Streaming Pattern Discovery in Multiple Time-Series

Spiros Papadimitriou
Jimeng Sun
Christos Faloutsos

Carnegie Mellon University

VLDB 2005, Trondheim, Norway

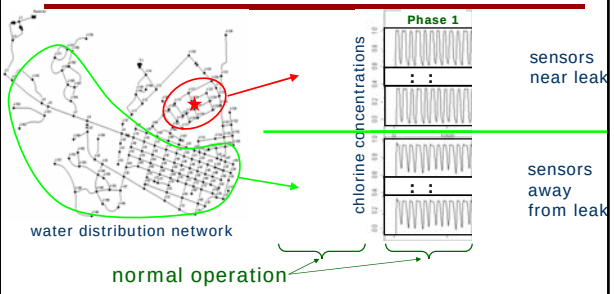
Motivation

- Several settings where many deployed sensors measure some quantity—e.g.:
  - Traffic in a network
  - Temperatures in a large building
  - Chlorine concentration in water distribution network

Values are typically correlated

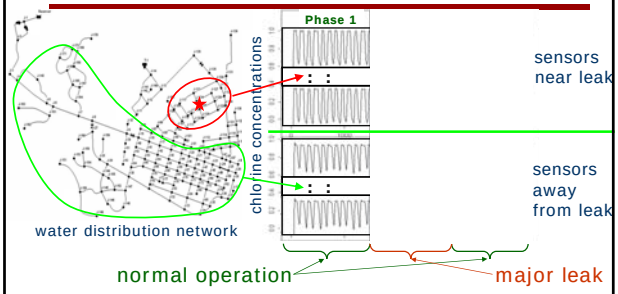
Would be very useful
if we could summarize them on the fly

Motivation



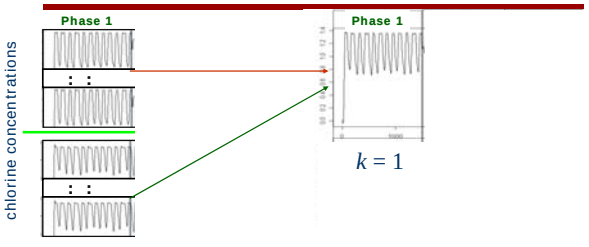
May have hundreds of measurements, but
it is **unlikely they are completely unrelated**!

Motivation



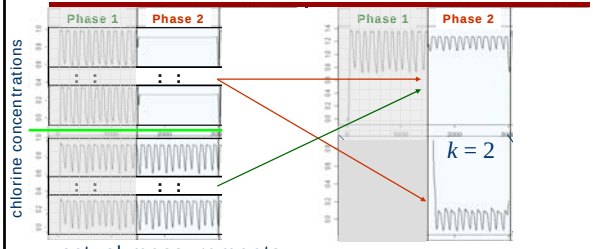
May have hundreds of measurements, but
it is **unlikely they are completely unrelated**!

Motivation



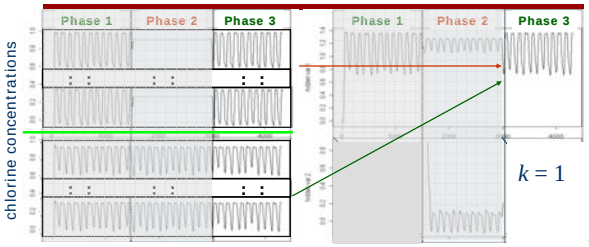
We would like to discover a few "hidden
(latent) variables" that summarize the key trends

Motivation



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

Motivation



actual measurements
(*n* streams)

k hidden variable(s)

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

7

Goals

- Discover "hidden" (latent) variables for:
  - Summarization of main trends for users
  - Efficient forecasting, spotting outliers/anomalies
- Incremental, real-time computation
- Limited memory requirements

8

Related work
Stream mining

- Stream SVD [Guha, Gunopulos, Koudas / KDD03]
- StatStream [Zhu, Shasha / VLDB02]
- Clustering
  [Aggarwal, Han, Yu / VLDB03], [Guha, Meyerson, et al / TKDE], [Lin, Vlachos, Keogh, Gunopulos / EDBT04],
- Classification
  [Wang, Fan, et al / KDD03], [Hulten, Spencer, Domingos / KDD01]
- Piecewise approximations
  [Palpanas, Vlachos, Keogh, etal / ICDE 2004]
- Queries on streams
  [Dobra, Garofalakis, Gehrke, et al / SIGMOD02], [Madden, Franklin, Hellerstein, et al / OSDI02], [Considine, Li, Kollios, et al / ICDE04], [Hammad, Aref, Elmagarmid / SSDBM03]
- ...

9

Overview

- **Method outline**
- Experiments

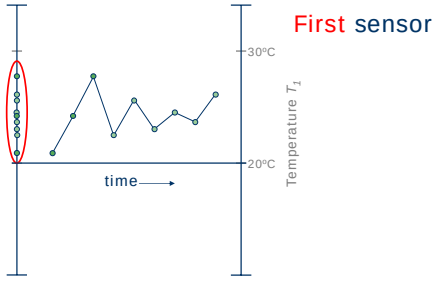
10

Stream correlations

- **Step 1:** How to capture correlations?
- Step 2: How to do it incrementally, when we have a very large number of points?
- Step 3: How to dynamically adjust the number of hidden variables?

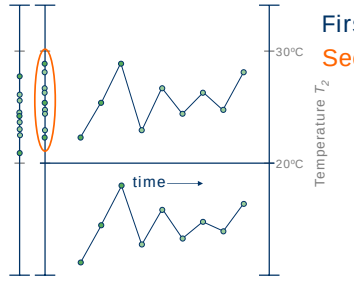
11

1. How to capture correlations?



12

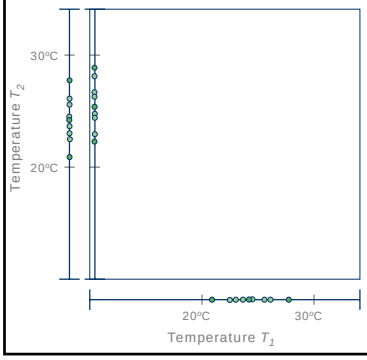
1. How to capture correlations?



First sensor
Second sensor

13

1. How to capture correlations

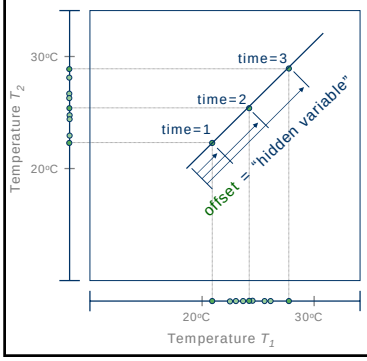


Correlations:

Let's take a closer look at the first three value-pairs…

14

1. How to capture correlations

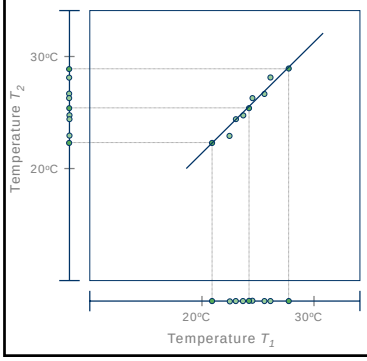


First three lie (almost) on a line in the space of value-pairs…

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

15

1. How to capture correlations



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

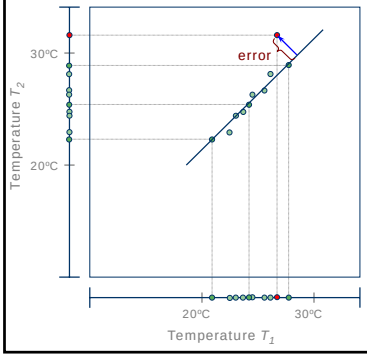
16

Stream correlations

- Step 1: How to capture correlations?
- **Step 2:** How to do it incrementally, when we have a very large number of points?
- Step 3: How to dynamically adjust the number of hidden variables?

17

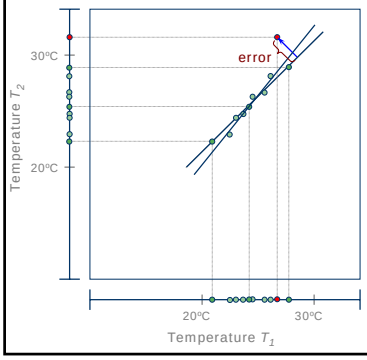
2. Incremental update



For each new point
- Project onto current line
- Estimate error

18

2. Incremental update

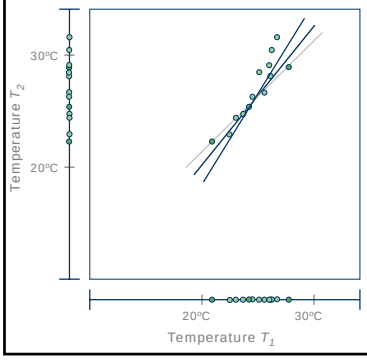


For each new point
- Project onto current line
- Estimate error
- Rotate line in the direction of the error and in proportion to its magnitude

➔ O(*n*) time

19

2. Incremental update

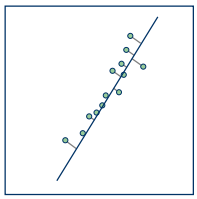


For each new point
- Project onto current line
- Estimate error
- Rotate line in the direction of the error and in proportion to its magnitude

20

Stream correlations
Principal Component Analysis (PCA)

- The "line" is the first *principal component (PC)* vector
- This line is optimal: it minimizes the sum of squared *projection* errors



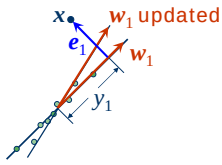
21

2. Incremental update
Given number of hidden variables *k*

- Assuming *k* is known
- We know how to update the slope (detailed equations in paper)

For each new point $\boldsymbol{x}$ and for $i = 1, \ldots, k$:

- $y_i := \boldsymbol{w}_i^T \boldsymbol{x}$ (proj. onto $\boldsymbol{w}_i$)
- $d_i \leftarrow \lambda d_i + y_i^2$ (energy $\propto$ i-th eigenval.)
- $\boldsymbol{e}_i := \boldsymbol{x} - y_i \boldsymbol{w}_i$ (error)
- $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i + (1/d_i)\, y_i \boldsymbol{e}_i$ (update estimate)
- $\boldsymbol{x} \leftarrow \boldsymbol{x} - y_i \boldsymbol{w}_i$ (repeat with remainder)



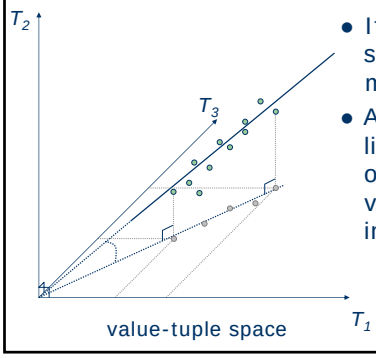
22

Stream correlations

- Step 1: How to capture correlations?
- Step 2: How to do it incrementally, when we have a very large number of points?
- **Step 3:** How to dynamically adjust *k*, the number of hidden variables?

23

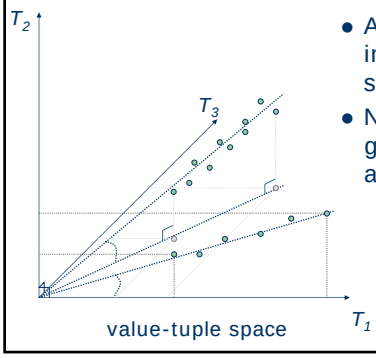
3. Number of hidden variables



- If we had three sensors with similar measurements
- Again: points would lie on a line (i.e., one hidden variable, k=1), but in 3-D space

24

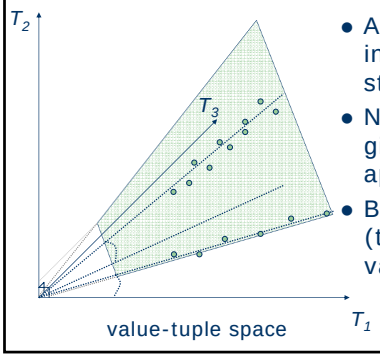
3. Number of hidden variables



- Assume one sensor intermittently gets stuck
- Now, no **line** can give a good approximation

25

3. Number of hidden variables



- Assume one sensor intermittently gets stuck
- Now, no **line** can give a good approximation
- But a **plane** will do (two hidden variables, ***k* = 2**)

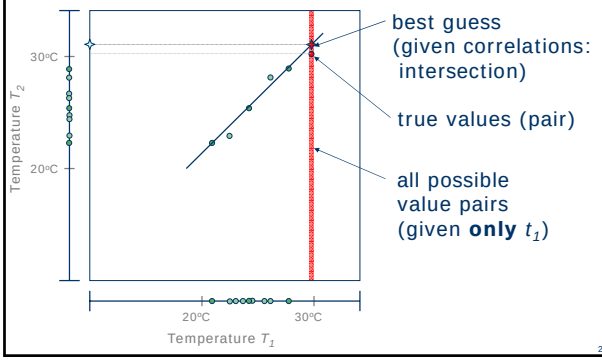
26

Number of hidden variables (PCs)

- Keep track of energy maintained by approximation with k variables (PCs):
  - Reconstruction accuracy, w.r.t. total squared error
- Increment (or decrement) k if fraction of energy maintained goes below (or above) a threshold
  - If below 95%, $k \leftarrow k + 1$
  - If above 98%, $k \leftarrow k - 1$

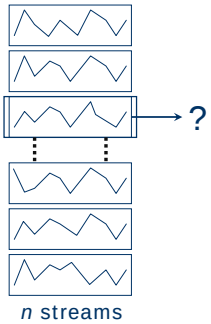
27

Missing values



28

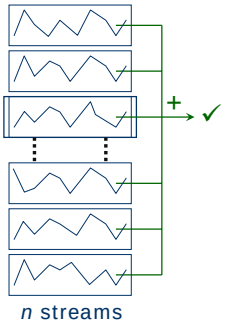
Forecasting



- Assume we want to forecast the next value for a particular stream (e.g. auto-regression)

29

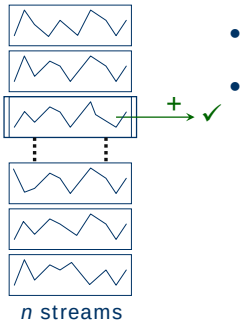
Forecasting



- Option 1: One complex model per stream
  - Next value = function of previous values on ***all*** streams
  - Captures correlations
  - Too costly! [~ O(n^3)]

30

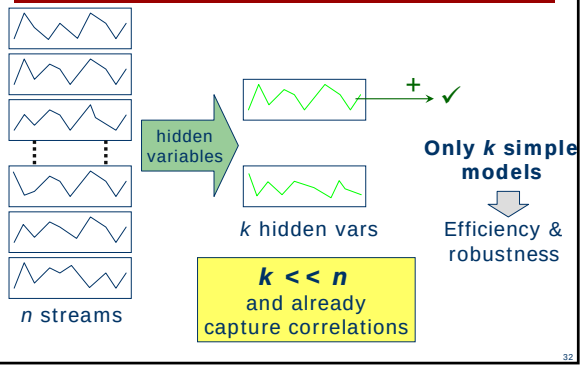
Forecasting



- Option 1: One complex model per stream
- Option 2: One simple model per stream
  - Next value = function of previous value on ***same*** stream
  - Worse accuracy, but maybe acceptable
  - But, still need *n* models

31

Forecasting



32

Time/space requirements
Incremental PCA

O(*nk*) space (total) and time (per tuple), i.e.,

- Independent of # points (*t*)
- Linear w.r.t. # streams (*n*)
- Linear w.r.t. # hidden variables (*k*)

In fact,

- Can be done in **real time** [demo]

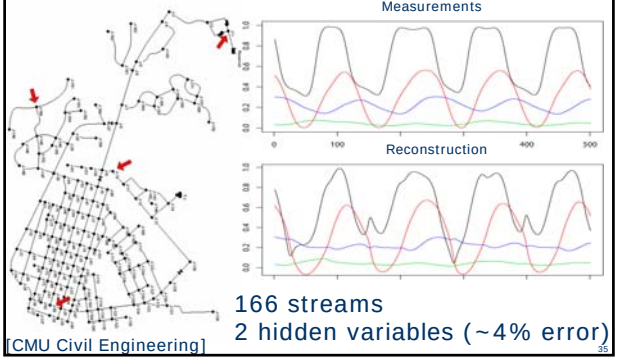
33

Overview

- Method outline
- **Experiments**

34

Experiments
Chlorine concentration

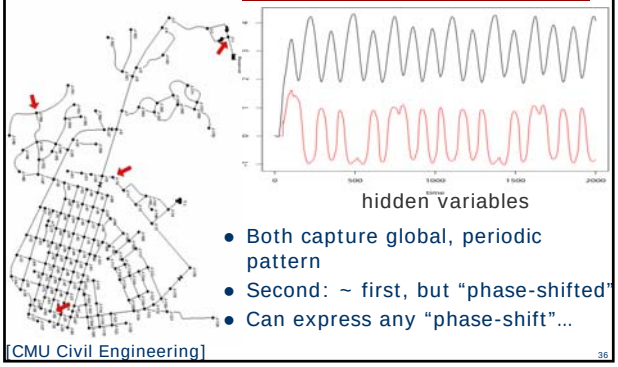


166 streams
2 hidden variables (~4% error)

[CMU Civil Engineering]

35

Experiments
Chlorine concentration

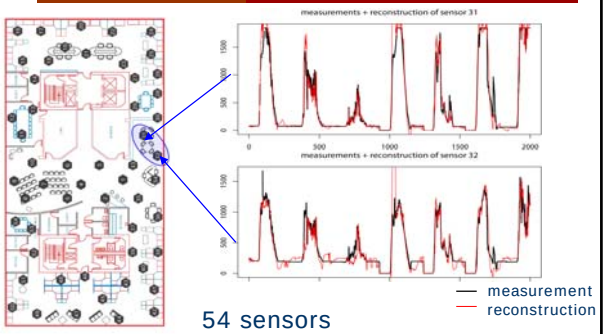


- Both capture global, periodic pattern
- Second: ~ first, but "phase-shifted"
- Can express any "phase-shift"…

[CMU Civil Engineering]

36

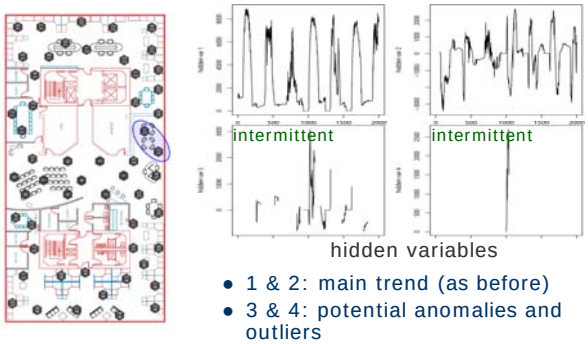
Experiments
Light measurements



54 sensors
2-4 hidden variables (~6% error)

37

Experiments
Light measurements

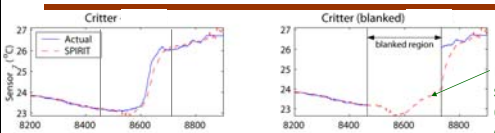


hidden variables

- 1 & 2: main trend (as before)
- 3 & 4: potential anomalies and outliers

38

Experiments
Missing values



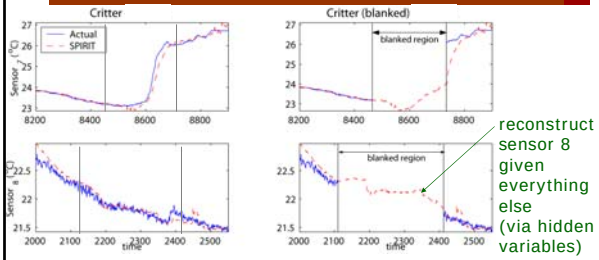
reconstruct sensor 7 given everything else (via hidden variables)

- Correlations already captured by hidden variables
- Provide information about missing values
  - Quickly back on track, if mis-estimated

[CMU ECE]

39

Experiments
Missing values



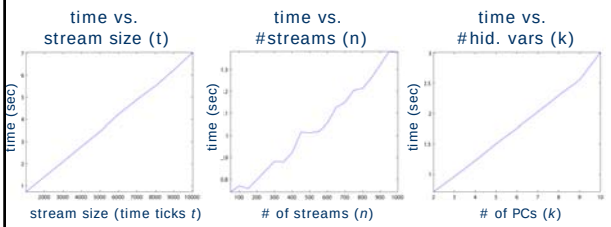
reconstruct sensor 8 given everything else (via hidden variables)

- Correlations already captured by hidden variables
- Provide information about missing values
  - Quickly back on track, if mis-estimated

[CMU ECE]

40

Wall-clock times



constant time per tuple and per stream

41

Conclusion

- Many settings with *hundreds* of streams, **but**
  - Stream values are, by nature, related
  - In reality, there are only a few variables

✓ Discover hidden variables for
  - Summarization of main trends for users
  - Efficient forecasting, spotting outliers/anomalies

✓ Incremental, real time computation

✓ With limited memory

42