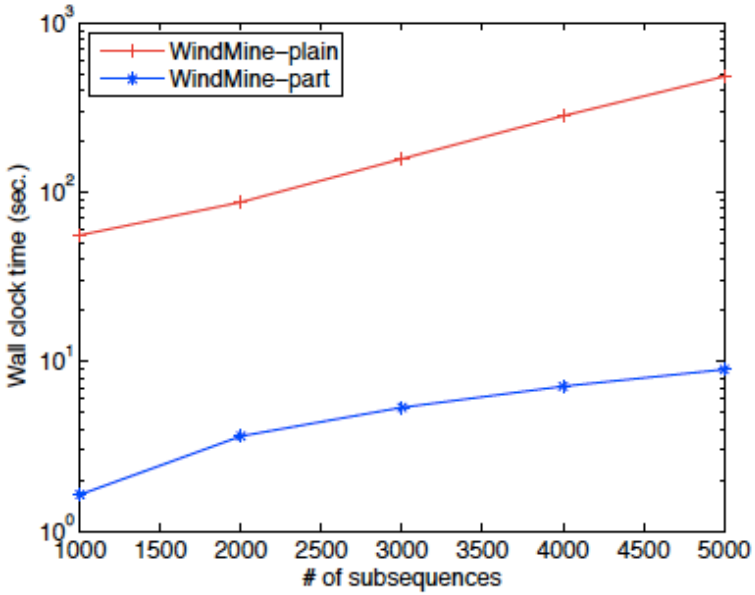


(d) *Sunspot*

Computation time



Ondemand TV

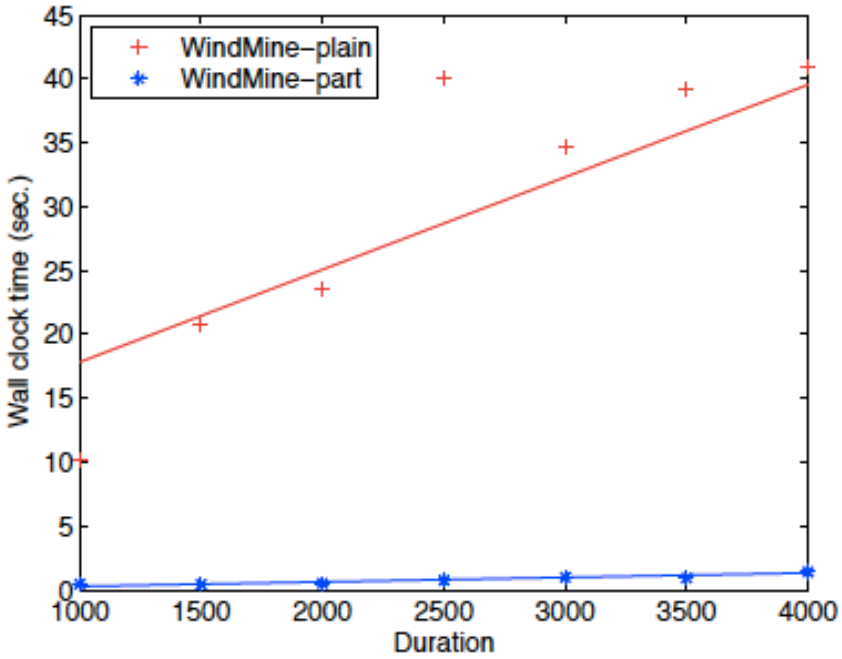


WebClick

Wall clock time vs. # of subsequences

- Up to 70 times faster

Computation time



Ondemand TV

Wall clock time vs. duration

Conclusions

Scalable pattern extraction and anomaly detection in large web-click sequences

1. Scalable, parallelizable method for breaking sequences into a few, fundamental ingredients
2. Linearly over the sequence duration, and near-linearly on the number of sequence