Fast Algorithms for Mining and Summarizing Co-evolving Sequences

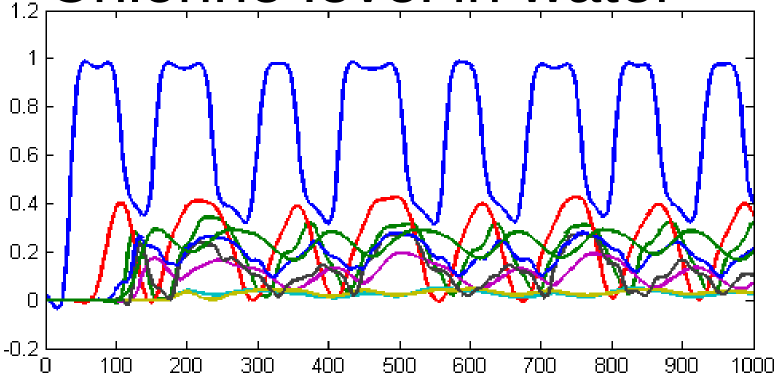
Lei Li

Computer Science Department

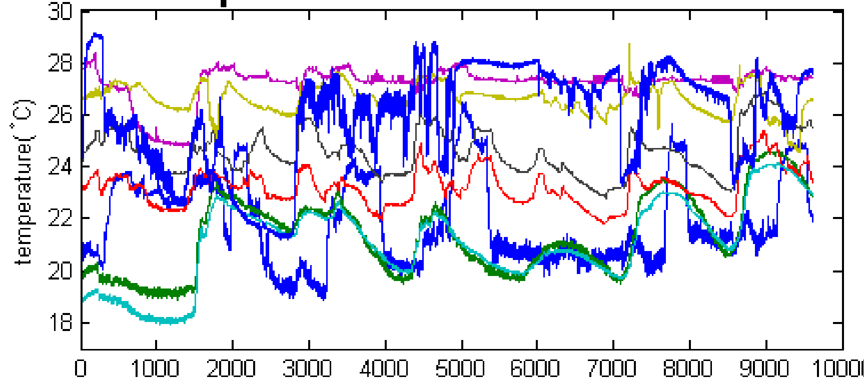
Carnegie Mellon University

Time Series (TS)

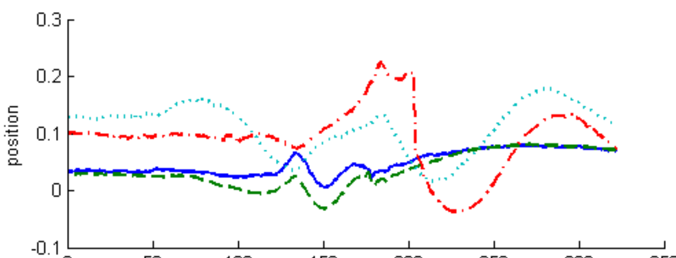
Chlorine level in water



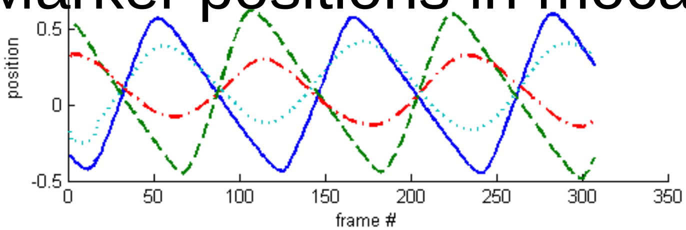
Temperature in datacenter



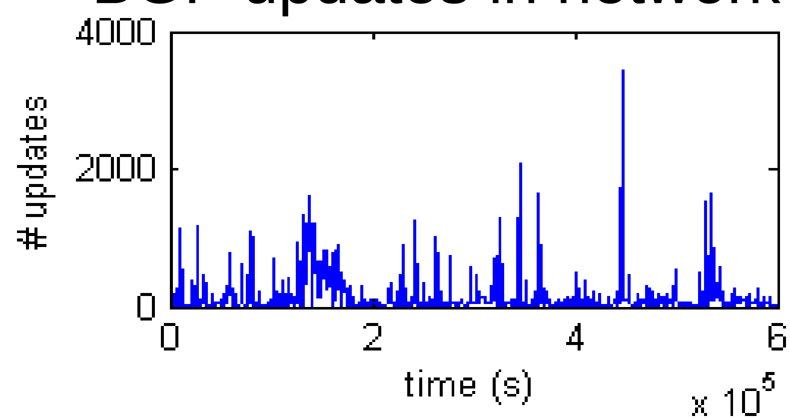
Need fast algorithms for time series mining



Marker positions in mocap



BGP updates in network

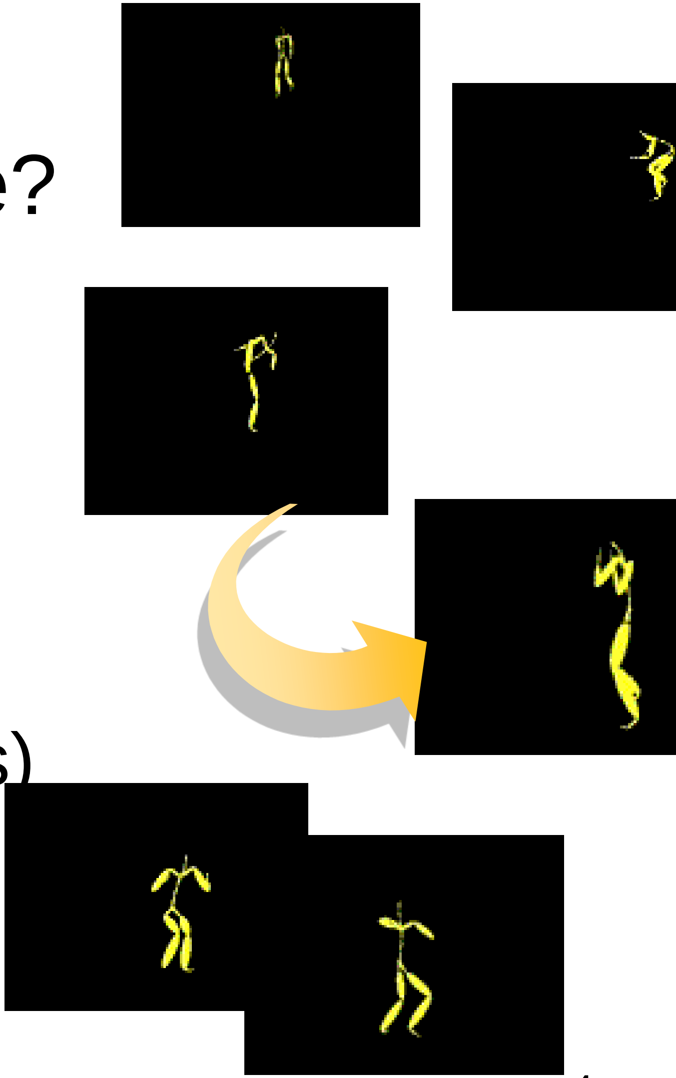


Outline

- Motivation
  - Mining tasks, goals
- Background
  - Principal Component Analysis (PCA)
  - Kalman Filters
- P1:Mining w/ Missing Value [Li+ 2009]
- P2:Parallel Learning [Li+ 2008b]
- Conclusion

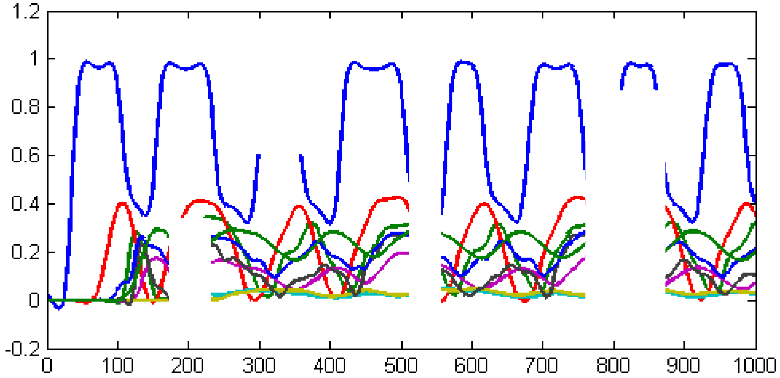
M1: Natural Motion Generation

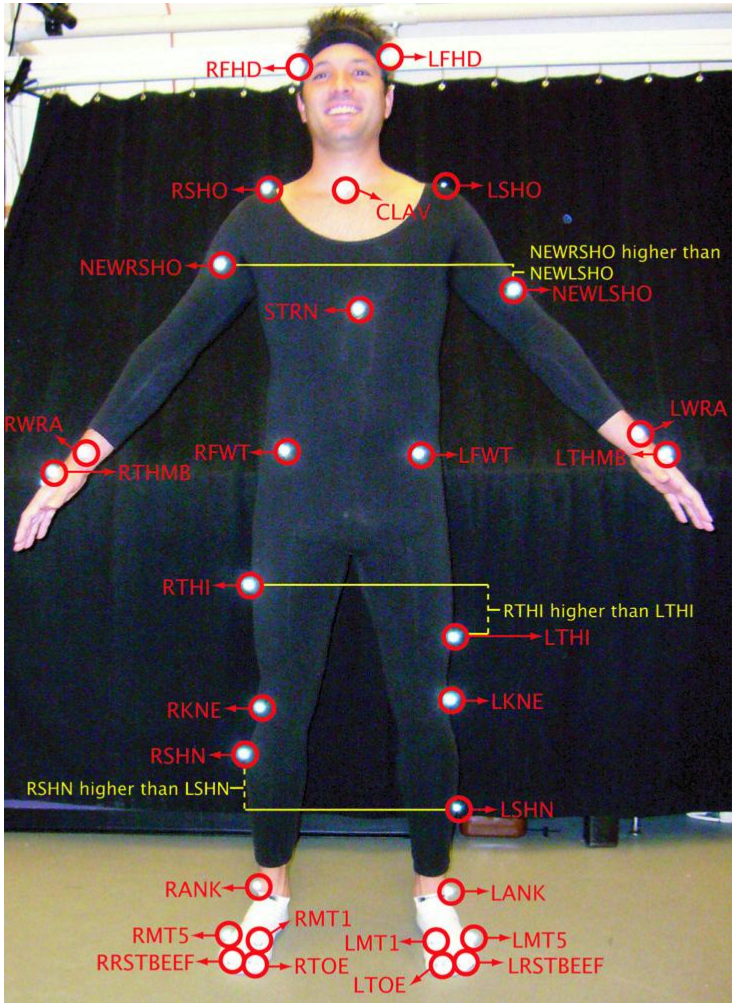
- How to generate new *realistic* motions from mocap database?
- e.g. “karate kick” → “boxing”
- Applications:
  - Game ($57billion 2009)
  - Movie animation
  - Quality of Life (assistive devices)



M2: Missing Values

- How to recover missing values?
  - Occlusion in mocap
  - In sensor data, due to low battery, RF error

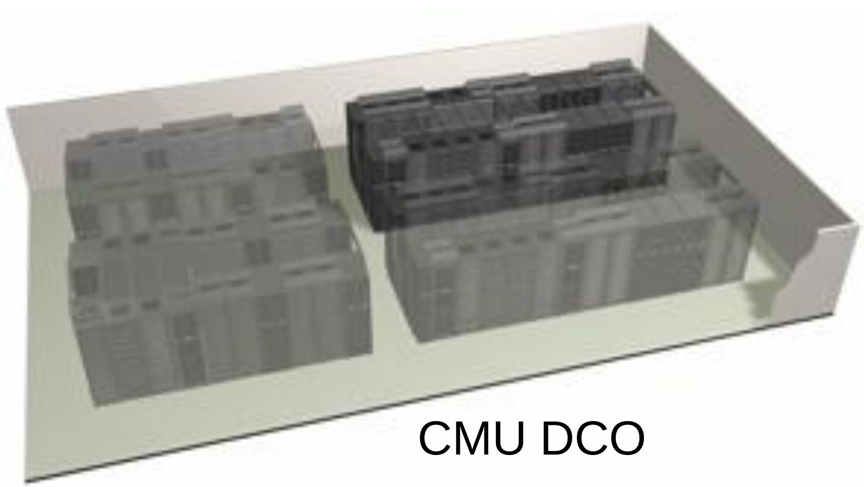




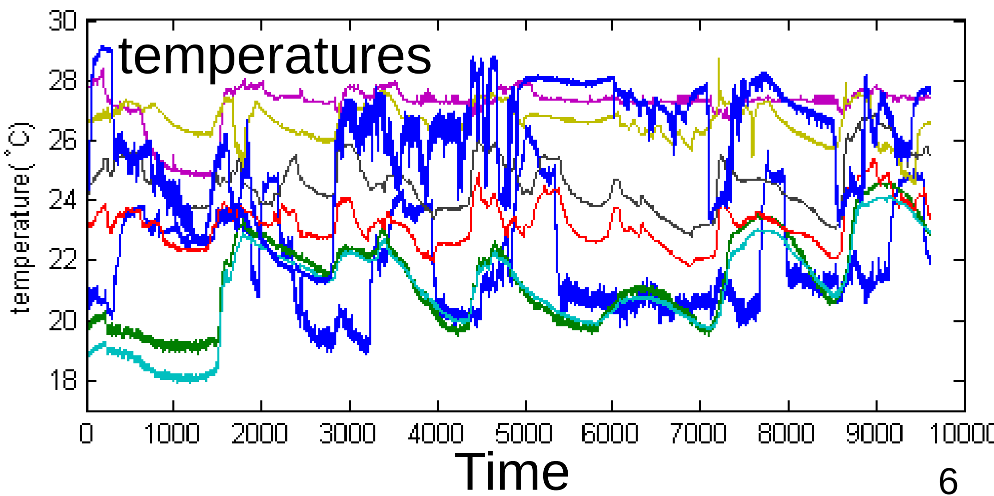
From mocap.cs.cmu.edu

M3: Data Summarization

- How to compress & manage large time series?
  - A datacenter with 5000 servers: 1TB data per day, 55 million streams ([Reeves+ 2009])
- Goal: save energy in data center
  - $4.5billion power for US dc's 2006

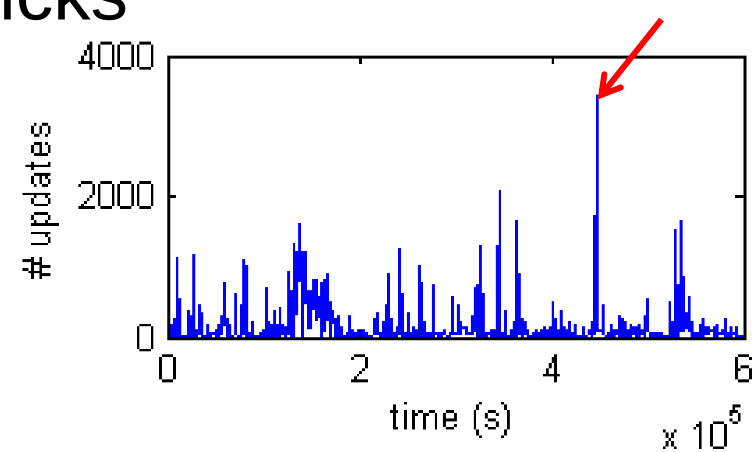


CMU DCO



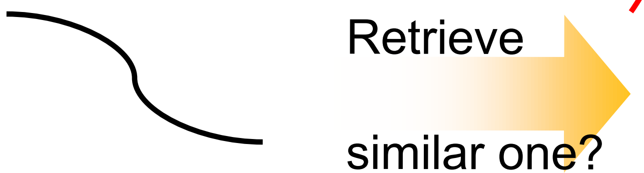
M4: Anomaly Detection

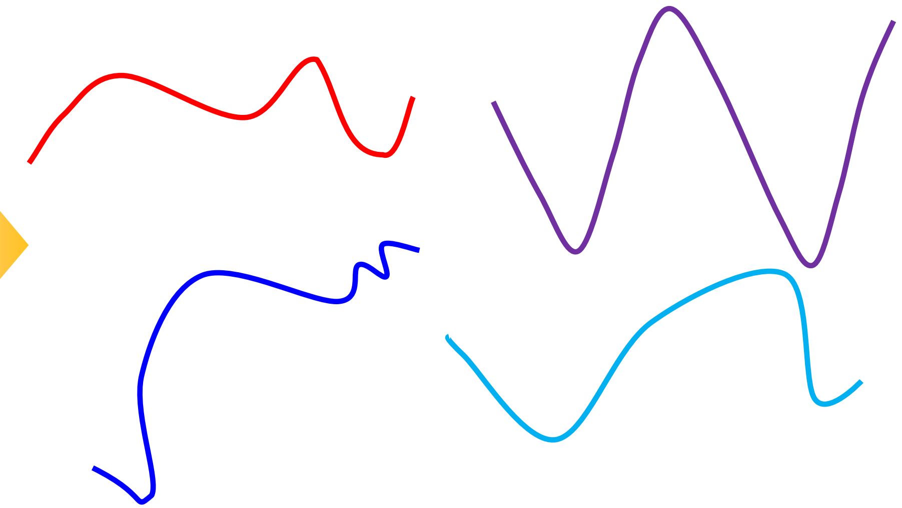
- How to detect anomalies?
- Applications:
  - Intrusion computer network traffic (e.g. # of packets)
  - Detect leakage or attack in drinking water system by monitoring chlorine levels
  - Spam/robot in web clicks



M5: Similarity Queries

- What is the most similar sequences from a large time series database?

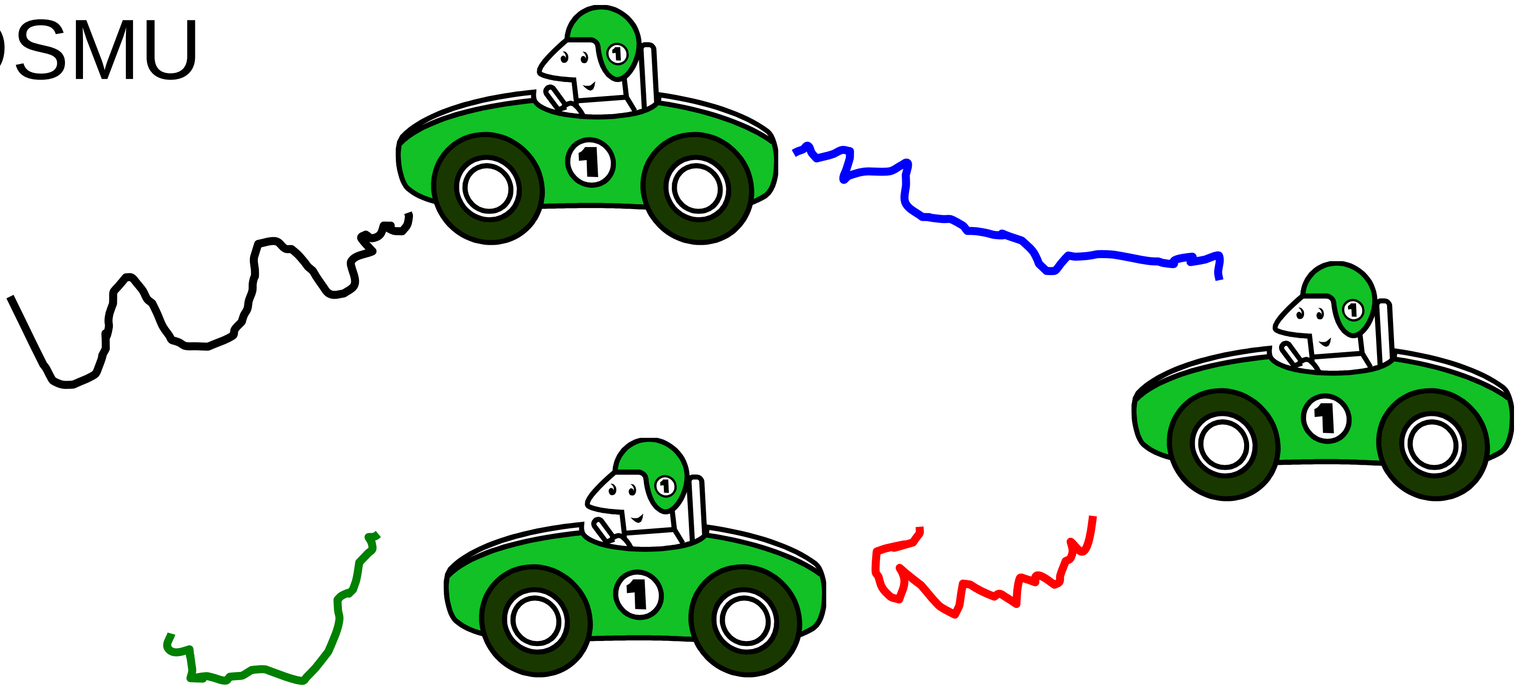




- Applications:
  - Environmental monitoring
  - Datacenter monitoring
  - Motion capture (mocap) database

M6: Trajectory Mining

- Mining moving objects, by Rajesh BALAN, Kyriakos MOURATIDIS, David LO @SMU



Time Series Mining Tasks

- Pattern Discovery (e.g. cross-correlation, lag-correlation)
  - T1:Forecasting
  - T2:Summarization
  - T3:Segmentation (detecting change points)
  - T4:Anomaly detection
- Feature Extraction (e.g. wavelets coefficients)
  - T5:Clustering
  - T6:Indexing TS database
  - T7:Visualization

Goals for Mining Algorithms

- G1:Effective:
  - achieve low reconstruction error (mean square error)
- G2:Scalable:
  - to the size (e.g. length) of sequences
  - on modern hardware (e.g. multi-core)

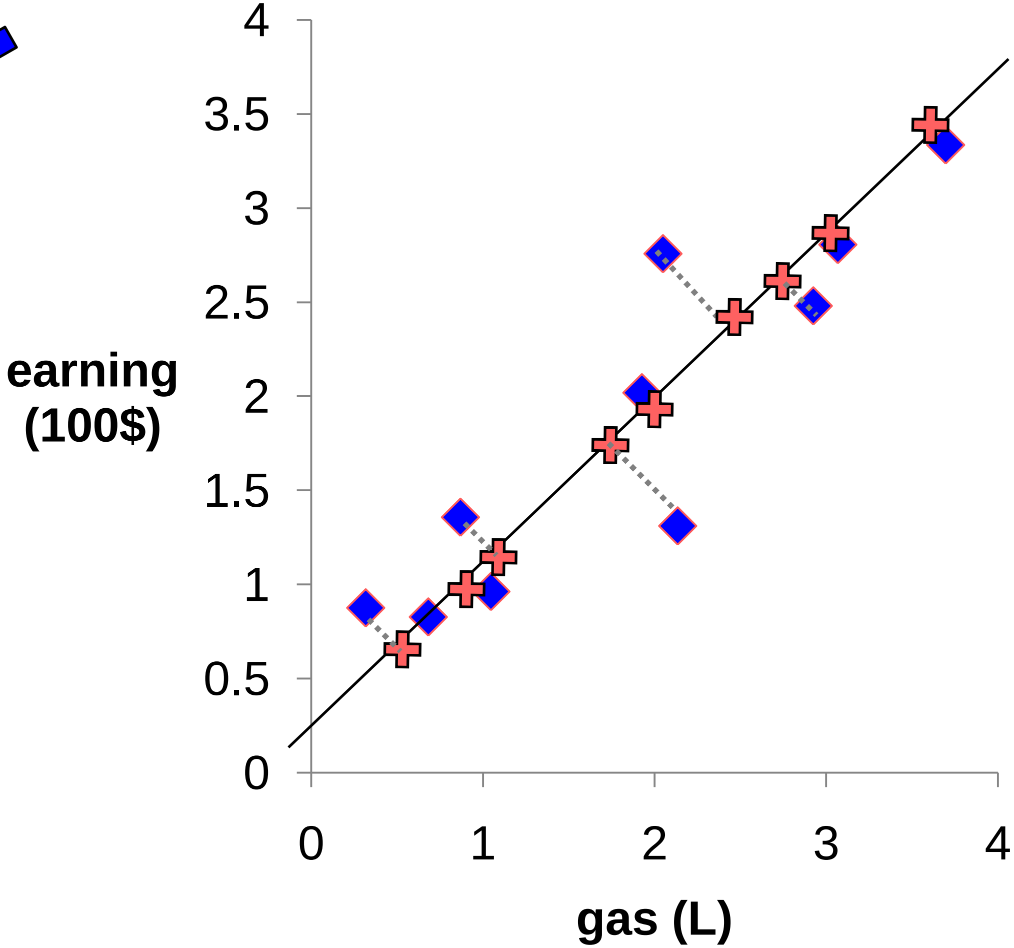
Outline

- Motivation
  - Mining tasks, goals
- Background
  - Principal Component Analysis (PCA)
  - Kalman Filters
- P1:Mining w/ Missing Value [Li+ 2009]
- P2:Parallel Learning [Li+ 2008b]
- Conclusion

Principal Component Analysis (PCA) / Singular Value Decomposition (SVD)

original data

gas	earning
0.32	0.88
0.68	0.83
0.87	1.36
1.05	0.96
2.13	1.31
1.93	2.02
2.05	2.76
2.92	2.48
3.07	2.81
3.70	3.34



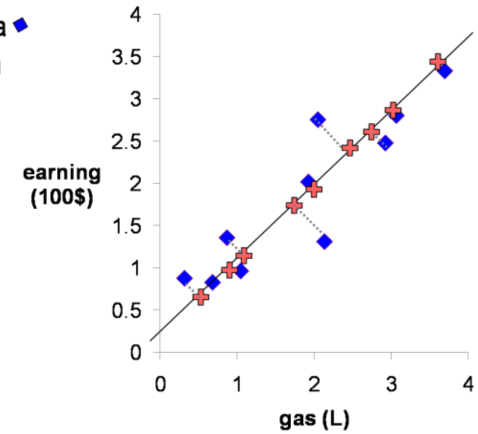
PC1

−1.84
−1.58
−1.10
−1.21
−0.15
0.13
0.69
1.20
1.52
2.34

PCA

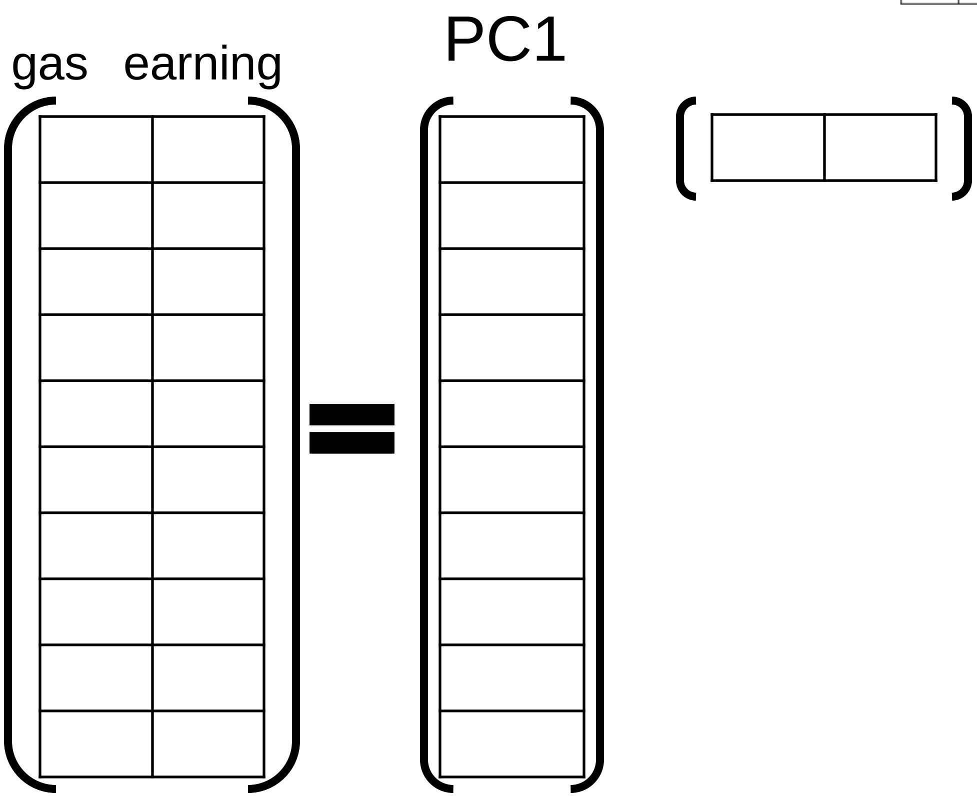
original data

gas	earning
0.32	0.88
0.68	0.83
0.87	1.36
1.05	0.96
2.13	1.31
1.93	2.02
2.05	2.76
2.92	2.48
3.07	2.81
3.70	3.34



PC1

PC1
−1.84
−1.58
−1.10
−1.21
−0.15
0.13
0.69
1.20
1.52
2.34



PCA: general data matrix

data:
column
centered

PC1, 2..k

$$X = U \, V^T$$

Outline

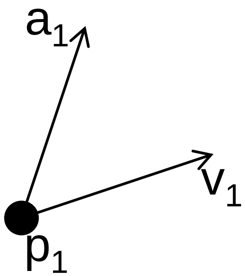
- Motivation
  - Mining tasks, goals
- Background
  - Principal Component Analysis (PCA)
  - Kalman Filters
- P1:Mining w/ Missing Value [Li+ 2009]
- P2:Parallel Learning [Li+ 2008b]
- Conclusion

Kalman Filters

- Example: tracking a car
- Given:
  - current observation of position
  - and current estimate of velocity, acceleration
- Find:
  - estimation of position, velocity, acceleration of next time tick
- Also known as Linear Dynamical Systems(LDS)

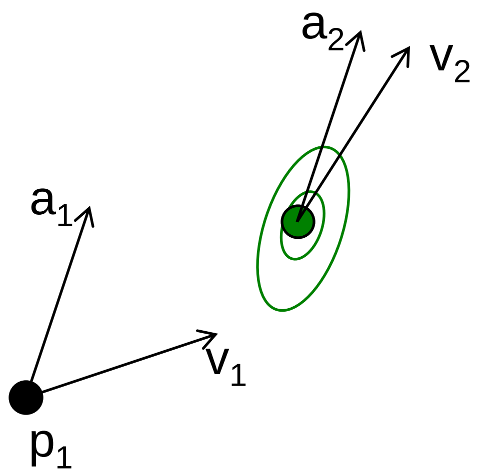
Kalman Filters

Current time tick



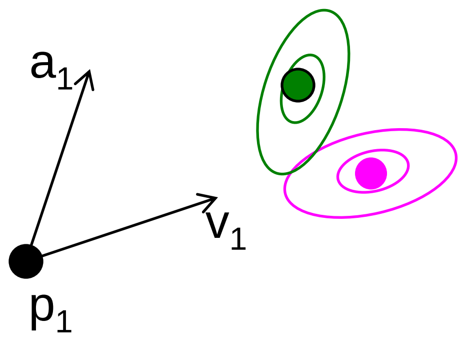
Kalman Filters

Calculate next time tick, w/ transition noise



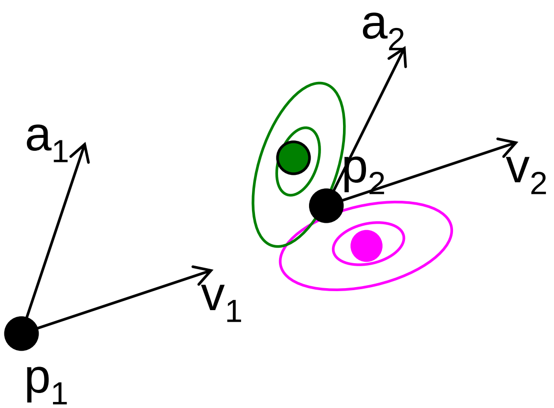
Kalman Filters

Taking a picture, w/ camera noise

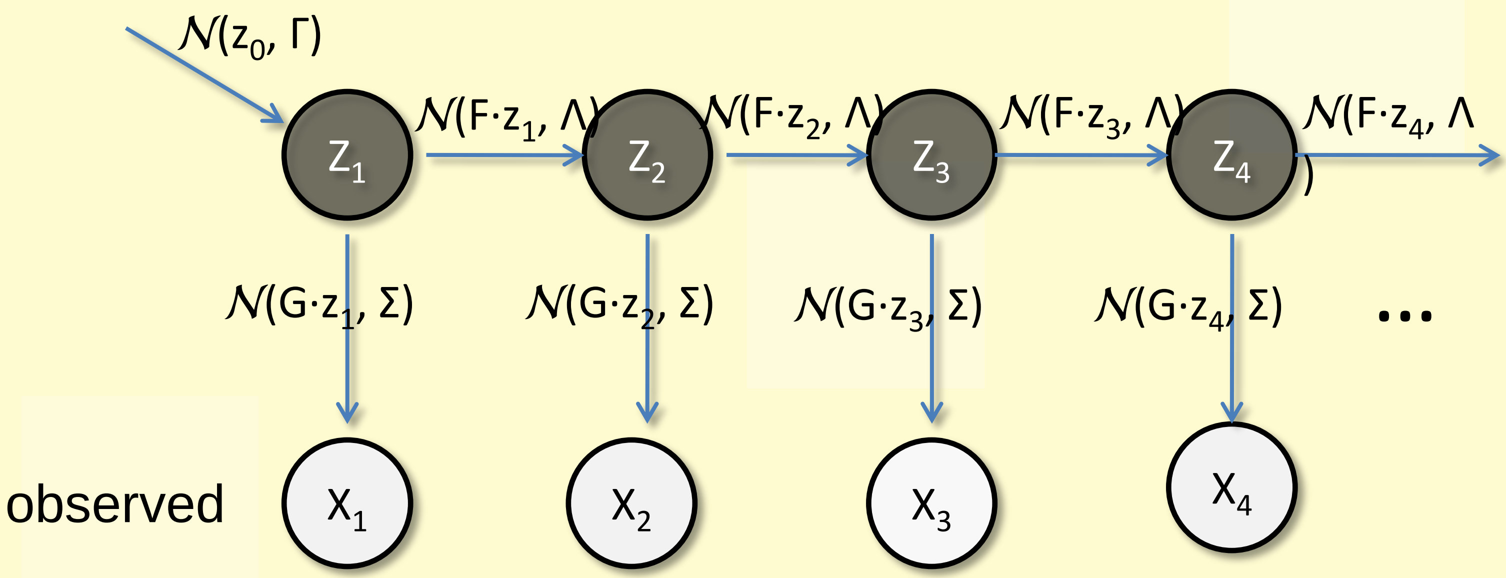


Kalman Filters

Now adjust our estimation of actual position, velocity, acceleration



Graphical Model of LDS (details)

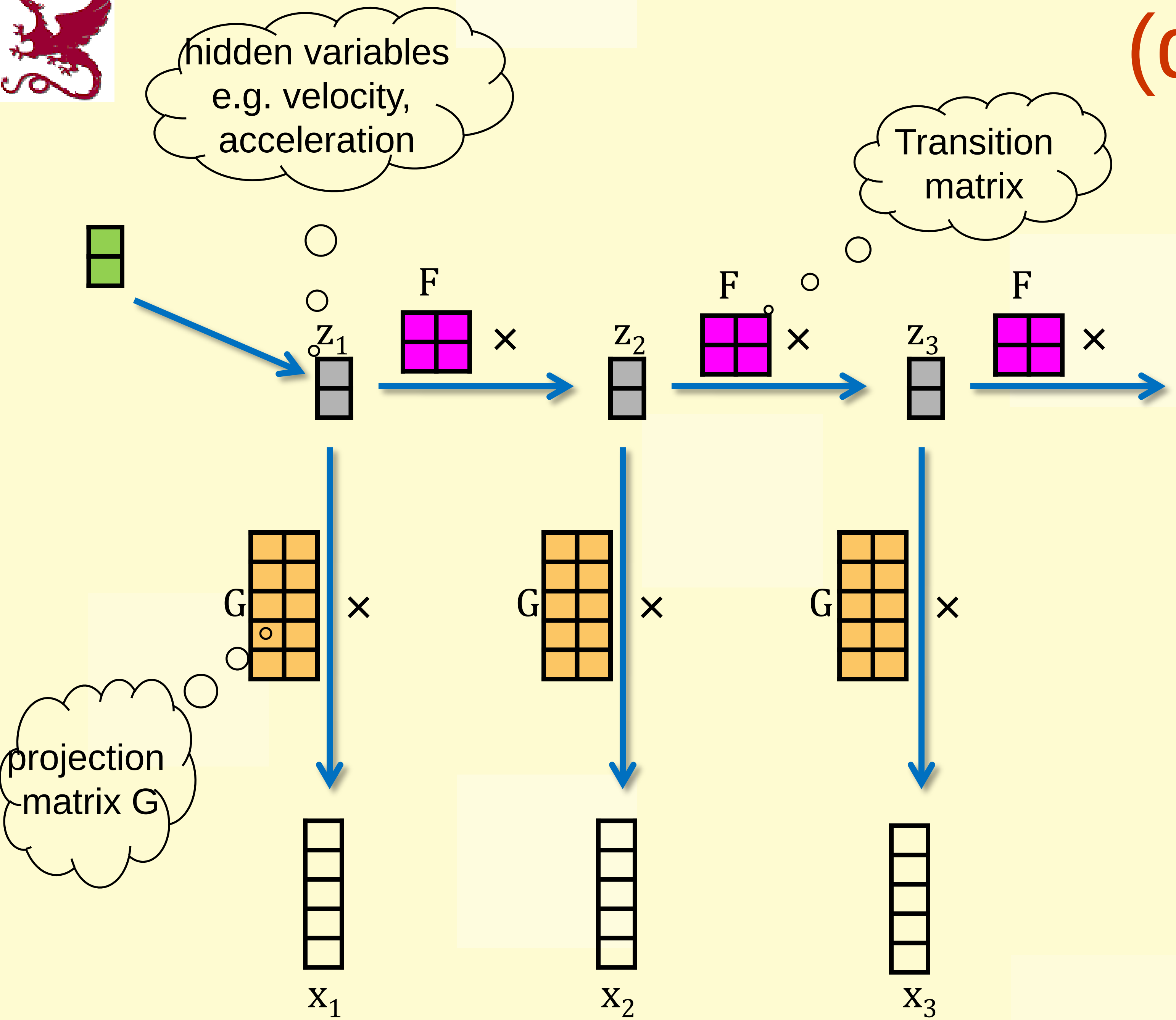


Model parameters:

$\theta = \{z_0, \Gamma, F, \Lambda, G, \Sigma\}$

$$z_1 = z_0 + \omega_0$$
$$z_{n+1} = F \cdot z_n + \omega_n$$
$$x_n = G \cdot z_n + \varepsilon_n$$

(details)

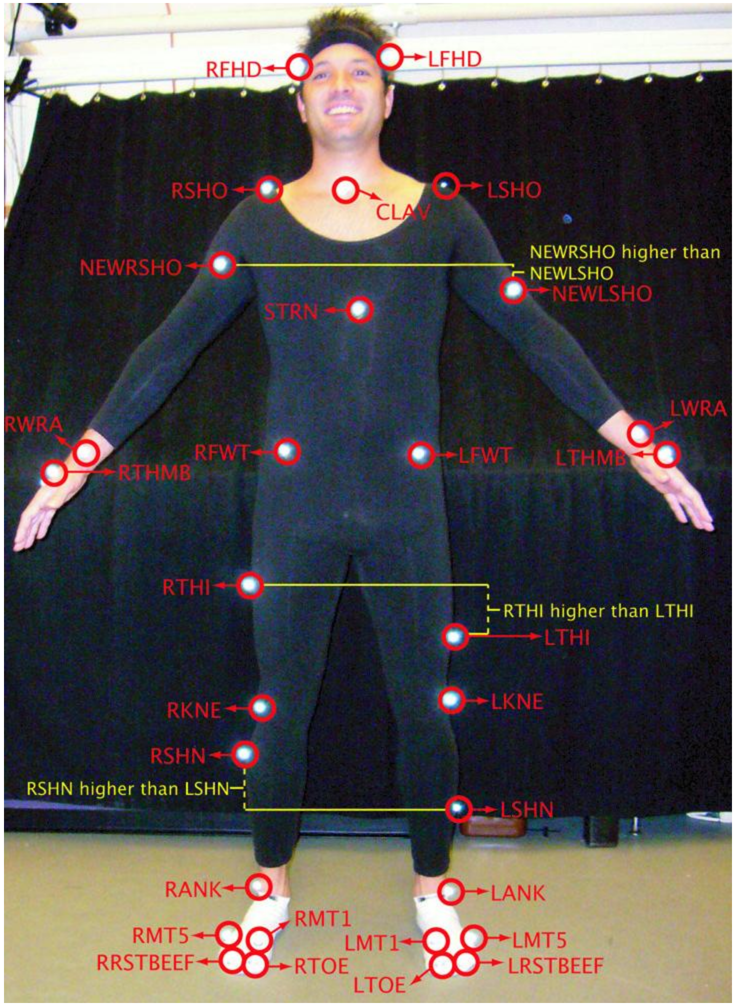

hidden variables e.g. velocity, acceleration
Transition matrix
F
F
F
z_1
z_2
z_3
×
×
×
G
G
G
×
×
×
projection matrix G
x_1
x_2
x_3

Outline

- Motivation
- Background
- P1:Mining w/ Missing Value [Li+ 2009]
  - Problem Definition
  - Proposed Method
    - recovering
    - compression
    - segmentation
  - Results
- P2:Parallel Learning [Li+ 2008b]
- Conclusion

Missing Values in Time Series

- Motion Capture:
  - Markers on human actors
  - Cameras used to track the 3D positions
  - Duration: 100-500
  - 93 dimensional body-local coordinates after preprocessing (31-bones)
- Sensor data missing due to:
  - Low battery
  - RF error

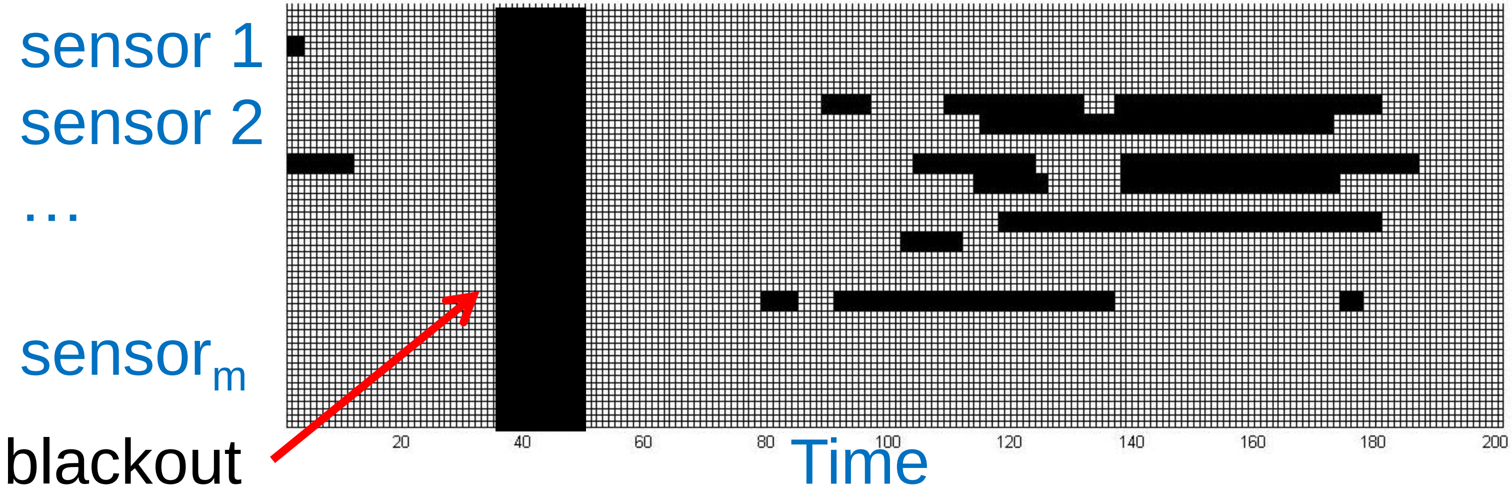


From mocap.cs.cmu.edu

joint work w/ C. Faloutsos, J. McCann, N. Pollard.
[Li et al, KDD 2009]

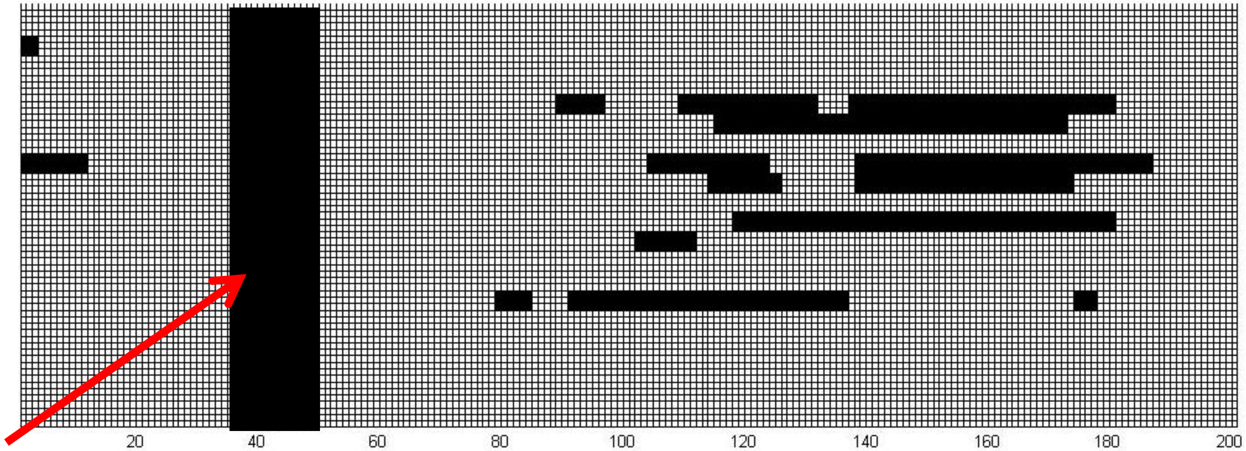
Problem Definition [Li+2009]

- Given



- Find algorithms for:
  – Recovering missing values
  – Compression/summarization (T2)
  – Segmentation (T3)

Problem Definition (cont’)

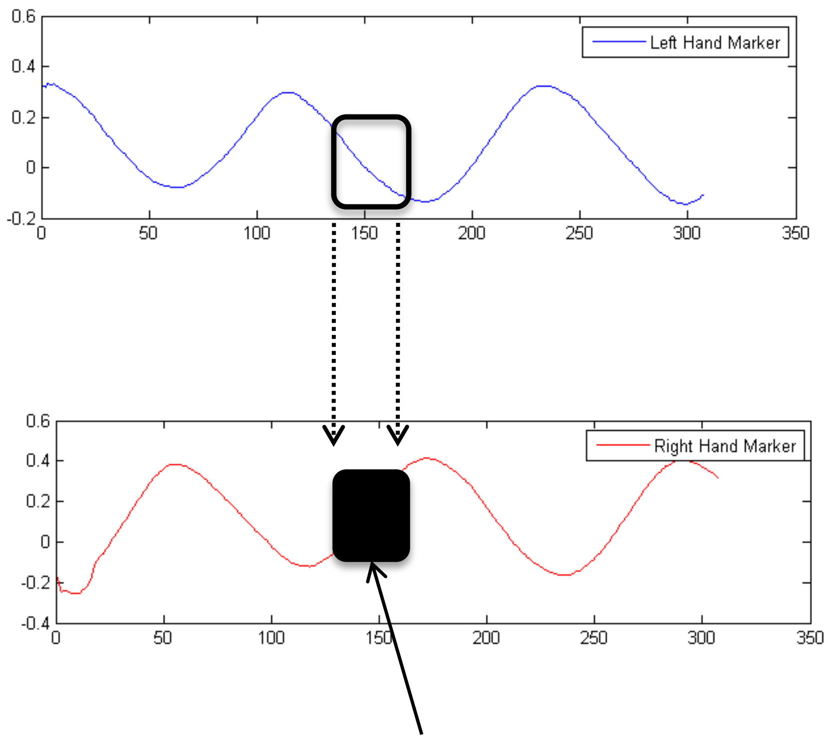


- Want the algorithms to be:
  - G1:*Effective*
  - G2:*Scalable:* to duration of sequences

Proposed Method: Intuition

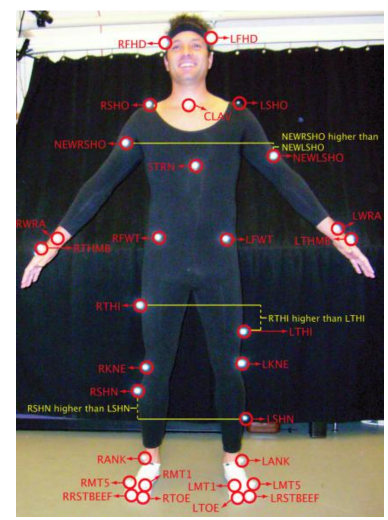
Position of Left hand marker

Position of right hand marker



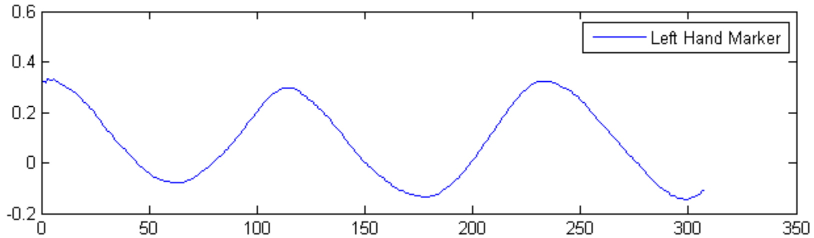
missing

Recover using Correlation among multiple sequences



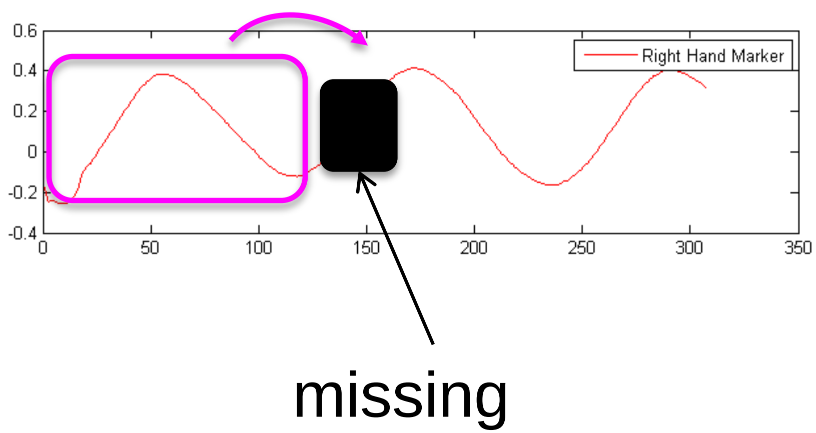
Proposed Method: DynaMMo Intuition

Position of Left hand marker



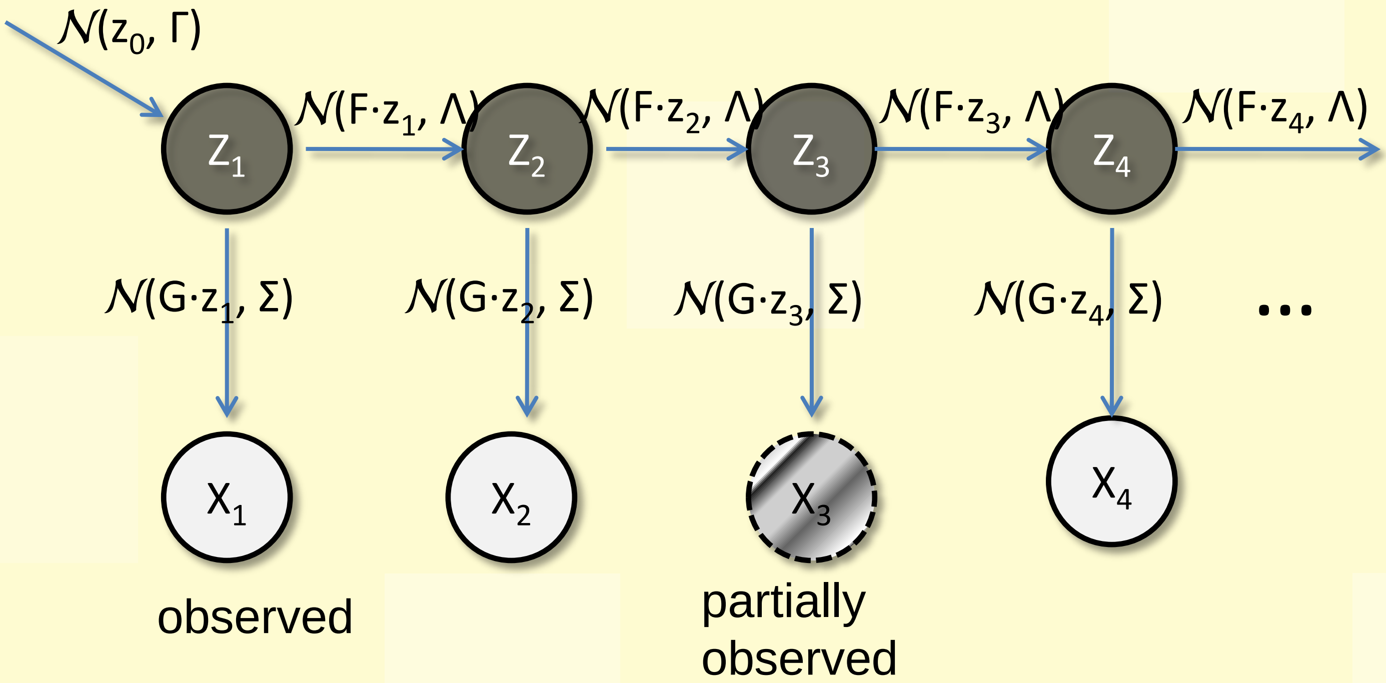
Recover using Dynamics temporal moving pattern

Position of right hand marker



Underlying Model (details)

Use *Linear Dynamical Systems* to model whole sequence.

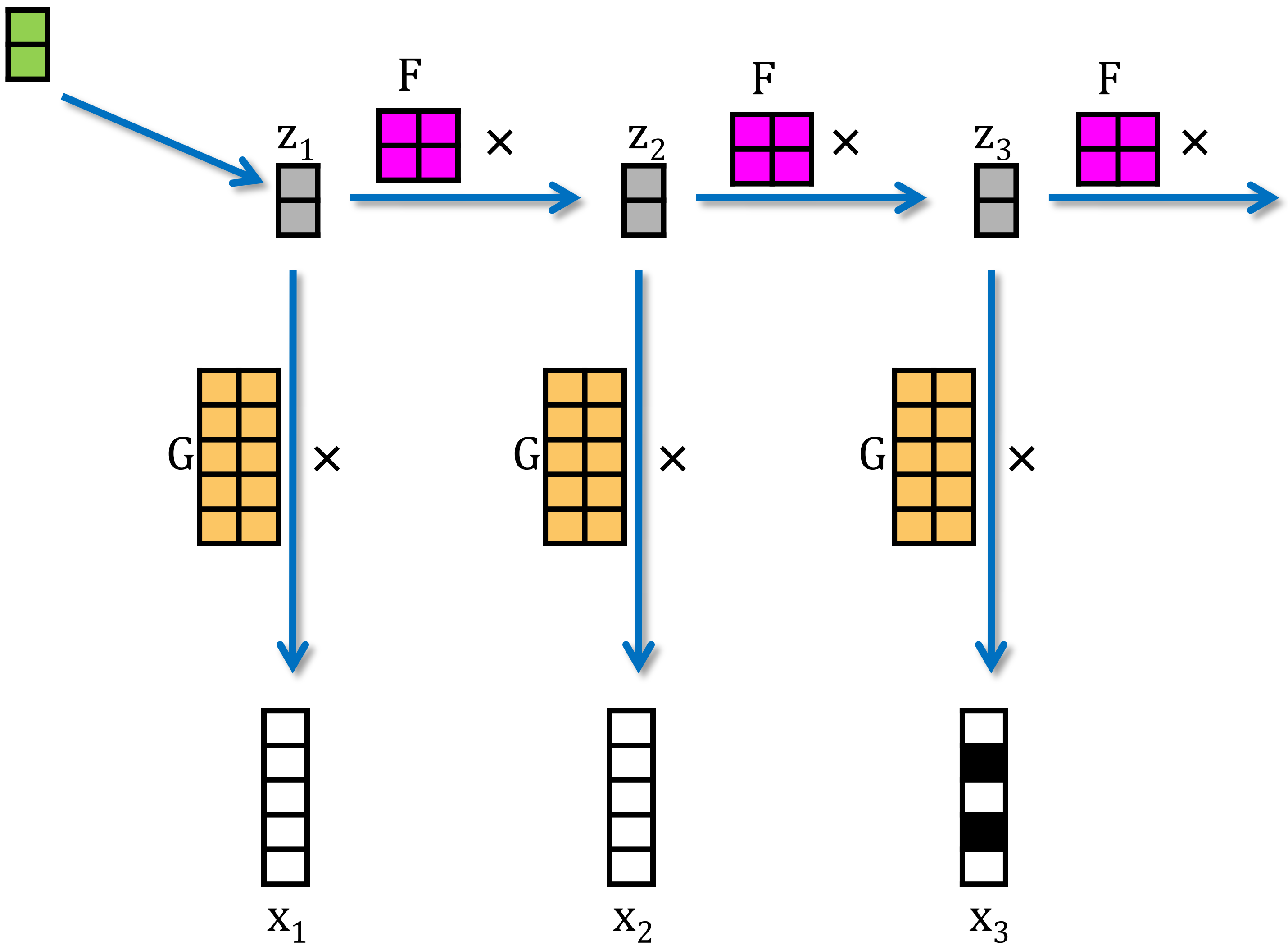


Model parameters:

$$\theta=\{z_0, \Gamma, F, \Lambda, G, \Sigma\}$$

$$z_1 = z_0+\omega_0$$
$$z_{n+1} = F\cdot z_n+\omega_n$$
$$x_n = G\cdot z_n+\varepsilon_n$$

DynaMMo learning: estimate all colored elements



Details in [Li+2009]

(details)

DynaMMo learning

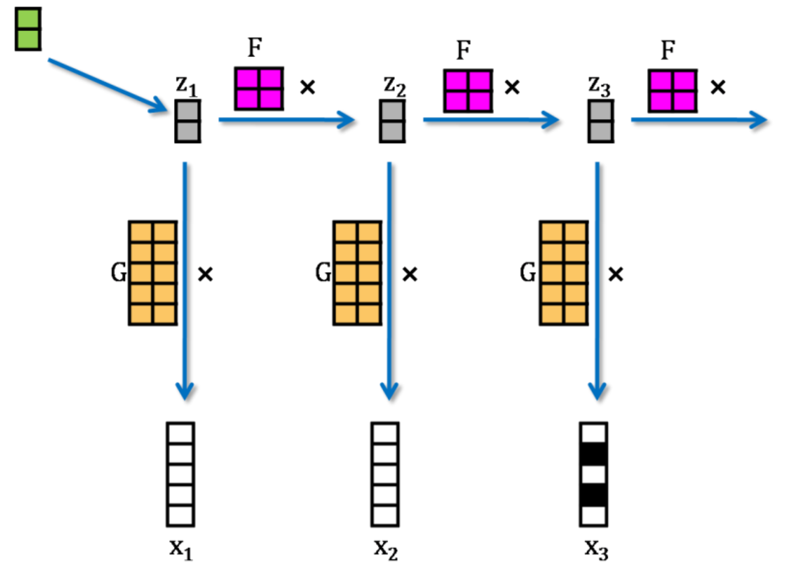
- Finding the best model parameters (θ) and missing values for X to minimize the expected loglikelihood:

$$Q(\theta) = E_{X_m, Z|X_g;\theta}\Big[-(z_1 - z_0)^T \Gamma^{-1}(z_1 - z_0)$$
$$- \sum_{n=2}^{N} (z_n - F \cdot z_{n-1})^T \Lambda^{-1} (z_n - F \cdot z_{n-1})$$
$$- \sum_{n=1}^{N} (x_n - G \cdot z_n)^T \Sigma^{-1} (x_n - G \cdot z_n)\Big]$$

- Proposed optimization method:
  - **Expectation-Recover-Maximization**

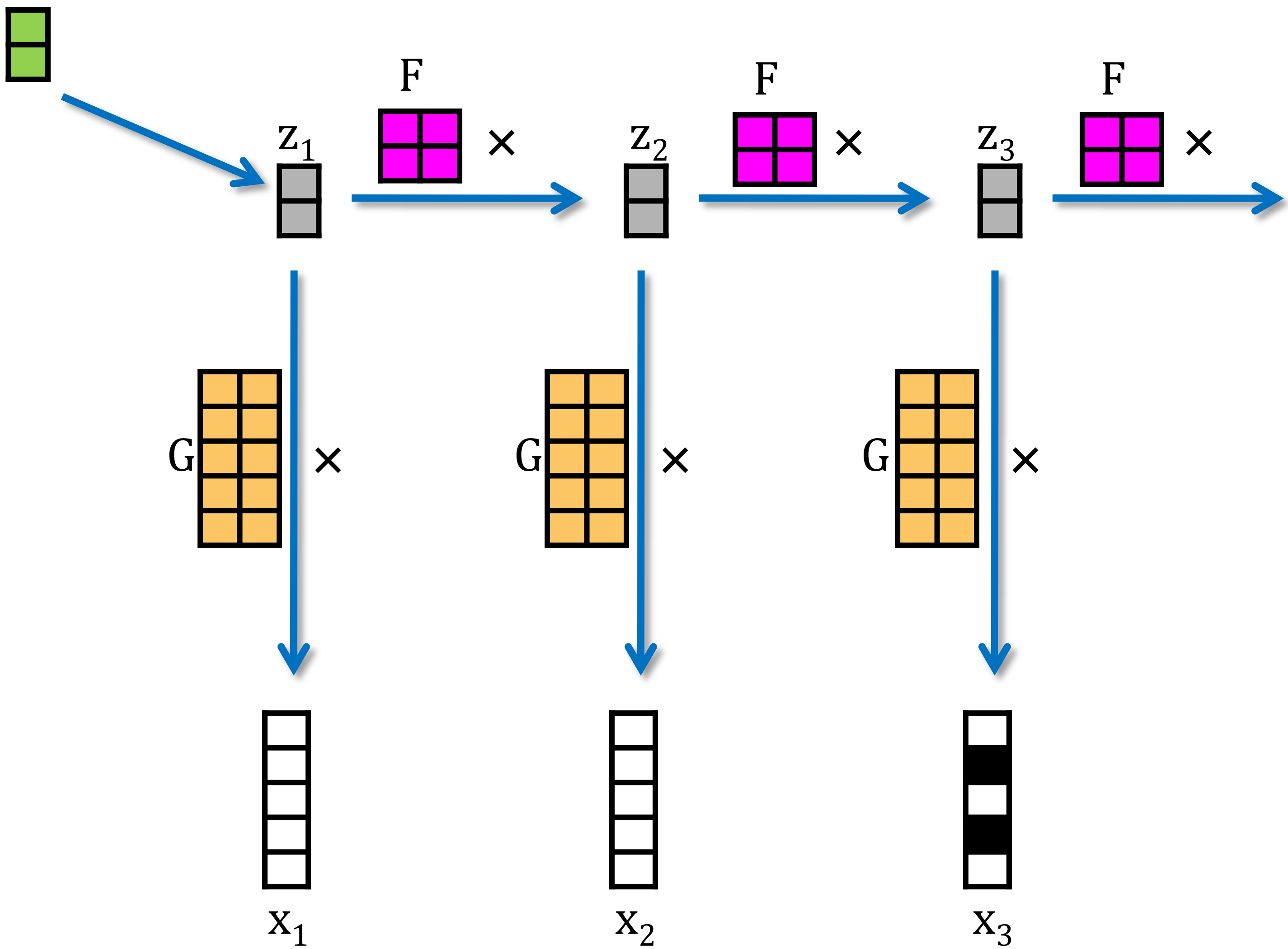
DynaMMo learning: estimate all colored elements

- Step1: Expectation
  – forward-backward estimate hidden variables
- Step 2: Recover
  – missing values
- Step 3: Maximization
  – update model parameters (transition matrix, projection matrix, …)



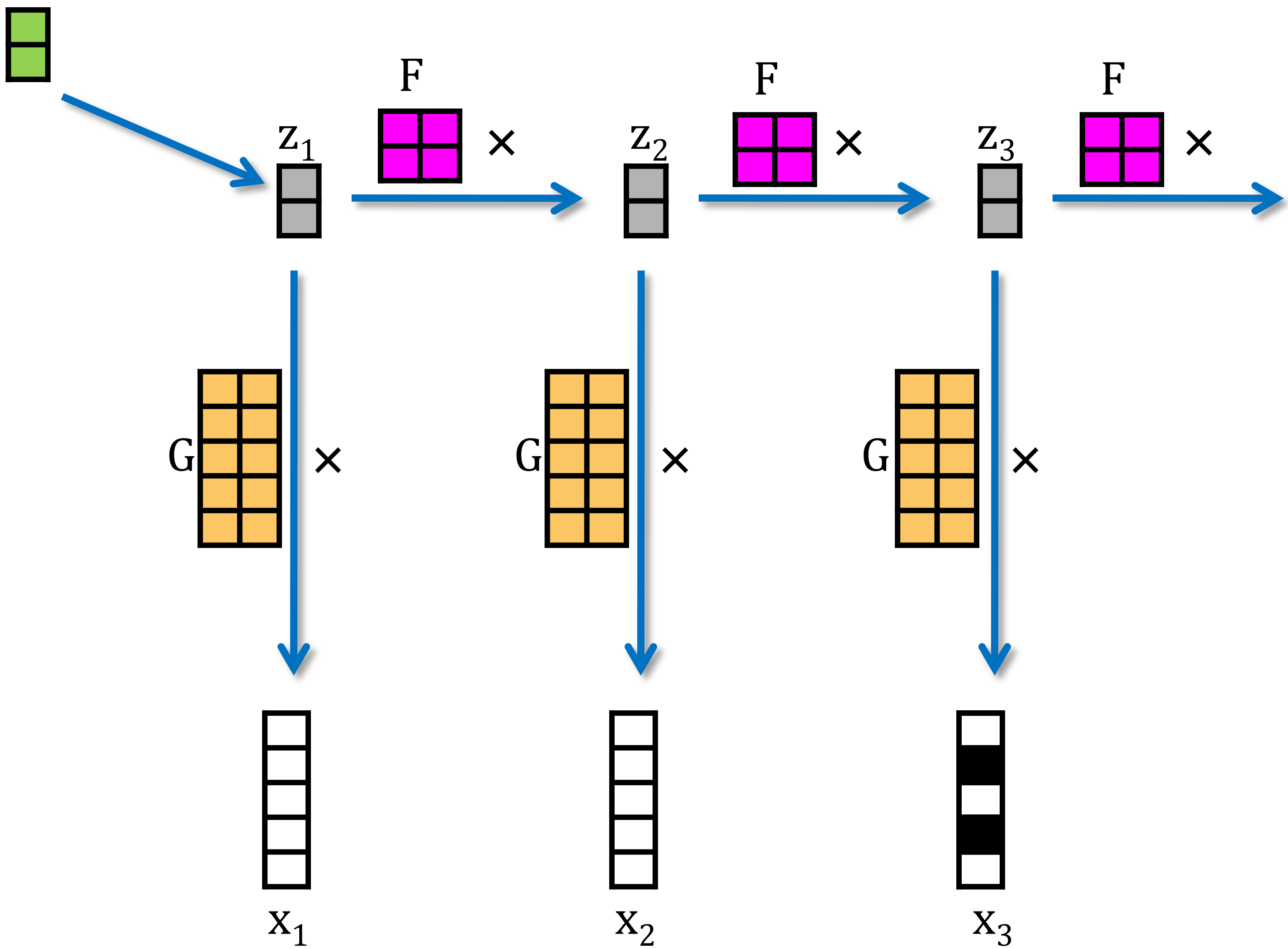
Details next

DynaMMo Illustration: step 1 estimate hidden variables



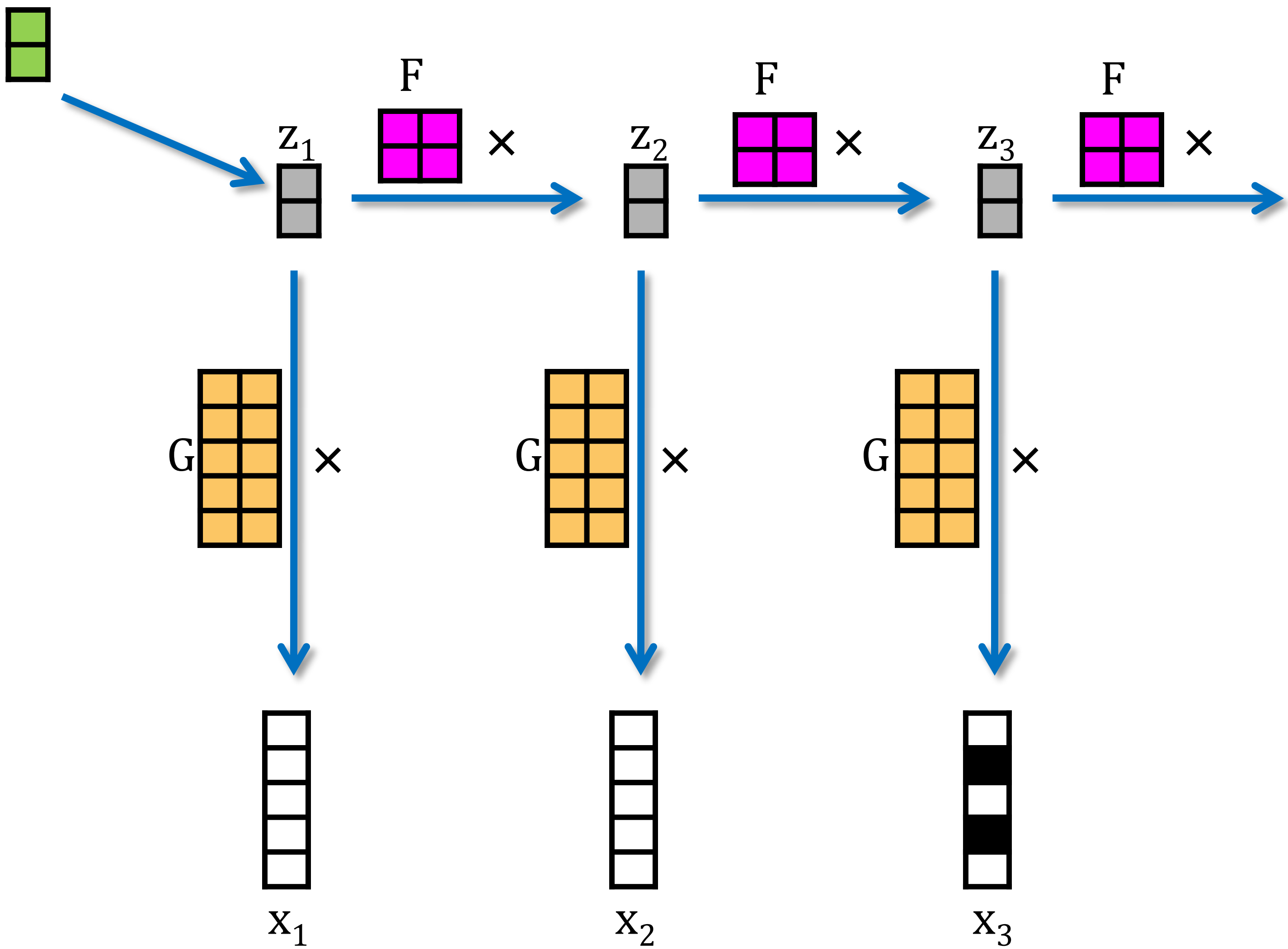
Details in [Li+2009]

DynaMMo Illustration: step 2 recover missing values



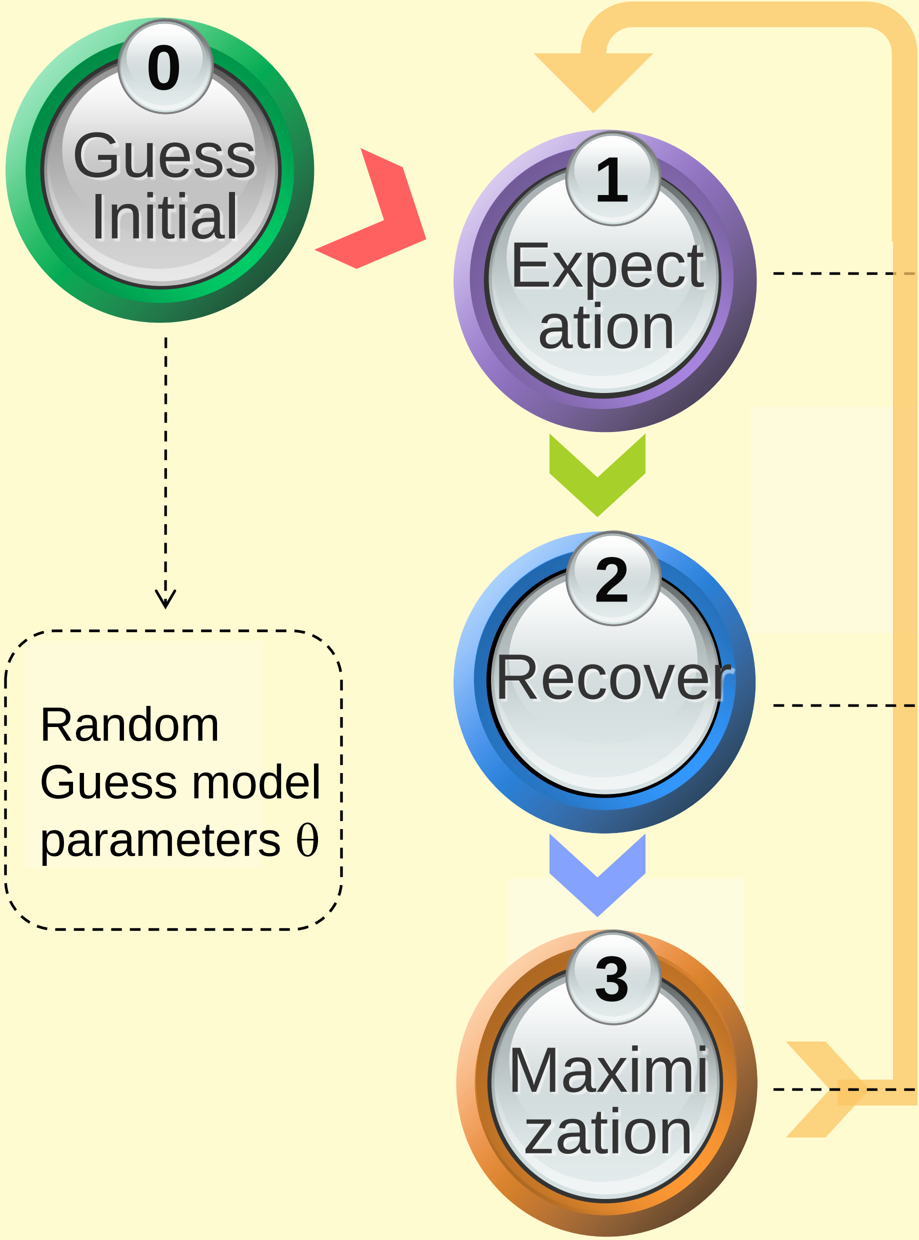
Details in [Li+2009]

DynaMMo Illustration: step 3 update model parameters



Details in [Li+2009]

DynaMMo Learning (details)



Random Guess model parameters θ

Fix X,
Estimate $P(Z|X;\theta)$:
$E(z_n|X;\theta)$,
$E(z_n z'_n|X;\theta)$
$E(z_n z'_{n+1}|X;\theta)$

fix Z,
estimate
$E(X\_missing|Z;\theta)$
Using $E(z_n|X;\theta)$,

Fix both X and Z,
estimate new model parameters θ
argmax $E[\log(X,Z;\theta)]$

Outline

- Motivation
- Background
- P1:Mining w/ Missing Value [Li+ 2009]
  - Problem Definition
  - Proposed Method
    - recovering
    - compression
    - segmentation
  - Results
- P2:Parallel Learning [Li+ 2008b]
- Conclusion

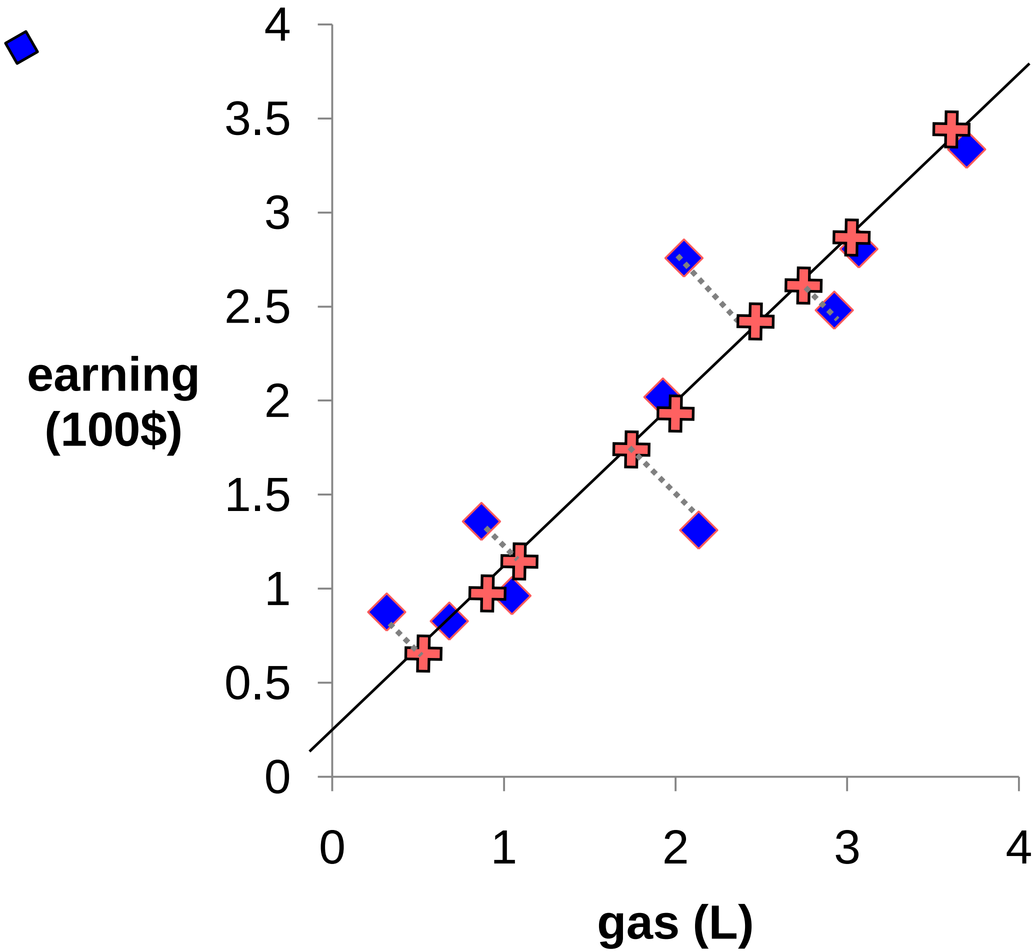
How to Compress

- Competitor #1: use PCA/SVD
- Competitor #2: store parameters of LDS
- Proposed Methods: DynaMMo compression and variants
  – Carefully choosing what to store

Recap: PCA/SVD

original data

gas	earning
0.32	0.88
0.68	0.83
0.87	1.36
1.05	0.96
2.13	1.31
1.93	2.02
2.05	2.76
2.92	2.48
3.07	2.81
3.70	3.34

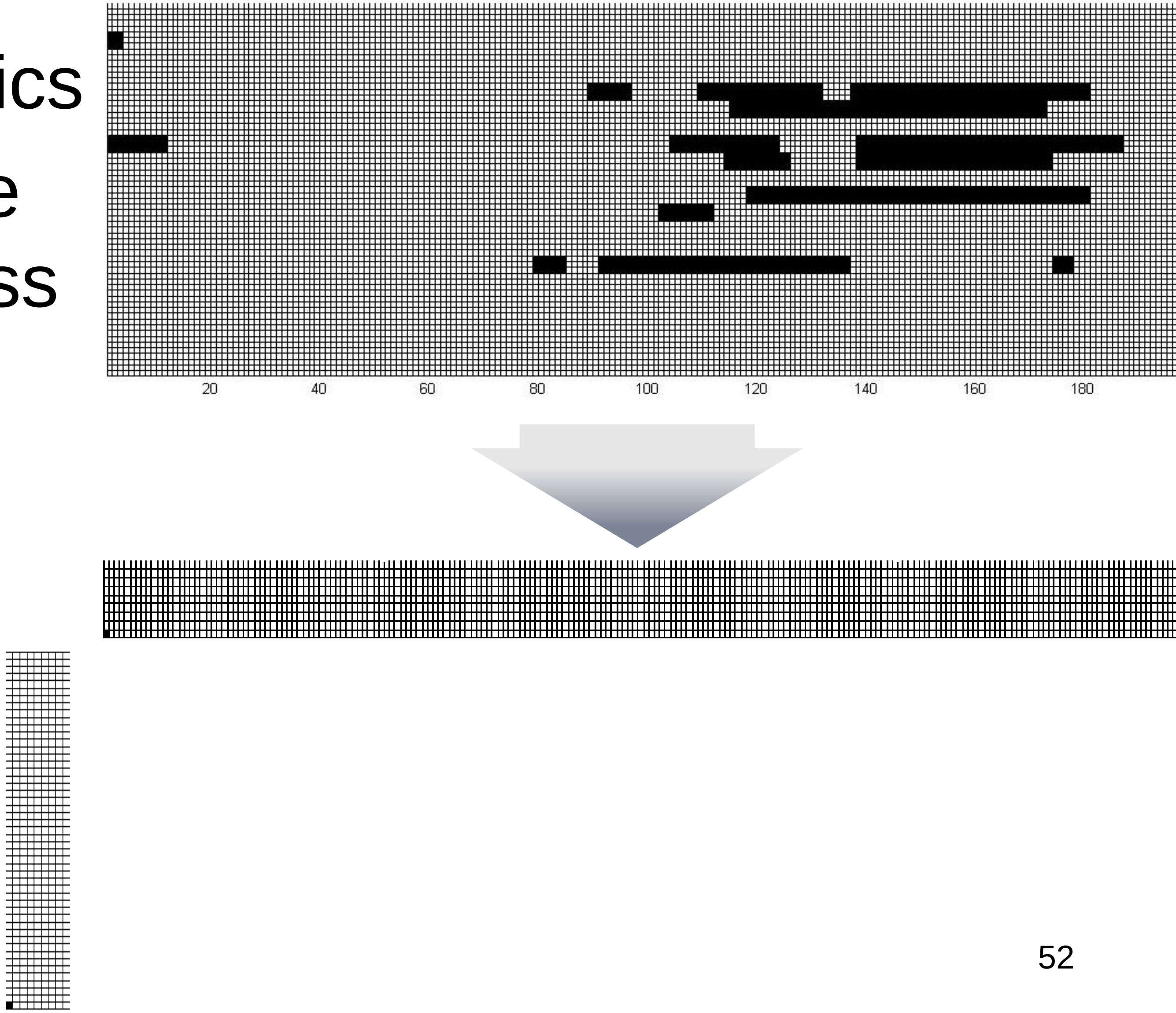


PC1

−1.84
−1.58
−1.10
−1.21
−0.15
0.13
0.69
1.20
1.52
2.34

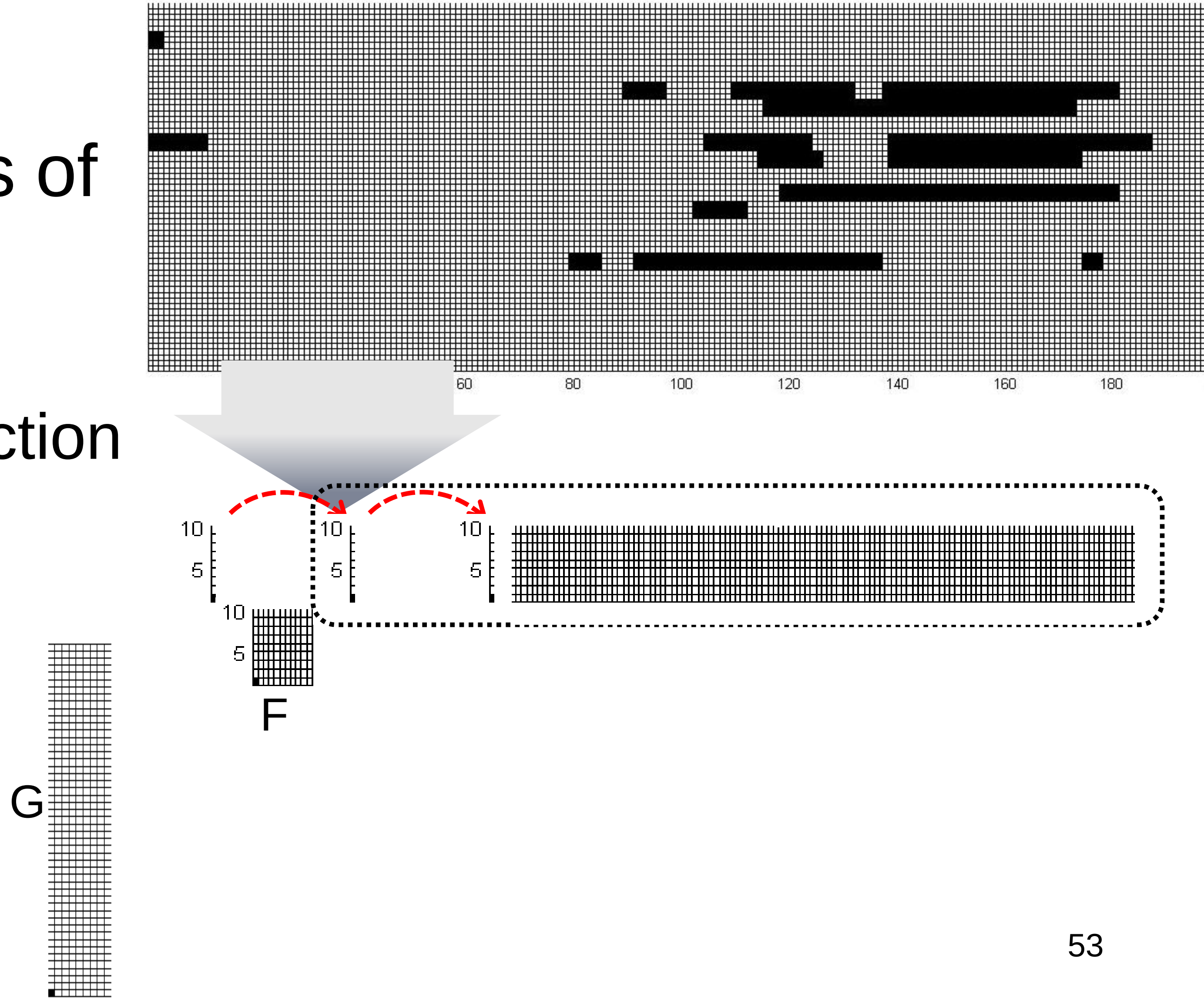
Why Not PCA/SVD? (competitor #1)

- No dynamics
- Need more to compress w/ same accuracy



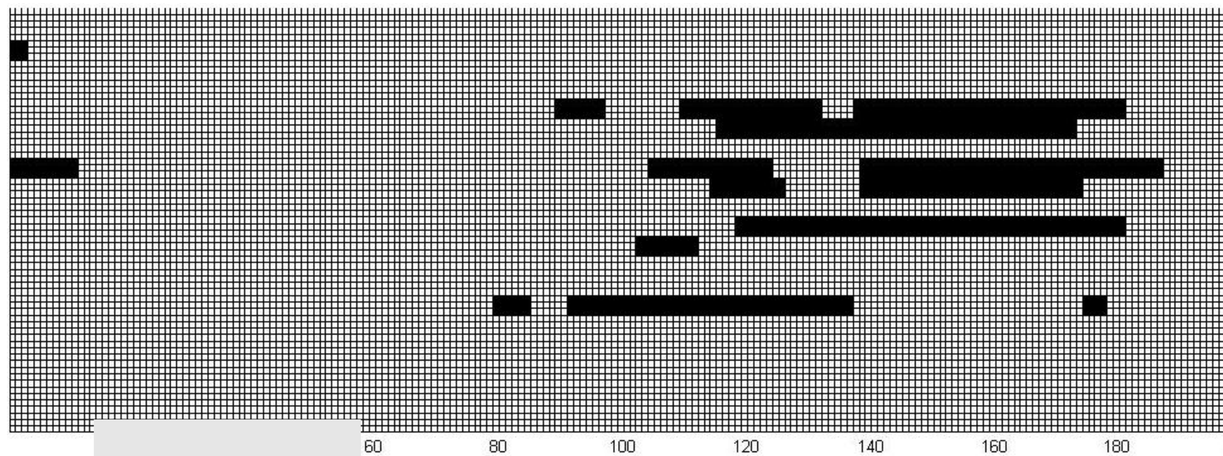
Why Not LDS? (competitor #2)

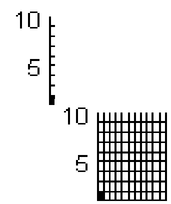
- Store parameters of LDS
  - bad reconstruction



Why Not LDS?

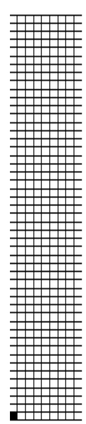
- Idea #2: store parameters of LDS
  - bad reconstruction





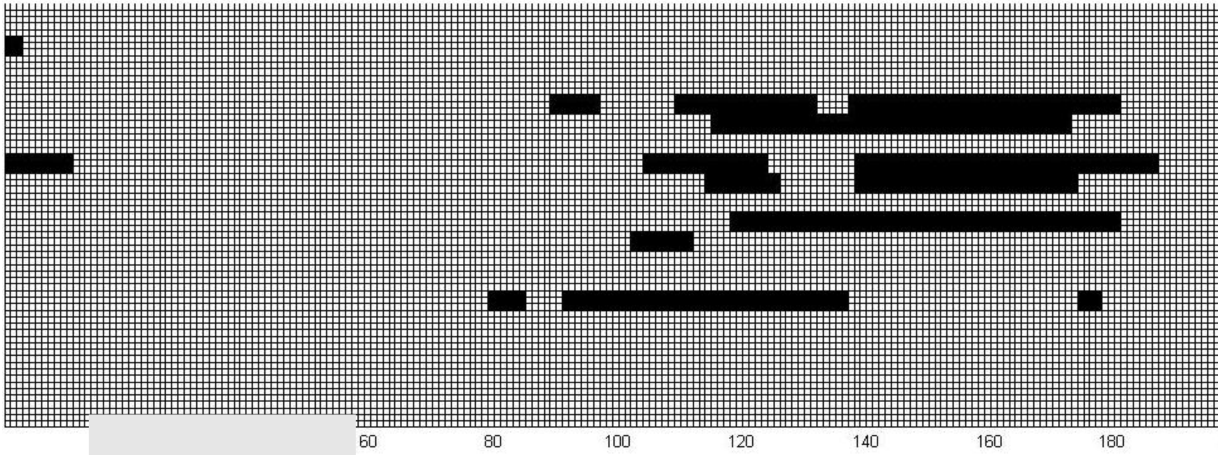
transition F

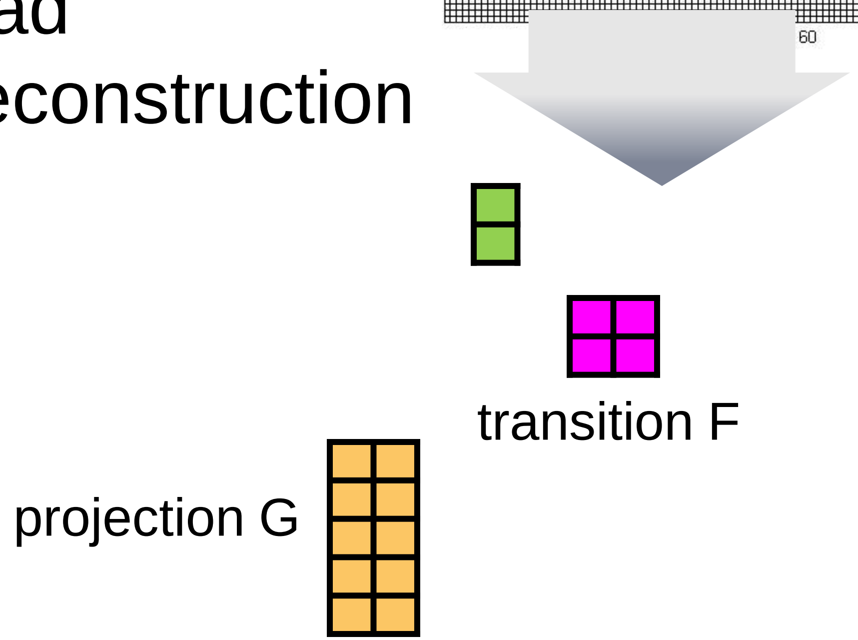
projection G



Why Not LDS? (competitor #2)

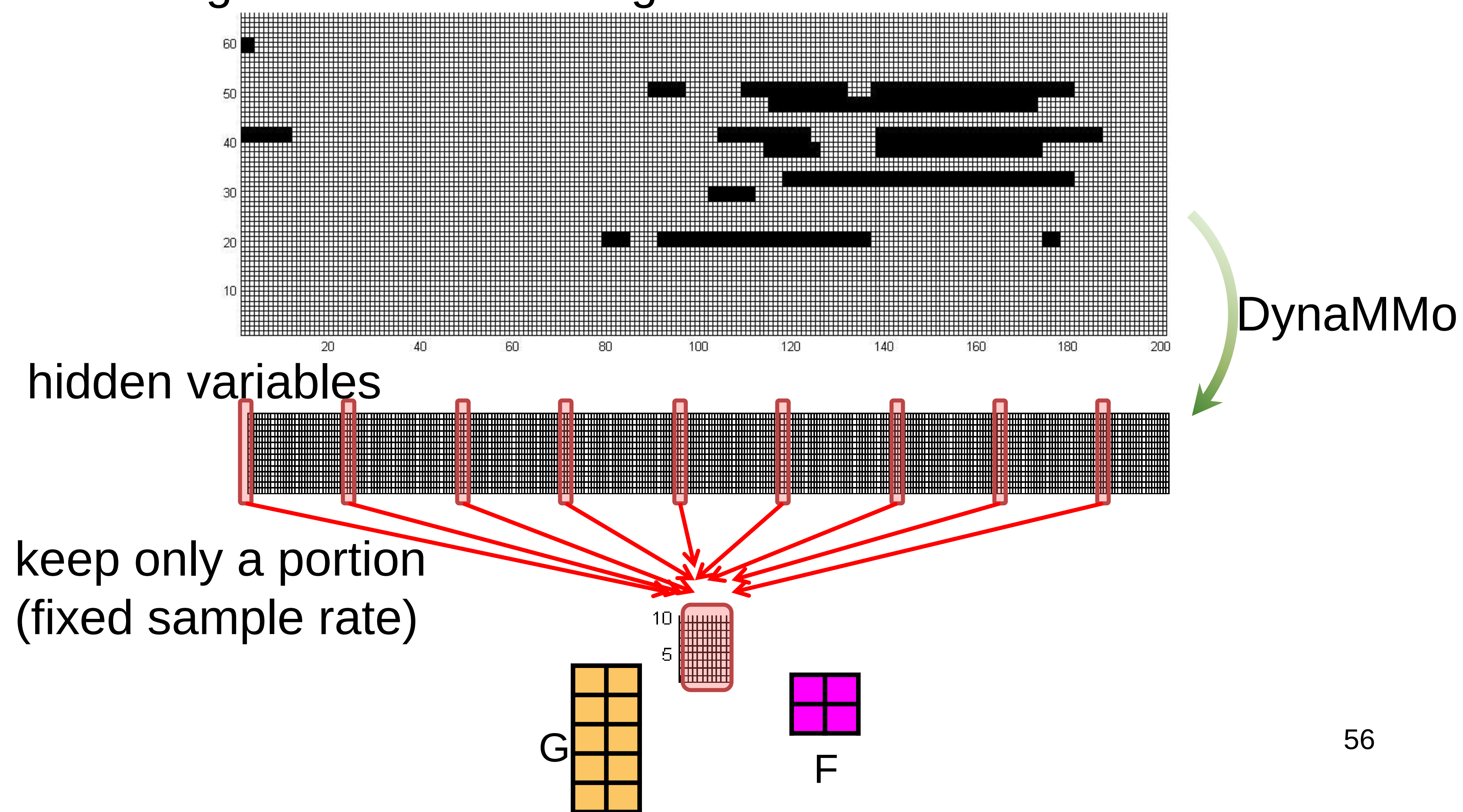
- Store parameters of LDS
  - bad reconstruction



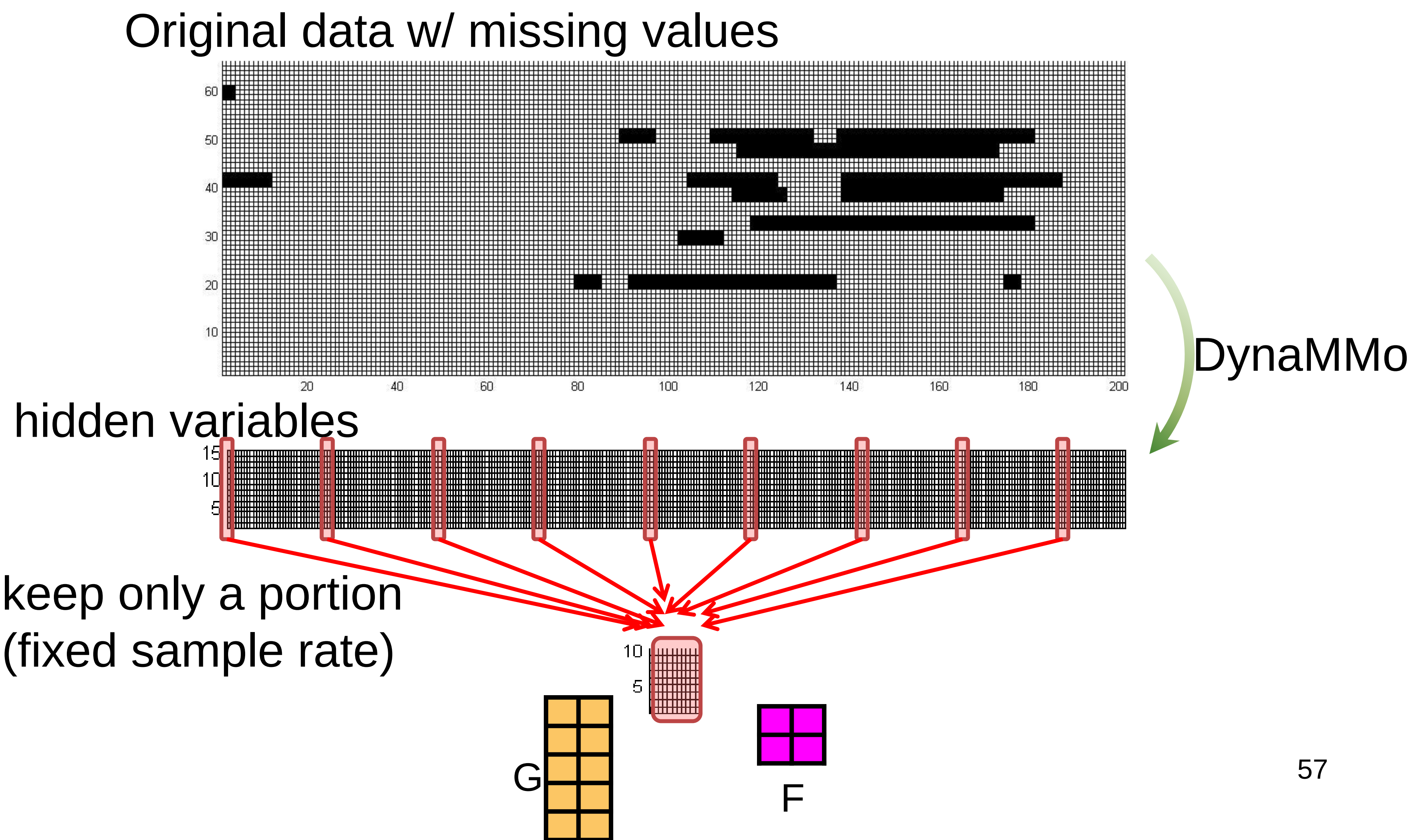


DynaMMo Compression: Intuition, like LDS but sync

Original data w/ missing values

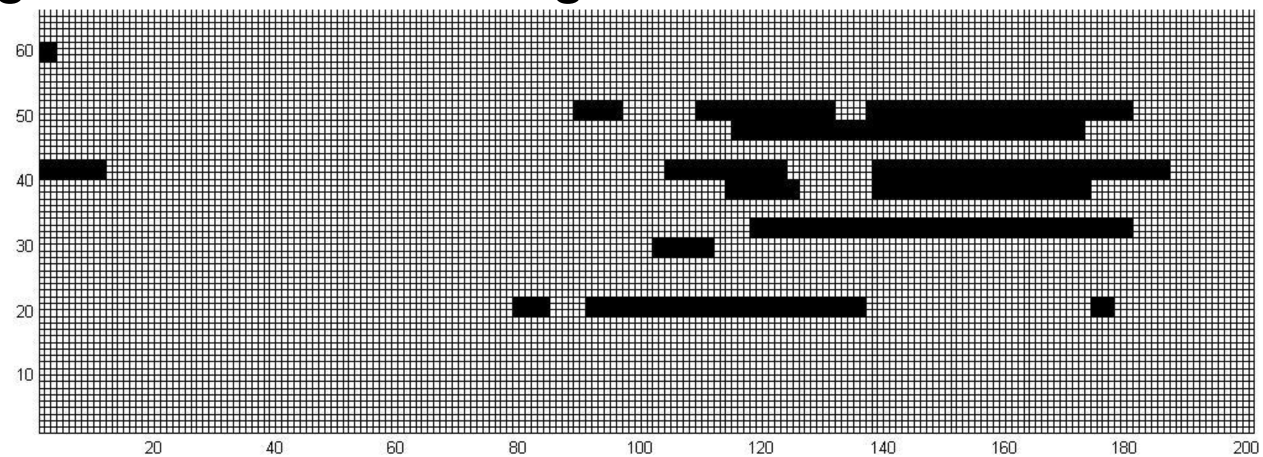


Q: Can we do better?



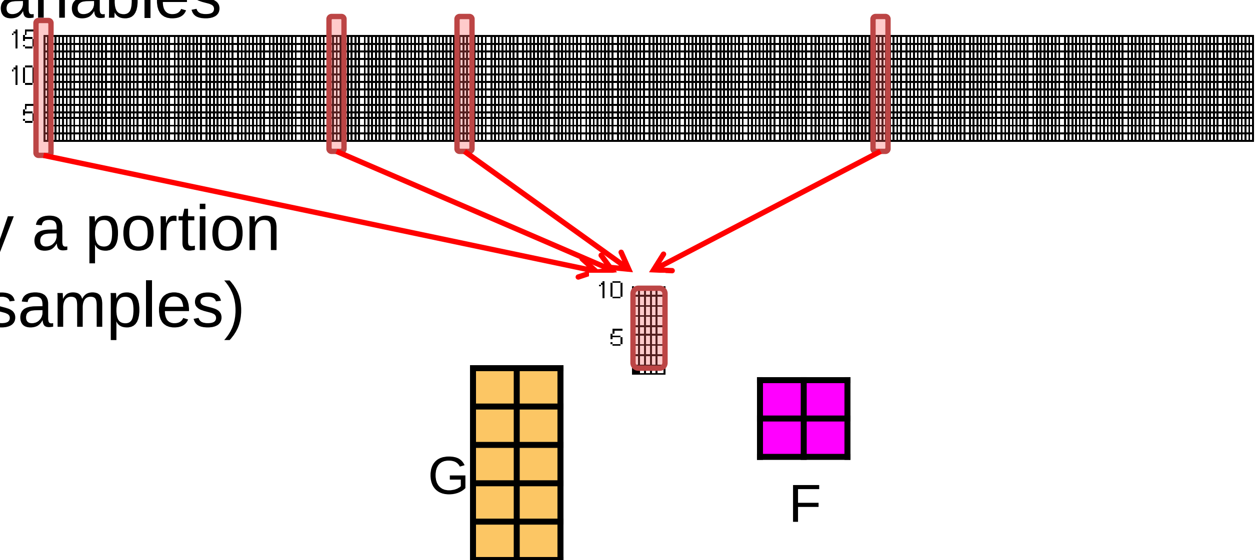
A: Yes, samples adaptively DynaMMo$_d$ Compression

Original data w/ missing values



DynaMMo

hidden variables



keep only a portion
(optimal samples)

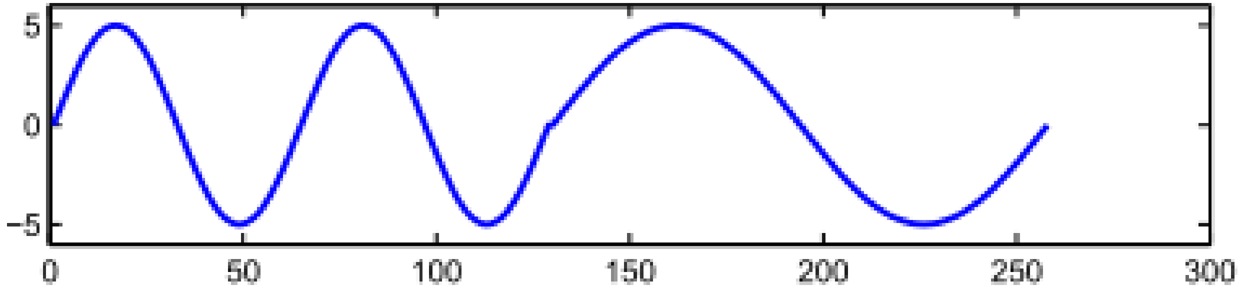
Outline

- Motivation
- Background
- P1:Mining w/ Missing Value [Li+ 2009]
  - Problem Definition
  - Proposed Method
    - recovering
    - compression
    - segmentation
  - Results
- P2:Parallel Learning [Li+ 2008b]
- Conclusion

How to Segment

- Segment by threshold on reconstruction error

original data

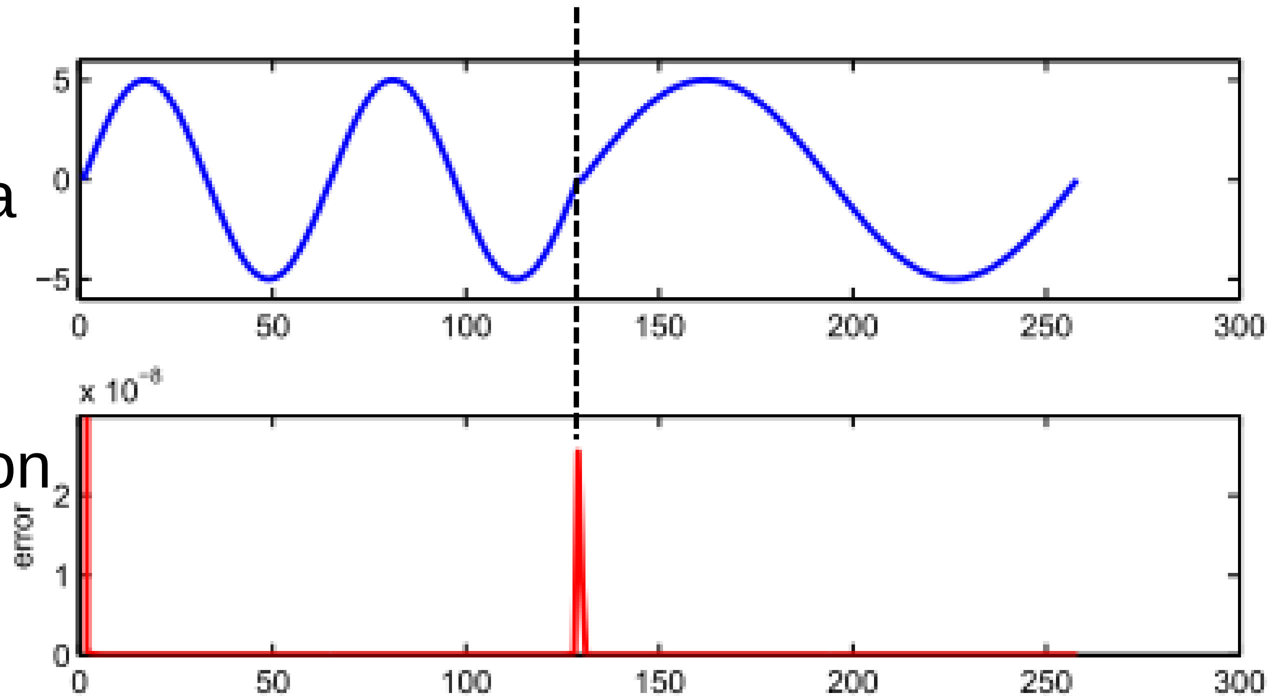


How to Segment

- Segment by threshold on reconstruction error

original data

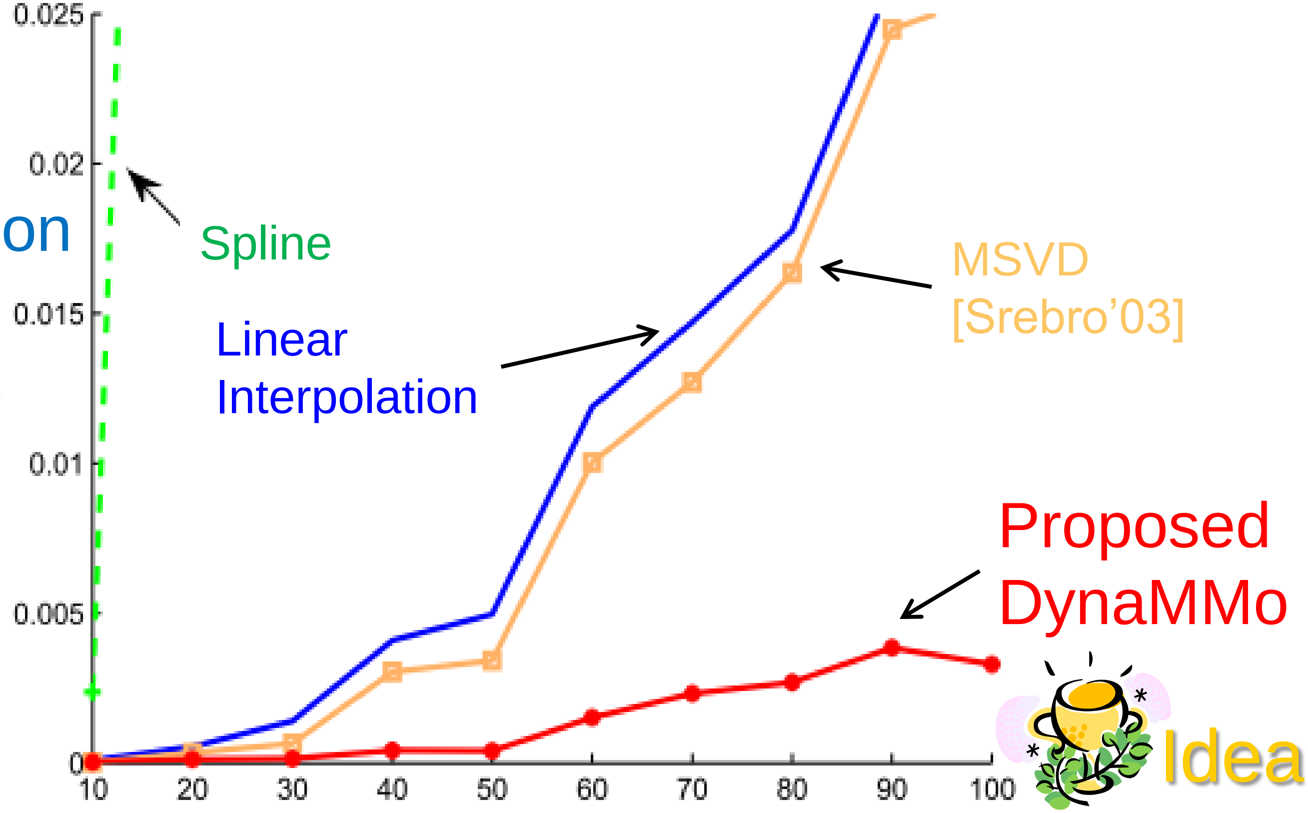
reconstruction error



Outline

- Motivation
- Background
- P1:Mining w/ Missing Value [Li+ 2009]
  - Problem Definition
  - Proposed Method
  - Results
    - Recovery
    - Compression
    - Segmentation

Results – Better Missing Value Recovery



Reconstruction error

Average missing length

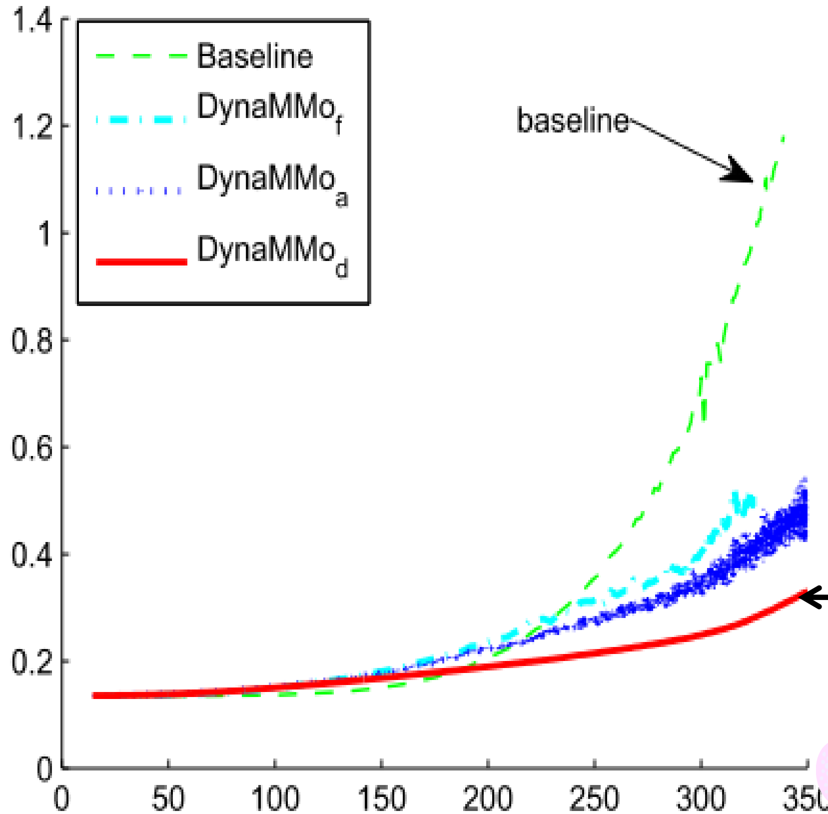
Ideal

Dataset:
CMU Mocap #16
mocap.cs.cmu.edu

more results in [Li+2009]

Results – Better Compression

error



Compression ratio

DynaMMo
w/ optimal
compression

Ideal

Dataset:
Chlorine levels

more results in [Li+2009]

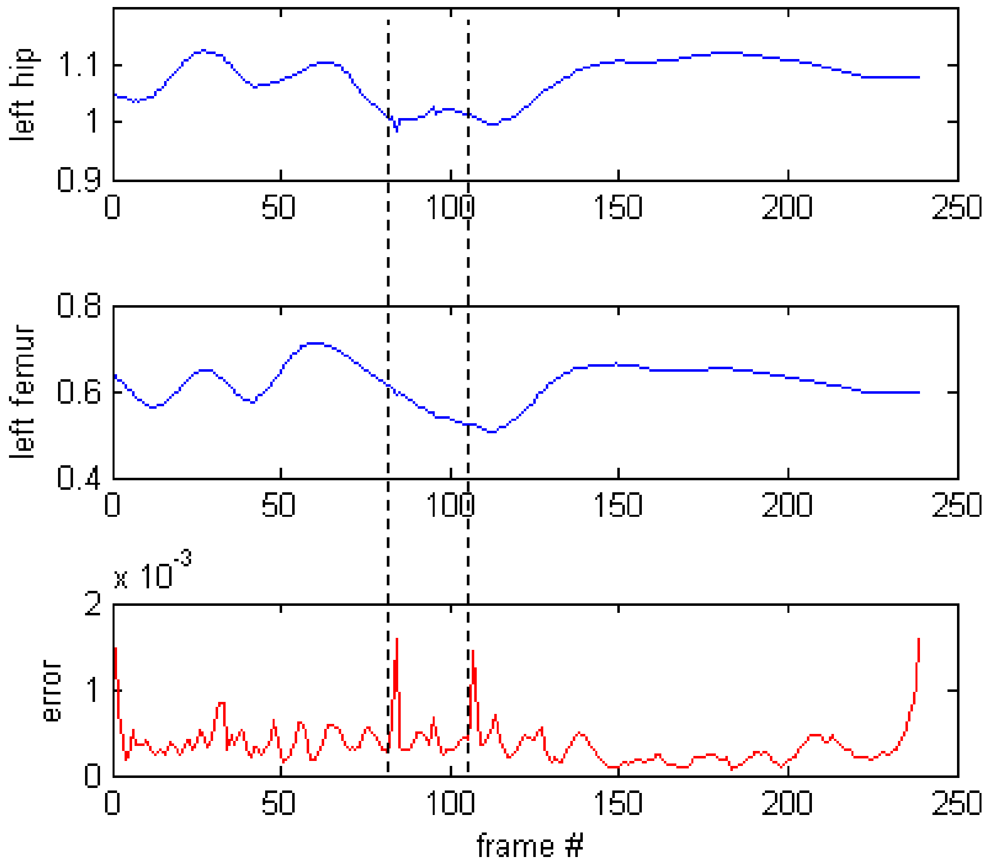
Results – Segmentation

- Find the *transition* during “running” to “stop”.

left hip

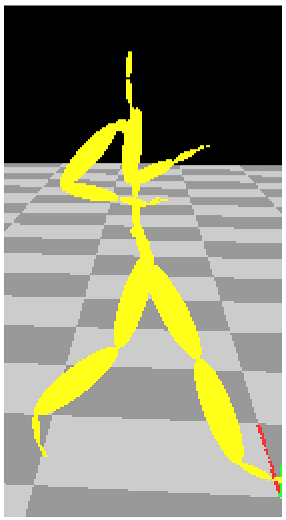
left femur

reconstruction error



Results – Segmentation

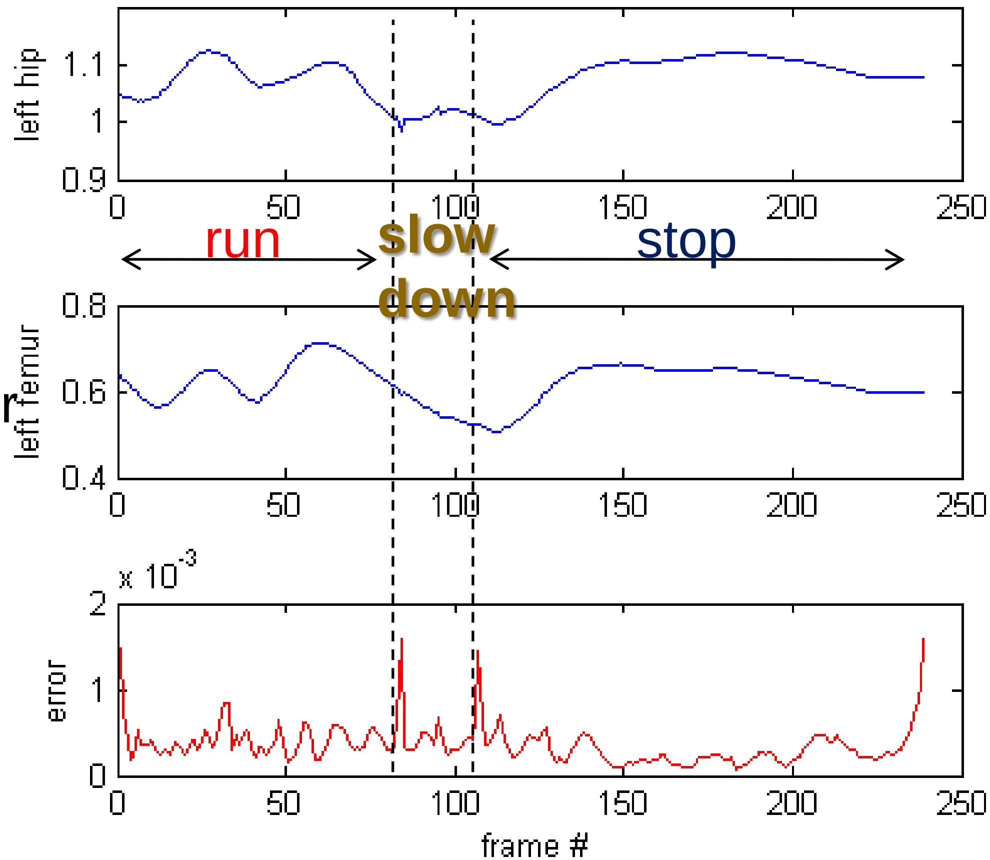
- Find the *transition* during “running” to “stop”.

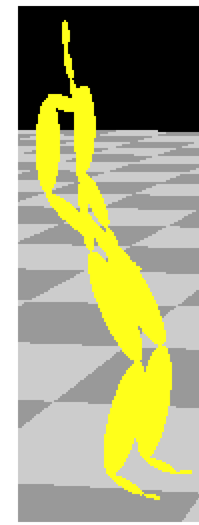


left hip

left femur

reconstruction error





Outline

- Motivation
- Background
- P1:Mining w/ Missing Value [Li+ 2009]
  – *Contribution*: the most accurate mining algorithms for TS with missing values so far.
- P2:Parallel Learning [Li+ 2008b]
- Conclusion

Outline

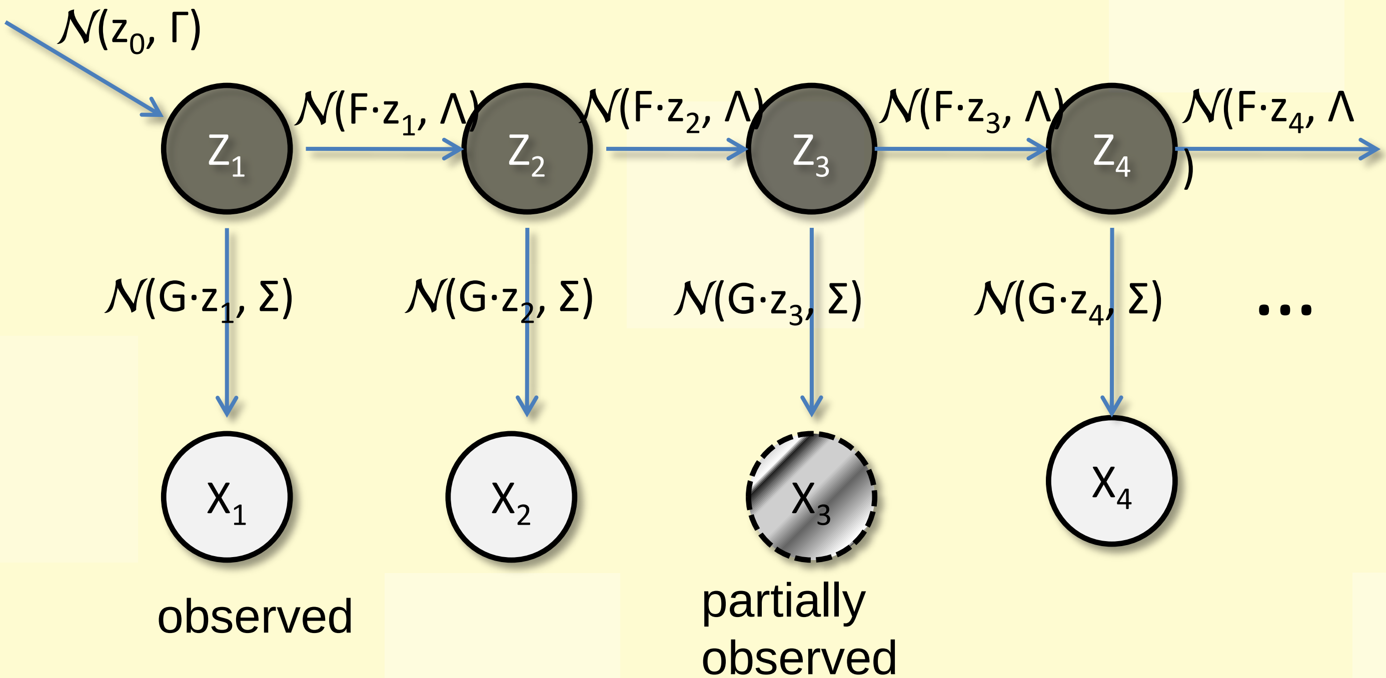
- Motivation
- Background
- P1:Mining w/ Missing Value [Li+ 2009]
- P2:Parallel Learning [Li+ 2008b]
  - Problem Definition
  - Proposed Method
  - Results

Goals for Mining Algorithms

- G1:Effective:
  - achieve low reconstruction error (mean square error)
- G2:Scalable:
  - to the size (e.g. length) of sequences
  - on modern hardware (e.g. multicore)

Recap Model for DynaMMo (details)

Use *Linear Dynamical Systems* to model whole sequence.



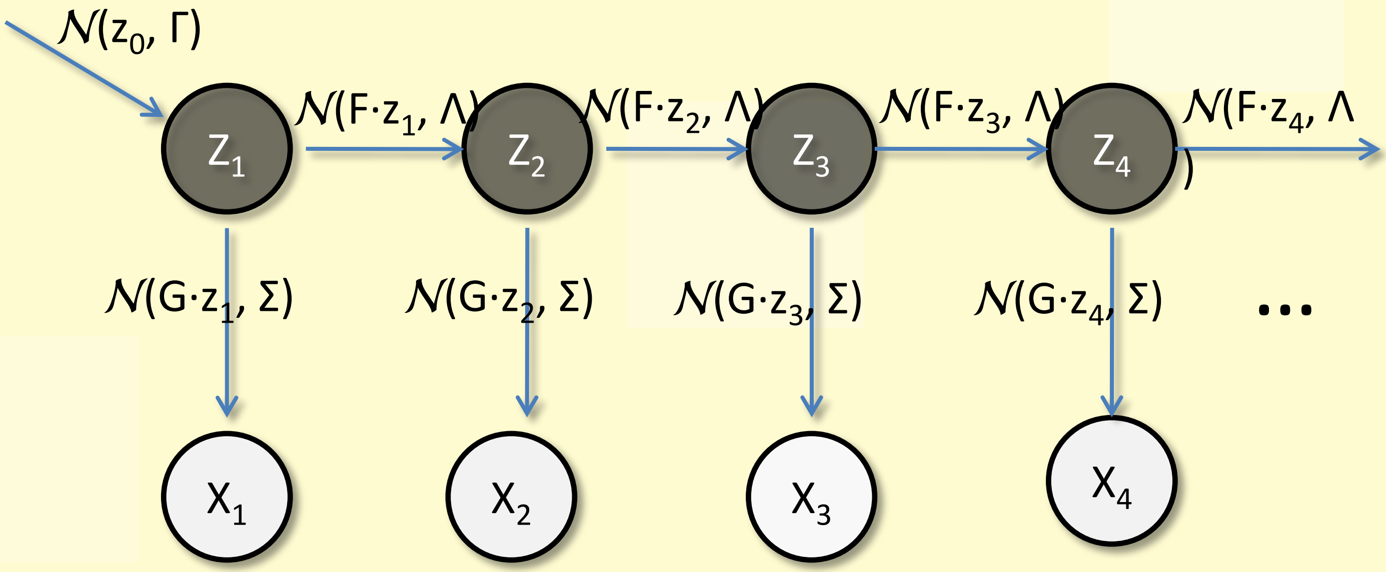
Model parameters:

$\theta=\{z_0, \Gamma, F, \Lambda, G, \Sigma\}$

$$z_1 = z_0 + \omega_0$$
$$z_{n+1} = F \cdot z_n + \omega_n$$
$$x_n = G \cdot z_n + \varepsilon_n$$

Consider a simpler problem (details)

Linear Dynamical Systems (w/o missing values).



observed

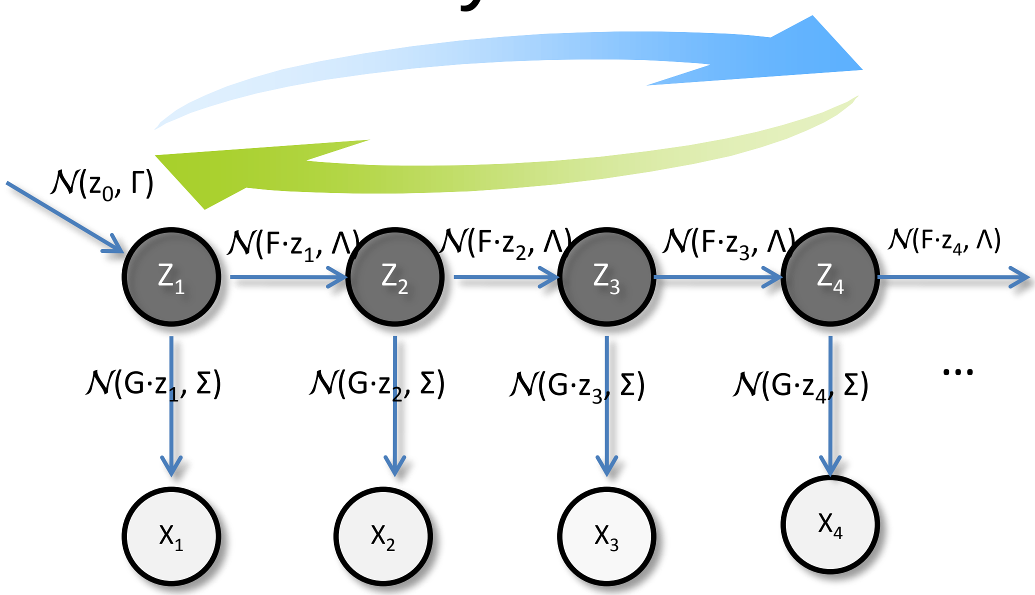
Model parameters:

$\theta=\{z_0, \Gamma, F, \Lambda, G, \Sigma\}$

$$z_1 = z_0 + \omega_0$$
$$z_{n+1} = F \cdot z_n + \omega_n$$
$$x_n = G \cdot z_n + \varepsilon_n$$

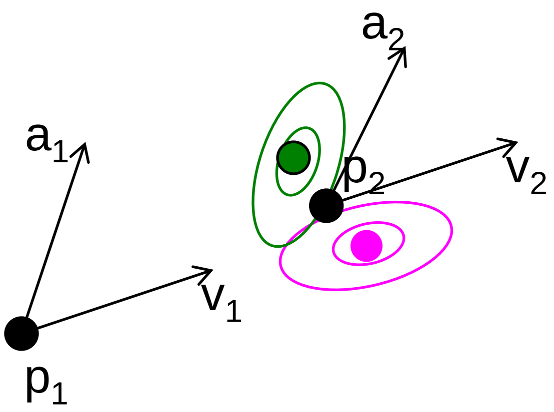
Challenge of Learning LDS: Expectation-Maximization Alg.

- Not easy to parallelize on multi-processors due to non-trivial data dependency (details in writeup)
- Q: How to parallelize the learning to achieve scalability?



Recap: example

tracking moving objects

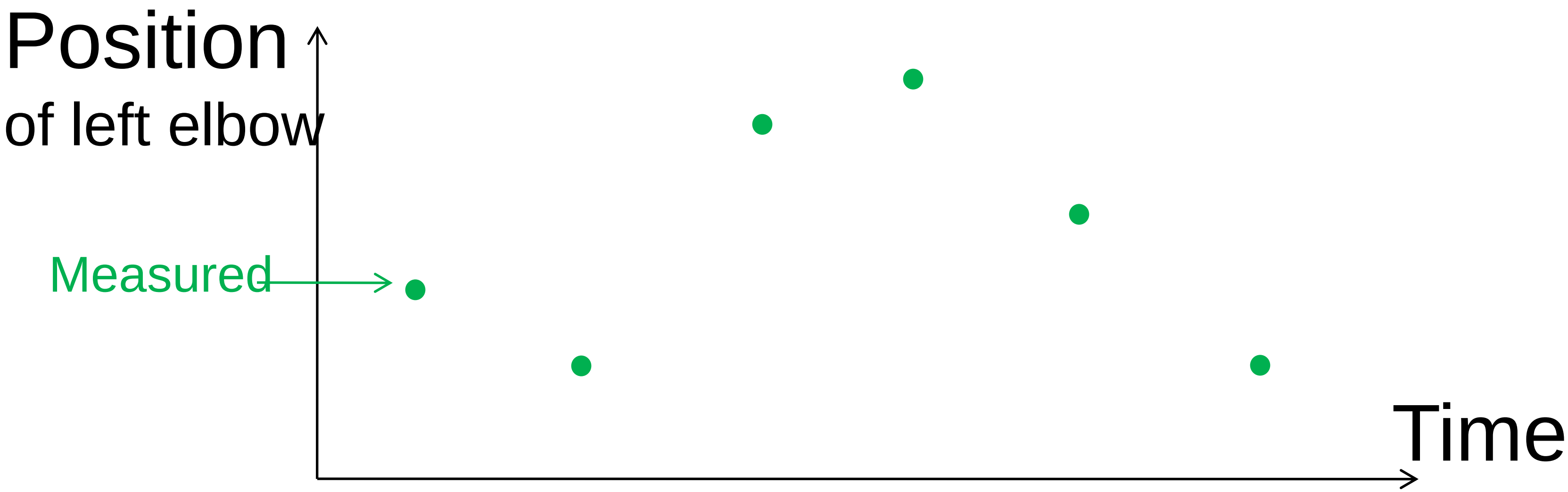


Learning LDS: EM alg.

Goal:

E-step: remove noise, identify true (hidden) trajectory

M-step: find best model parameters

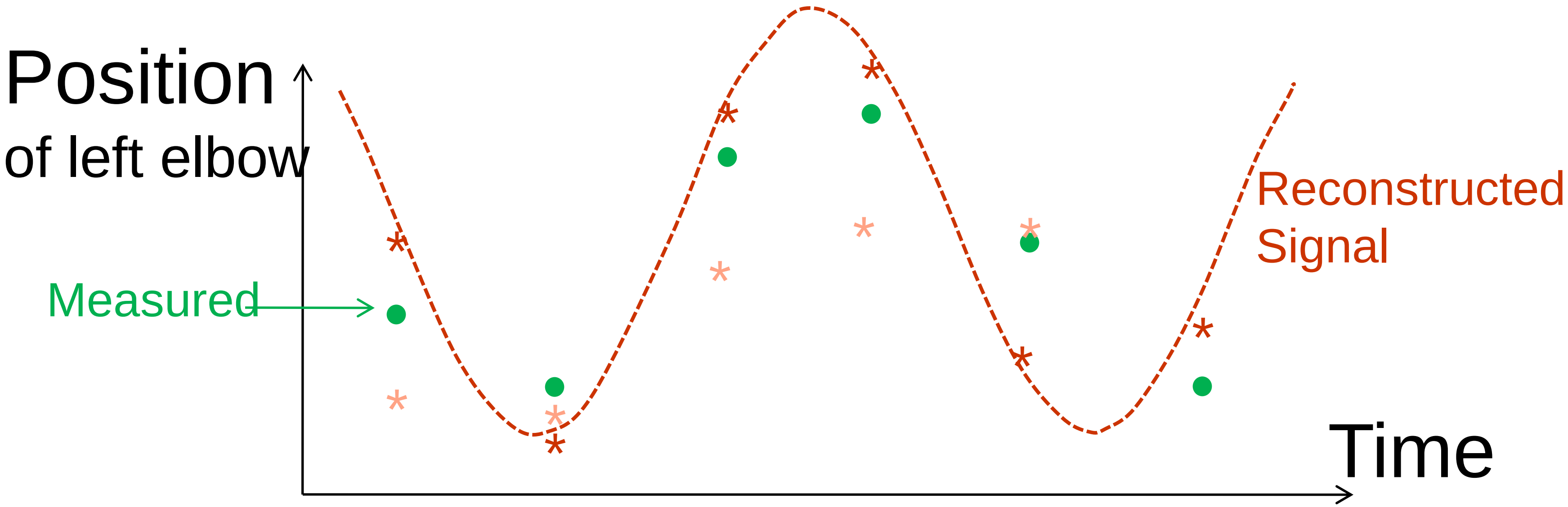


Learning LDS: EM alg.

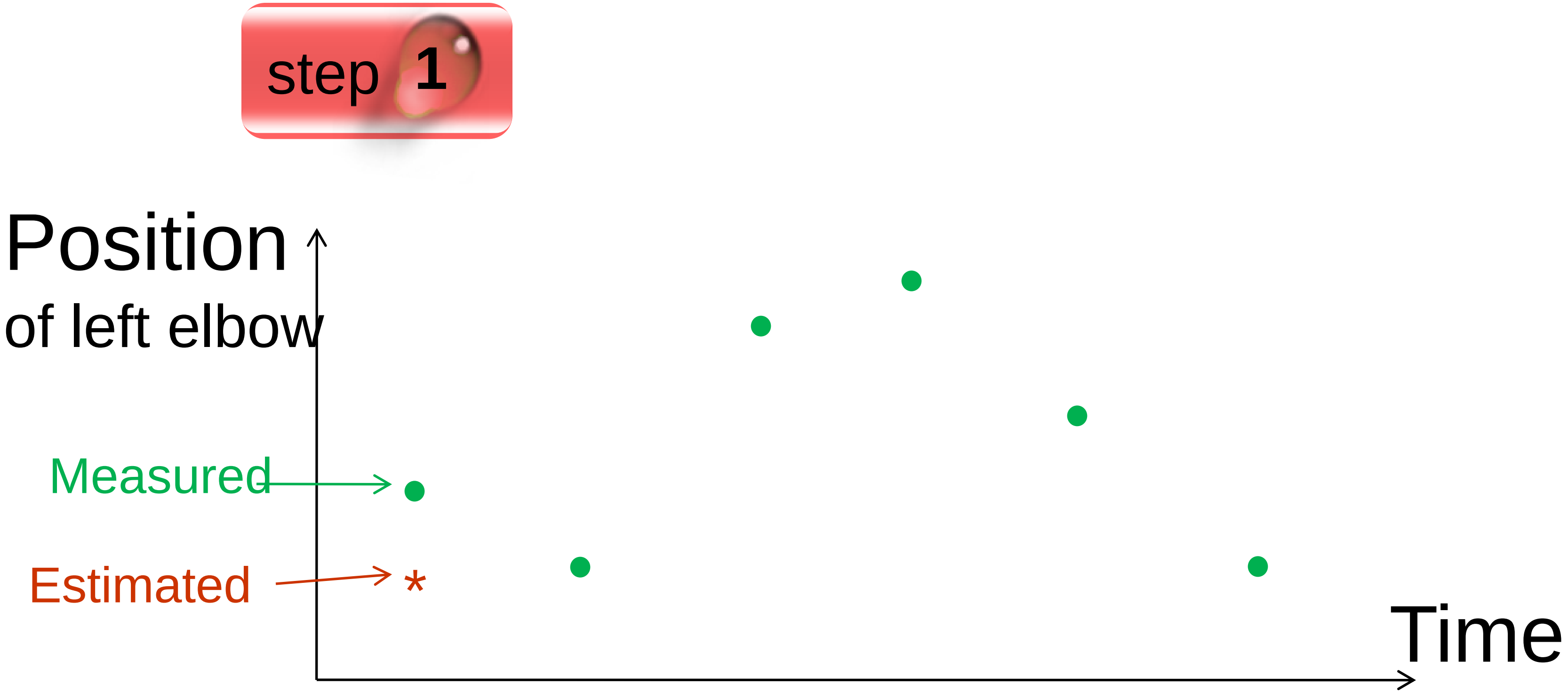
Goal:

E-step: remove noise, identify true (hidden) trajectory

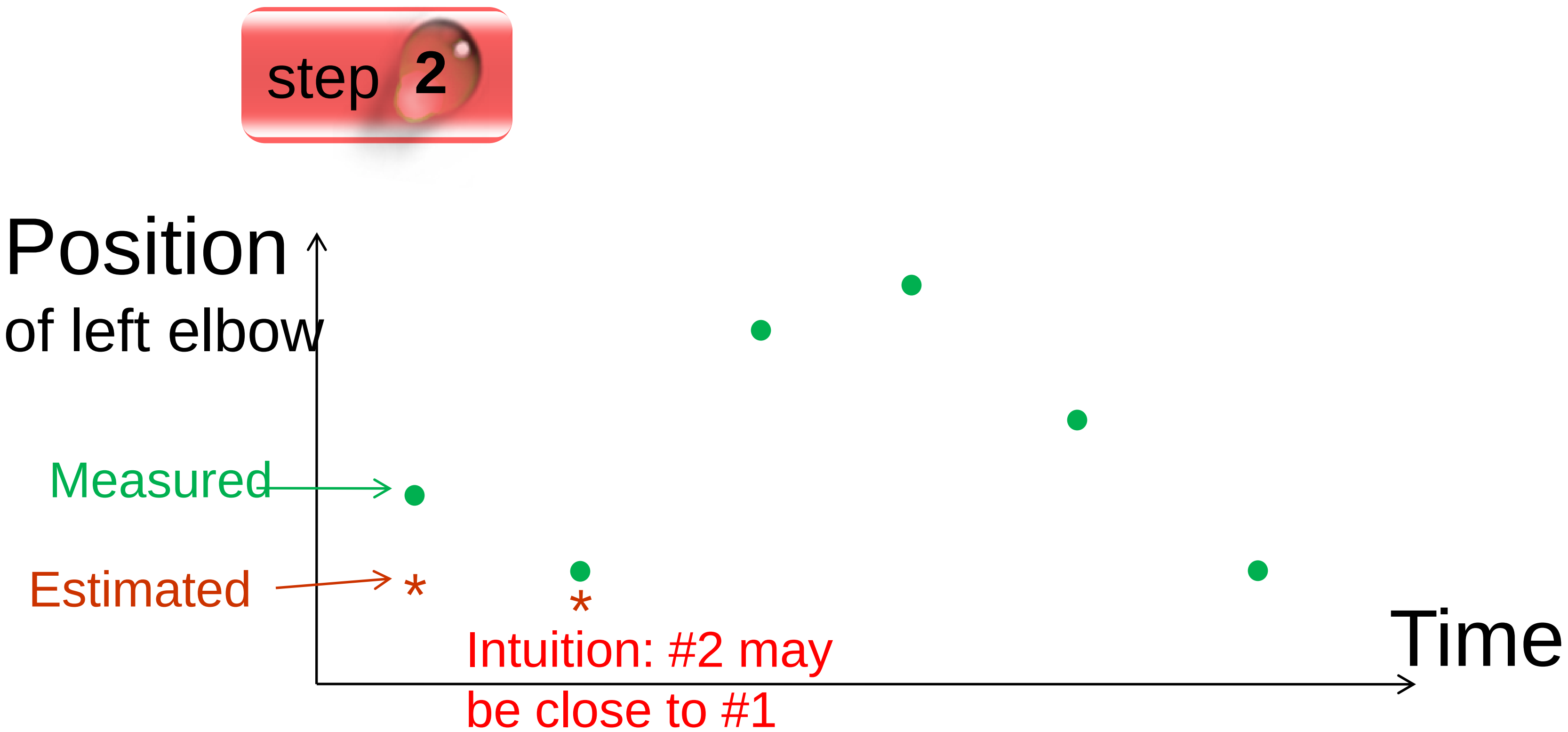
M-step: find best model parameters



Challenge for Learning LDS on SMP

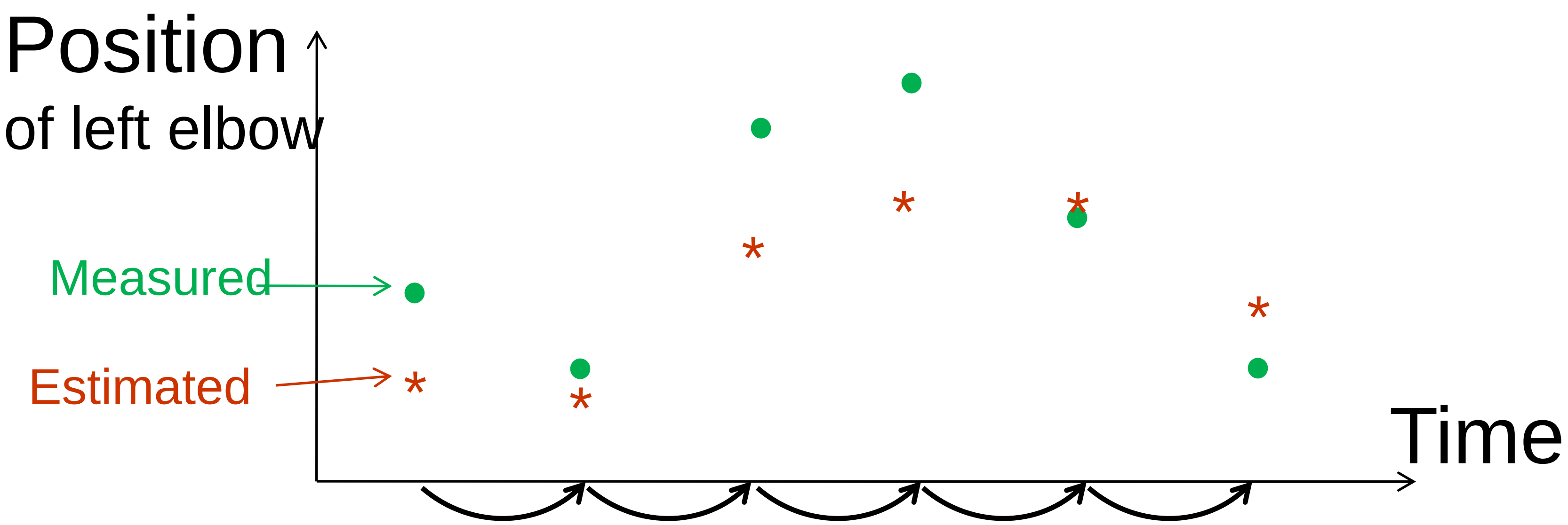


Challenge for Learning LDS on SMP



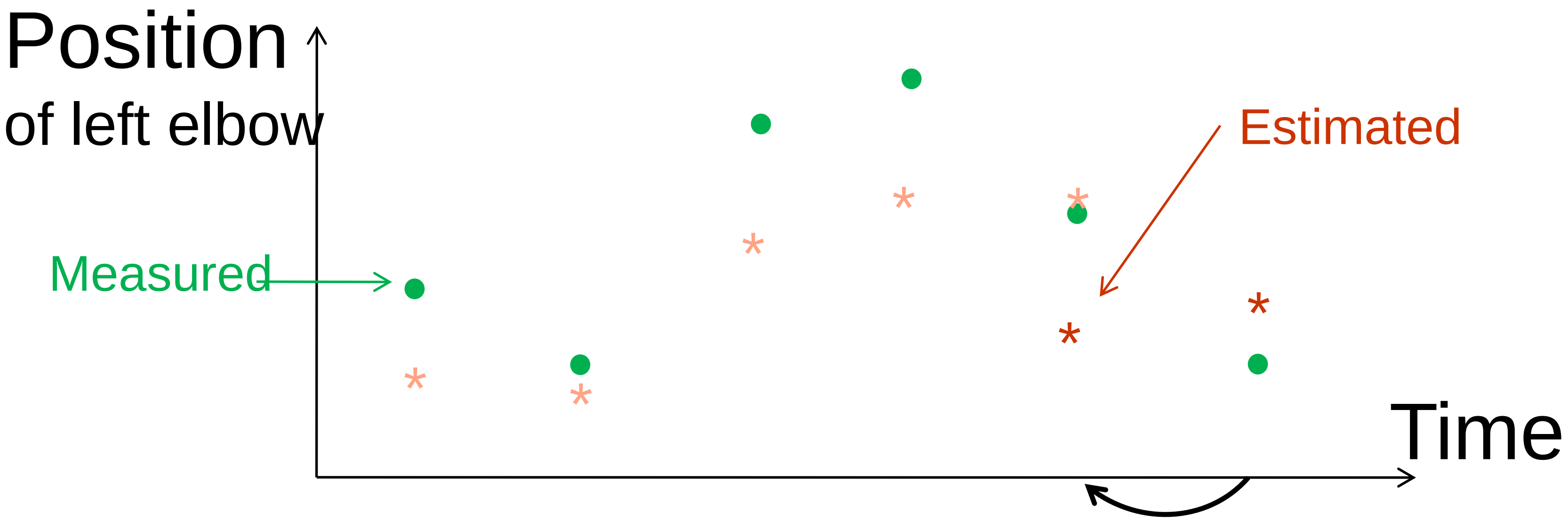
Challenge for Learning LDS on SMP

Forward



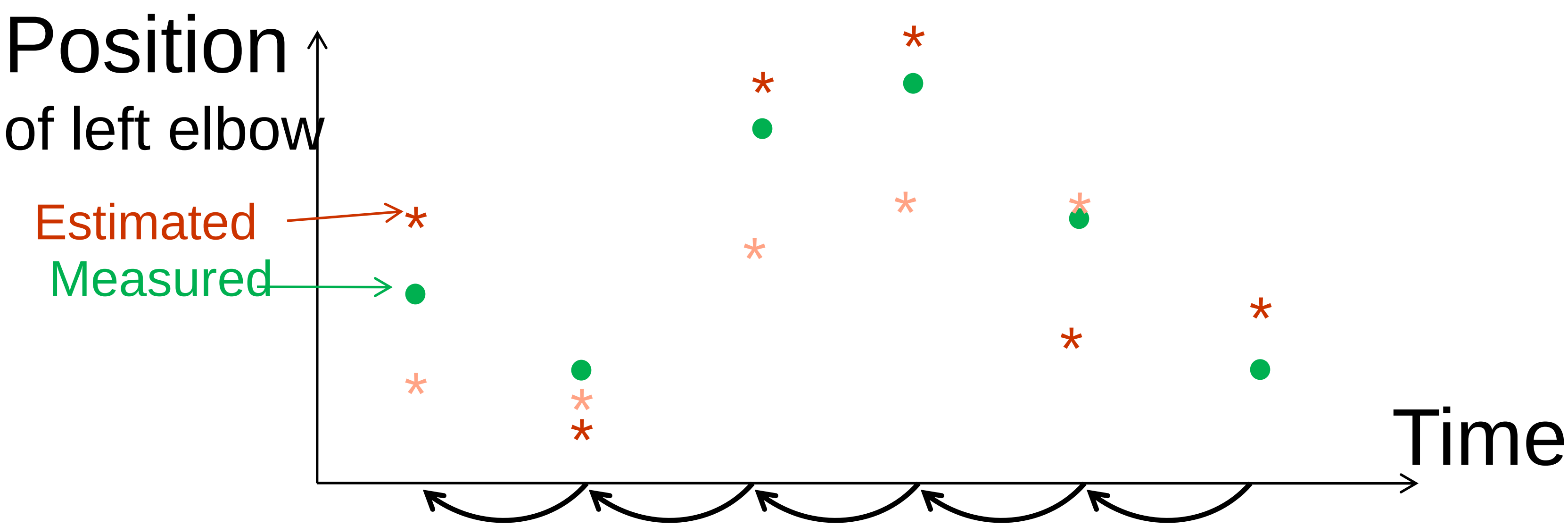
Challenge for Learning LDS on SMP

Backward



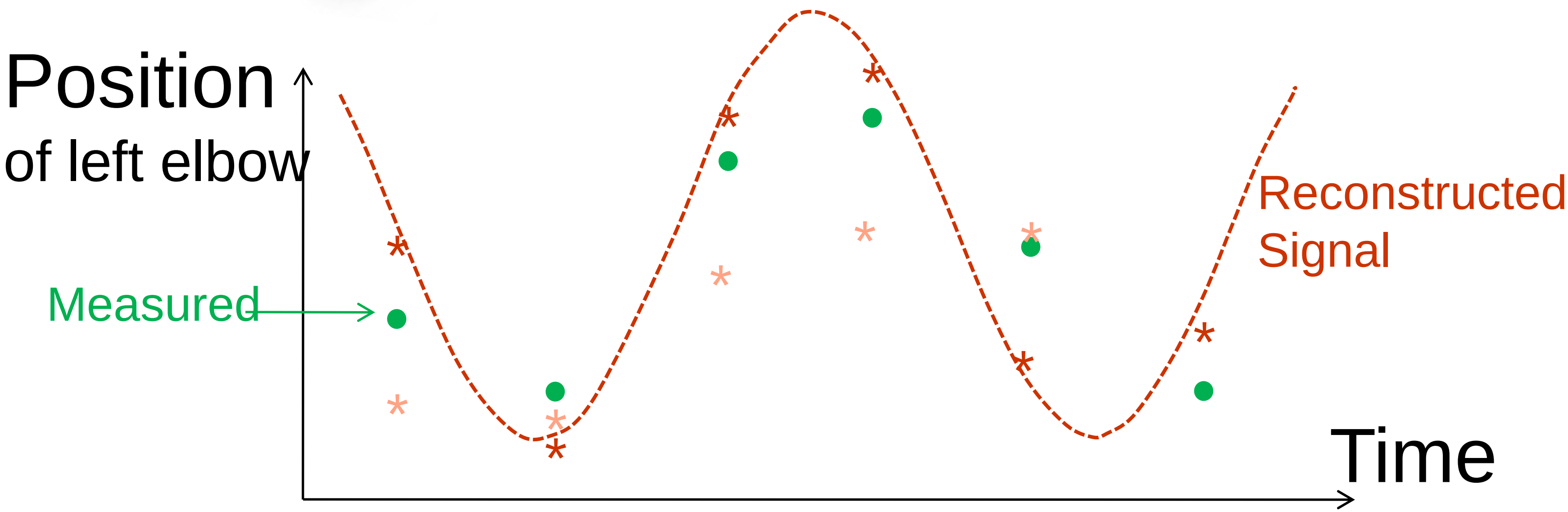
Challenge for Learning LDS on SMP

Backward



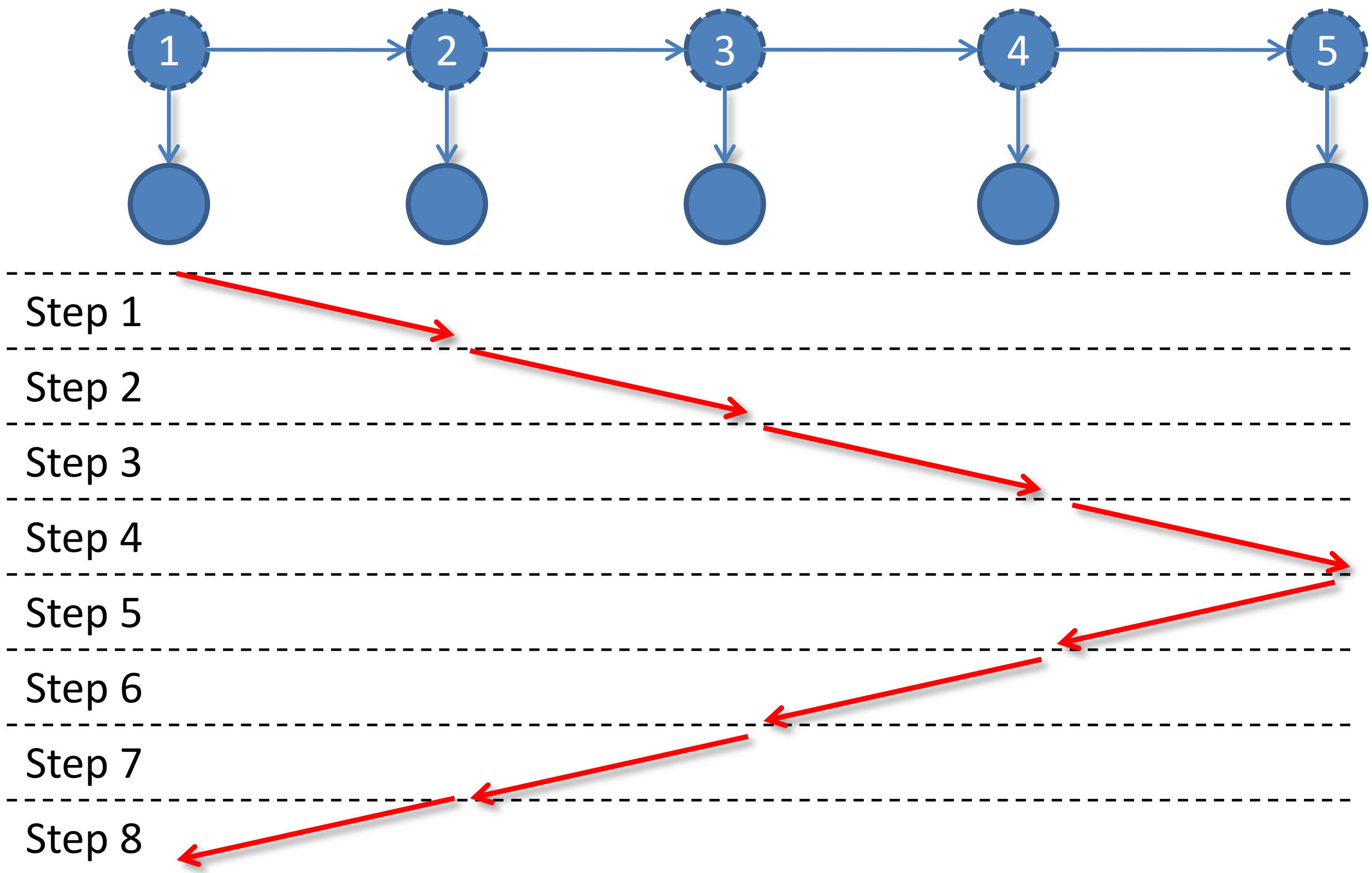
Challenge for Learning LDS on SMP

Backward



Challenge illustration Expectation-Maximization Alg.

Timeline for E-step (forward-backward) in learning LDS



EM can
only uses
Single CPU
Due to data
dependencies

Problem Definition

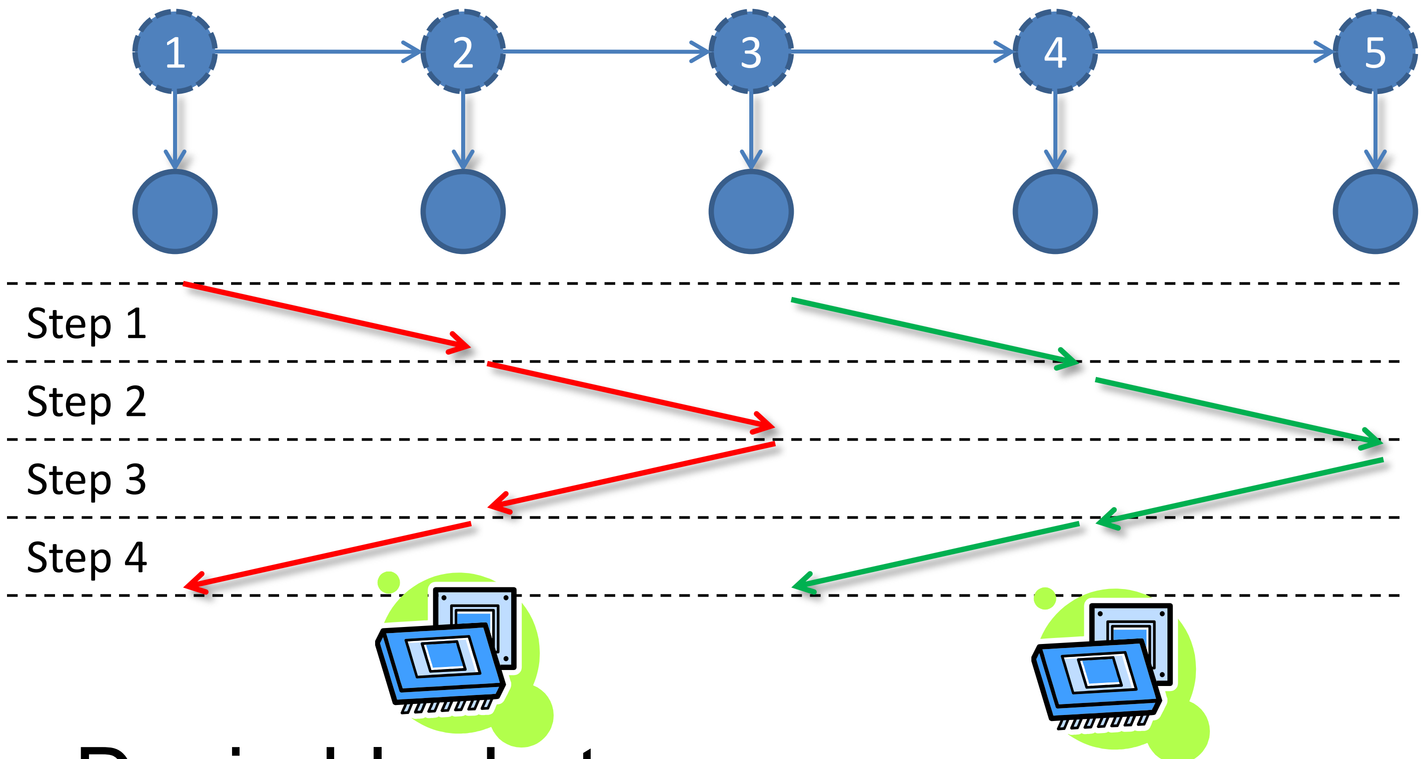
- Problem:
  - **Given** a sequence of numbers, **design** a parallel learning algorithm to find the best model parameters for Linear Dynamical Systems
- Goal:
  - Achieve ~ linear speed up on multi-core
- Assumption:
  - *Shared memory architecture* (e.g. multi-core)

joint work w/ W. Fu, F. Guo, T. Mowry, C. Faloutsos.
[Li et al, KDD 2008]

Proposed Method [Li 2008b]

Expected:

Goal:
with 2 CPUs



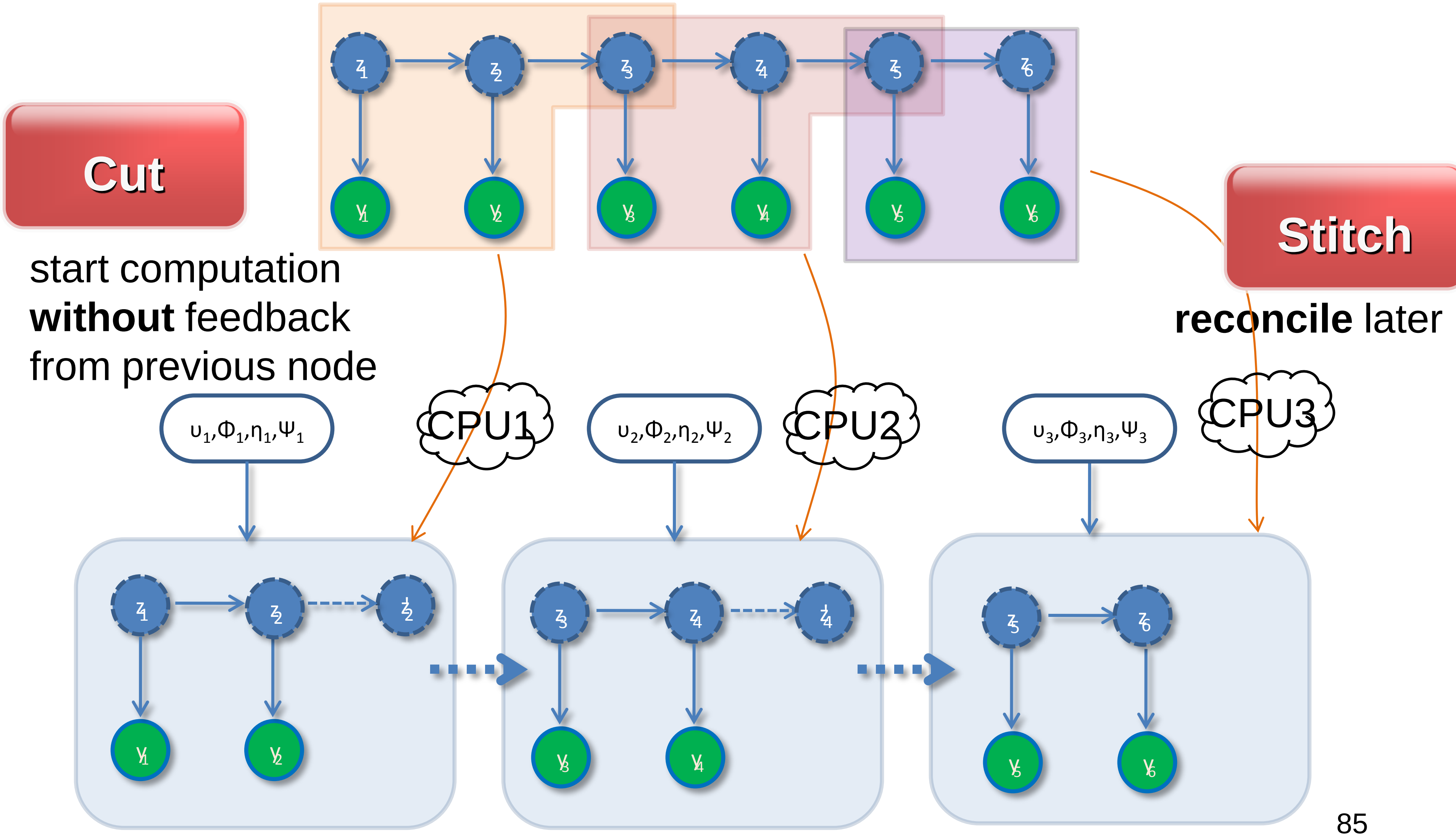
Desirable, but …

data dependencies!

Outline

- Motivation
- Background
- P1:Mining w/ Missing Value [Li+ 2009]
- P2:Parallel Learning [Li+ 2008b]
  - Problem Definition
  - Proposed Method
  - Results
- Conclusion

Cut-And-Stitch Intuition

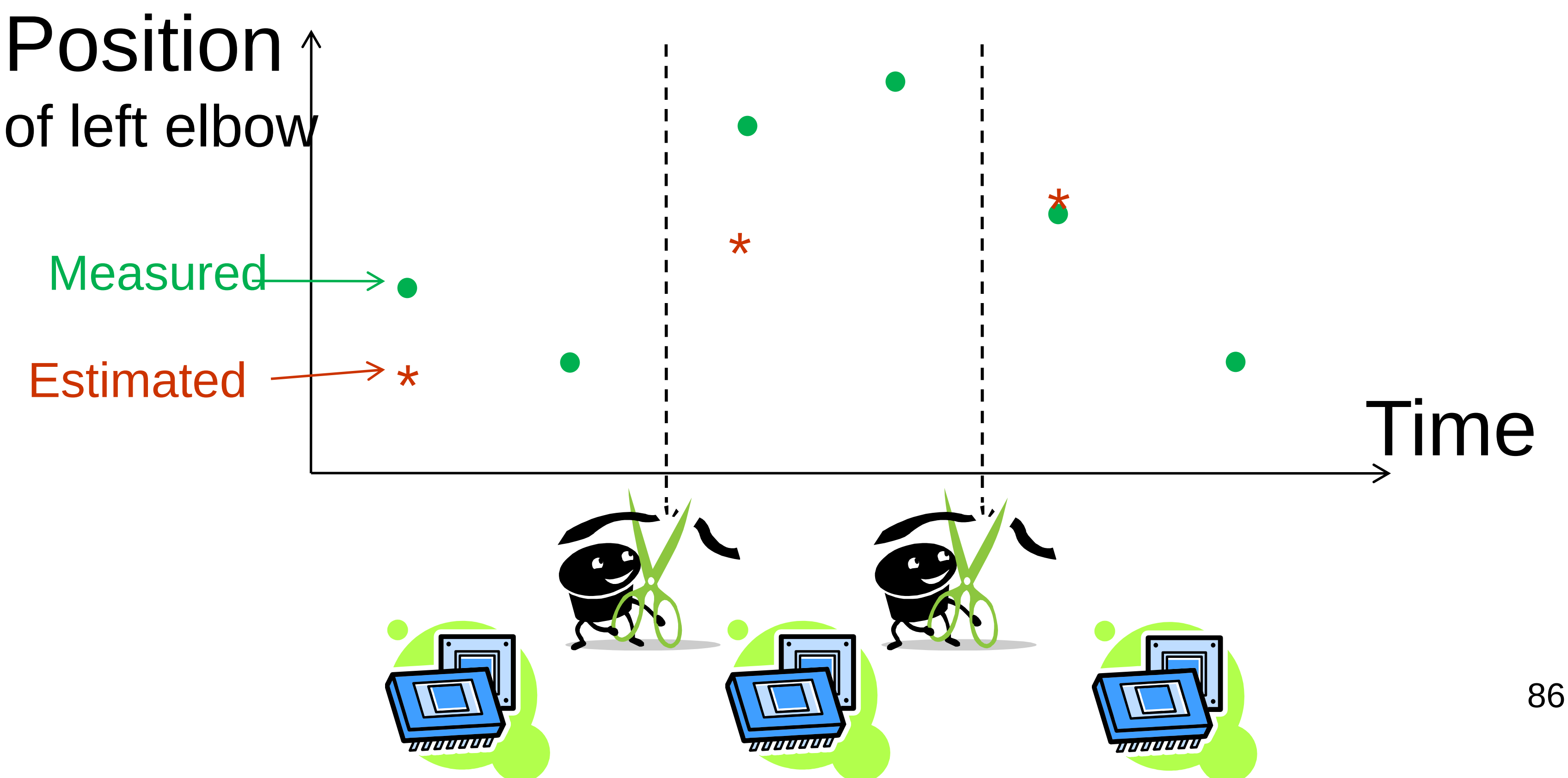


Implemented using OpenMP, details in [Li 2008b]

Cut-And-Stitch: illustration

Details in [Li 2008b]

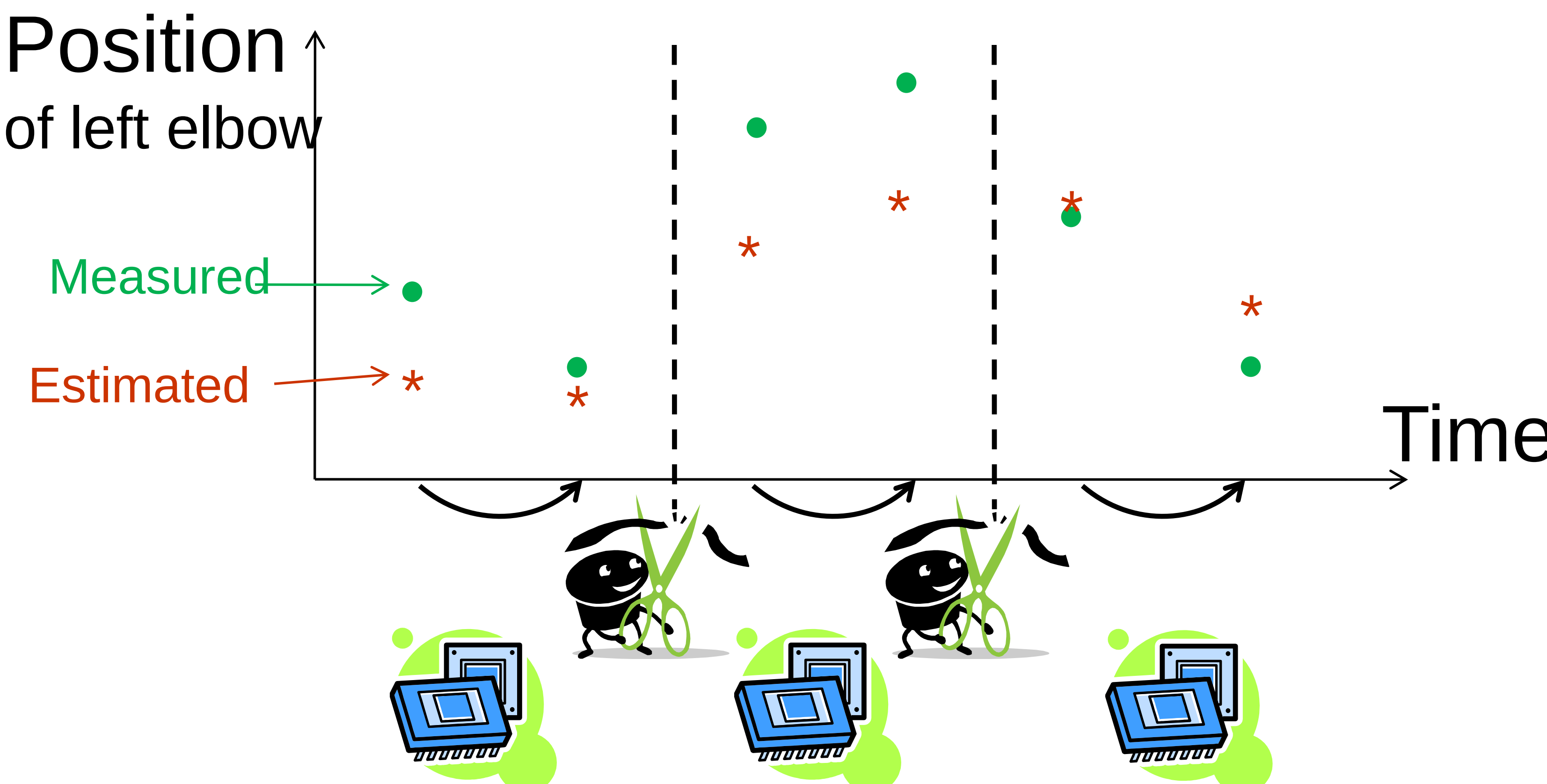
Cut-Forward 1



Cut-And-Stitch: illustration

Details in [Li 2008b]

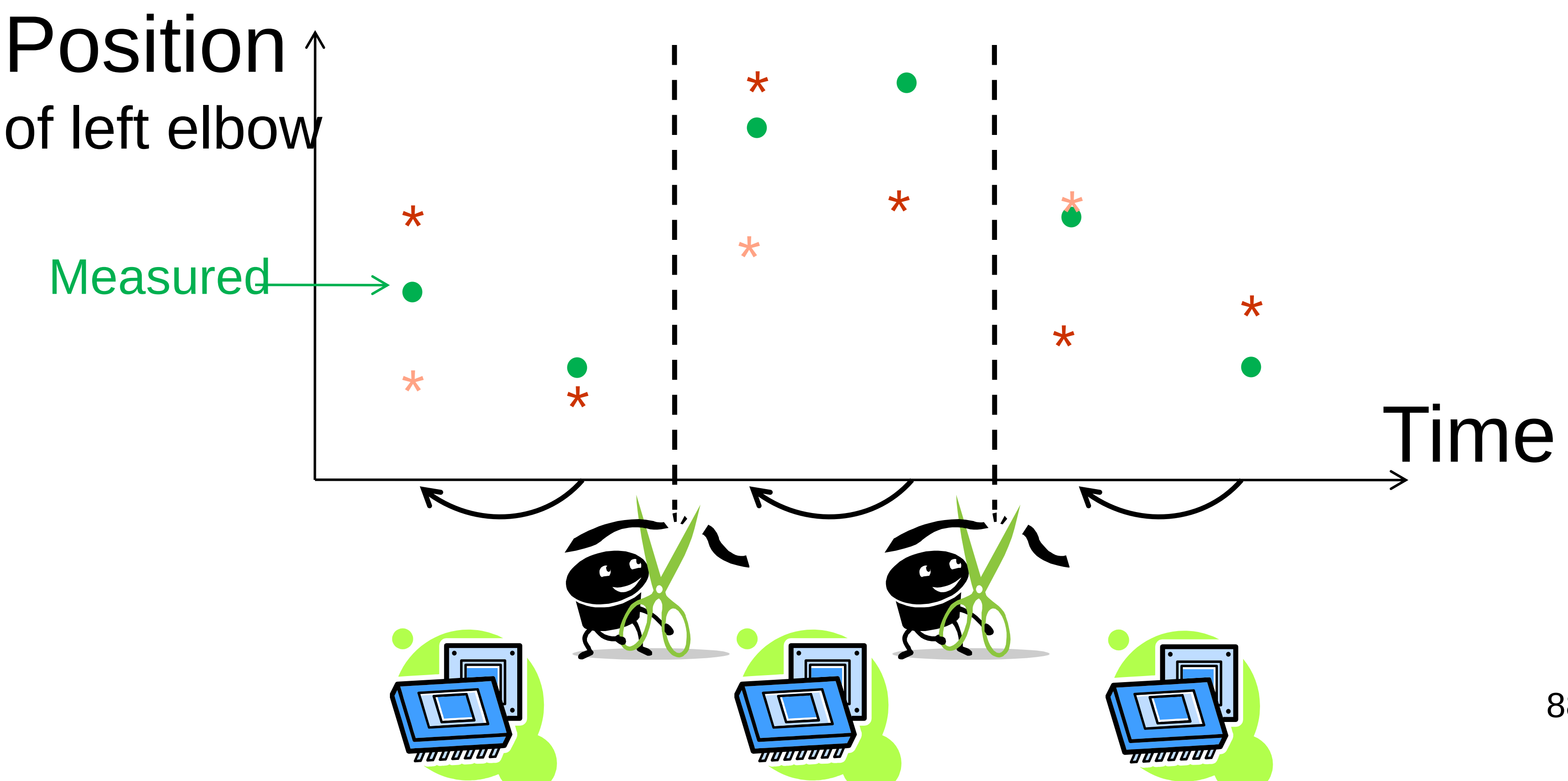
Cut-Forward 2



Cut-And-Stitch: illustration

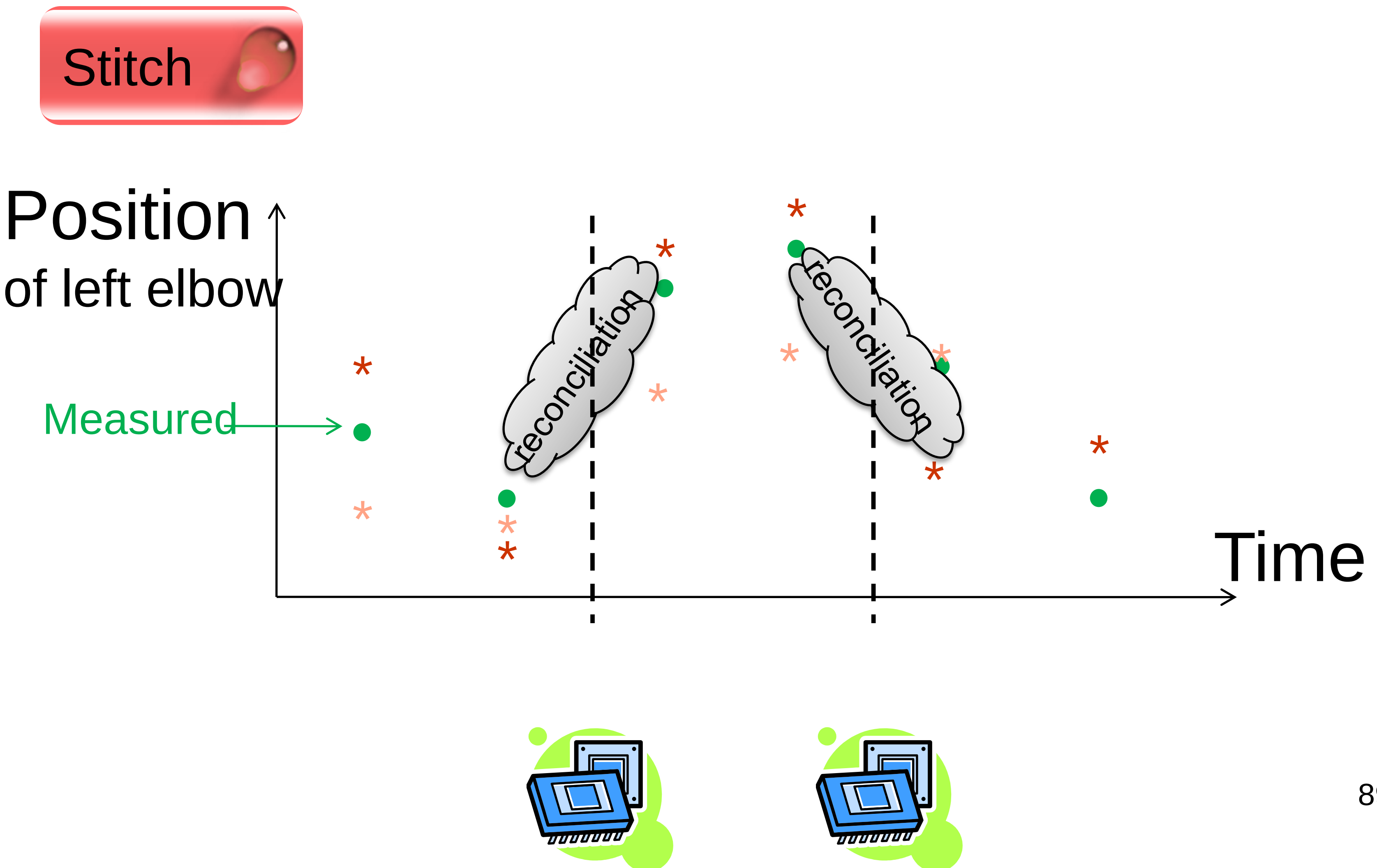
Details in [Li 2008b]

Cut-Backward



Cut-And-Stitch: illustration

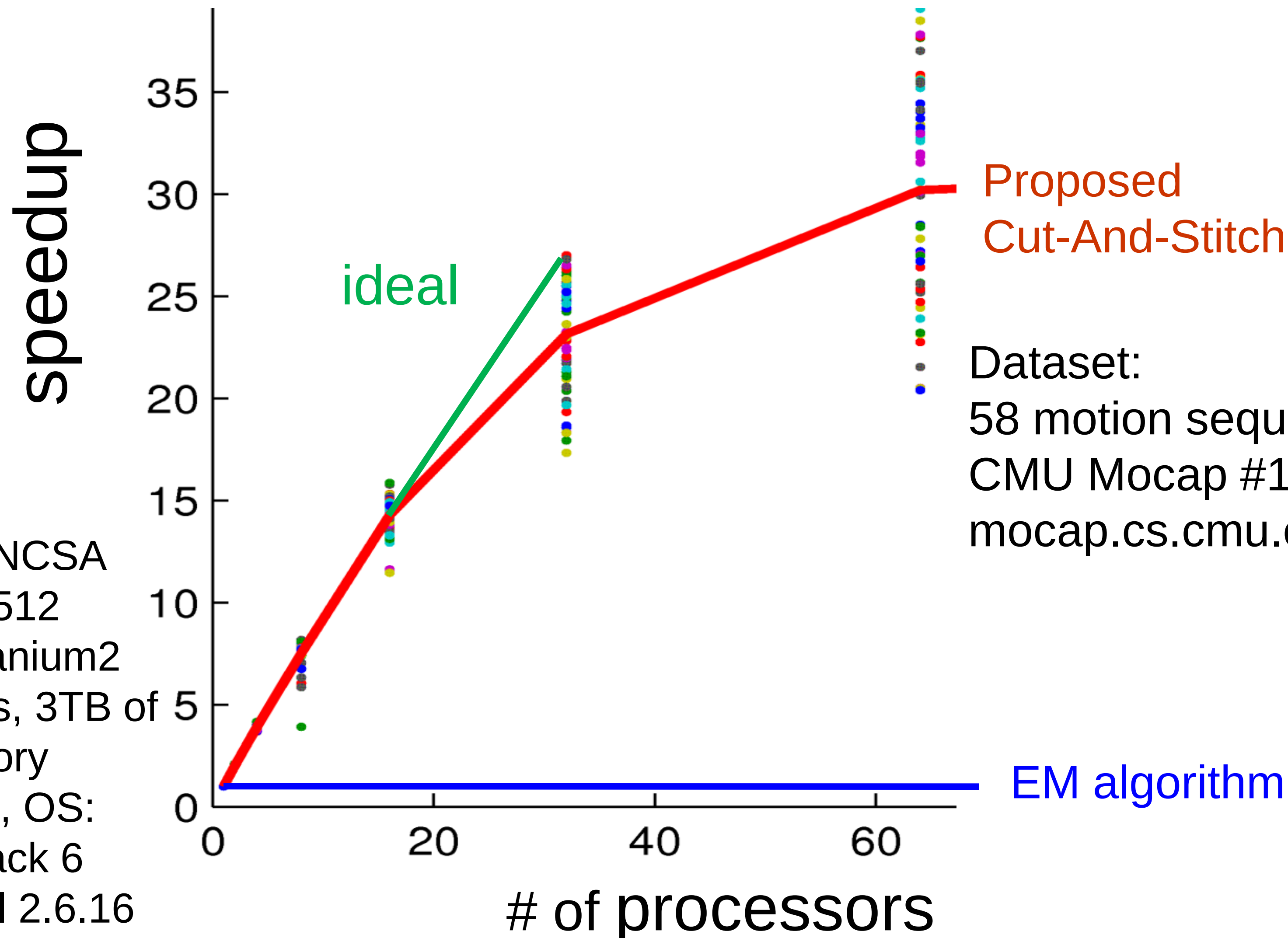
Details in [Li 2008b]



Outline

- Motivation
- Background
- P1:Mining w/ Missing Value [Li+ 2009]
- P2:Parallel Learning [Li+ 2008b]
  - Problem Definition
  - Proposed Method
  - Results
    - Speed up
    - Quality

Near Linear Speedup



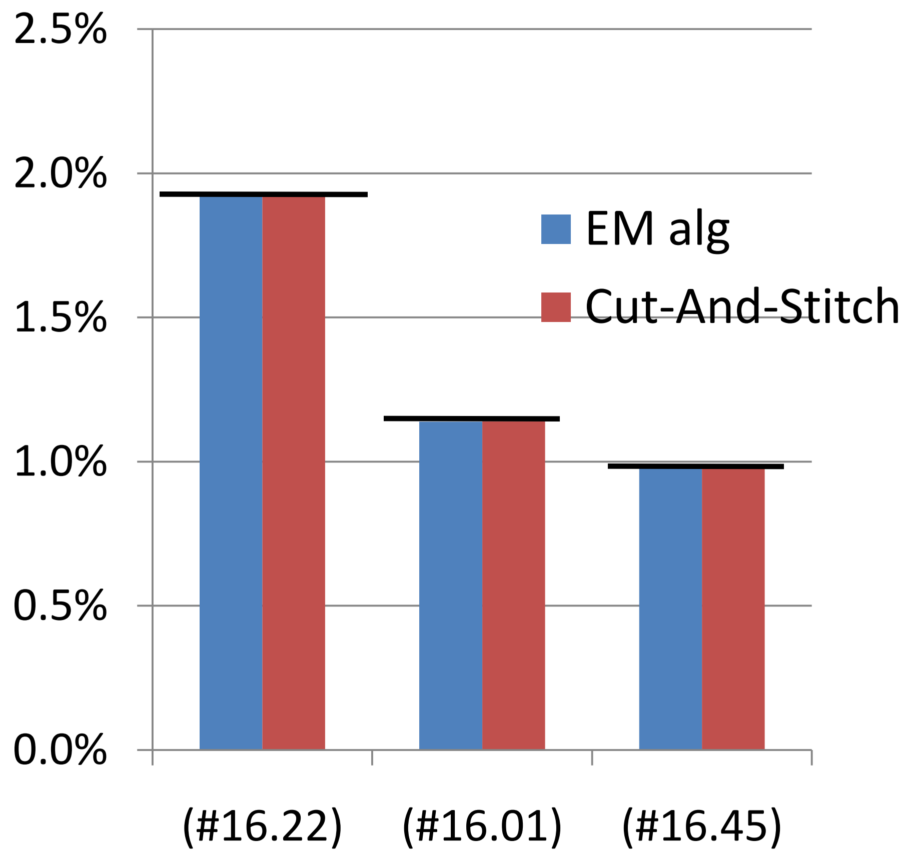
Dataset:
58 motion sequences
CMU Mocap #16
mocap.cs.cmu.edu,

tested on NCSA SGI Altix, 512 1.6GHz Itanium2 processors, 3TB of total memory (ccNUMA), OS: SGI ProPack 6 with kernel 2.6.16 Compiler: Intel 10.1 for C++

more results in [Li+2008b]

No loss of accuracy

Normalized Reconstruction Error



~ IDENTICAL

more results in [Li+2008b]

Outline

- Motivation
- Background
- P1:Mining w/ Missing Value [Li+ 2009]
- P2:Parallel Learning [Li+ 2008b]
  - *Contribution*: the 1st parallel algorithm for learning LDS
- Conclusion

Goals for Mining Algorithms

- ✓ G1:Effective:
  - achieve low reconstruction error (mean square error)
- ✓ G2:Scalable:
  - to the size (e.g. length) of sequences
  - on modern hardware (e.g. multicore)

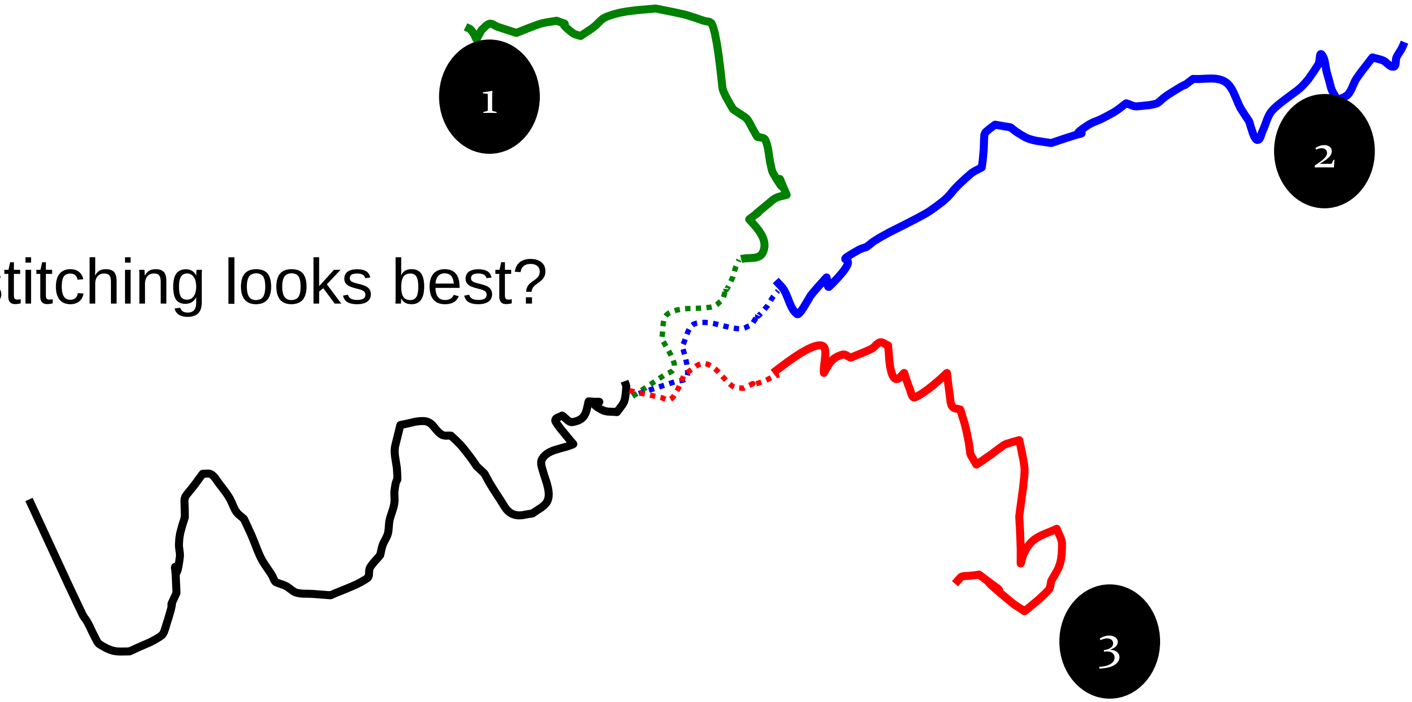
Conclusion

- Pattern discovery w/ missing values (DynaMMo)
  - Recovering missing values
  - Compression
  - Segmentation
- Scale up learning on multicore
  - Parallel learning algorithm for LDS (Cut-And-Stitch)

Additional Projects

- Natural motion stitching [Li et al, Eurographics 2008]
  - Given two motion-capture sequences that are to be stitched together, how can we assess the goodness of the stitching?

Which stitching looks best?



Additional Projects

- CDEM [Guo, Li, Foutsos, Xing. VLDB08]:
  - mining and answering multi-modal queries on drosophila embryo image databases
  - online demo:

    http://www.db.cs.cmu.edu:8080/cdem/
  - System spec:

    Linux kernel: 2.6.23.1

    Tomcat 5.5

    JSP+RMI+mysql

    multi-tier framework

Question

- Thanks!

- contact: Lei Li (leili@cs.cmu.edu)
- paper, software, dataset on http://www.cs.cmu.edu/~leili