CMU SCS

Indexing and Mining Time Sequences
Part 4
Kalman Filters

Christos Faloutsos

Lei Li

CMU

CMU SCS

Outline

- Intuition, example, and definition
- Extensions
- Kalman filters at work

CMU SCS

Intuition

- Tracking moving objects, estimate velocity and acceleration on the fly

from FIFA 2010

RoboCup 2010

CMU SCS

Linear Dynamical System

- Known parameters
  - Original Kalman Filters [Kalman 1960, Rauch 1965]
- Unknown parameters
  - Parameter estimation through EM algorithm [Shumway et al 1982, Ghahramani 1996]

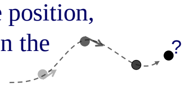
CMU SCS

Kalman Filters

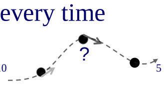
Given observations of the soccer ball position t=1..T, "Model parameters"

Goal: two types of prediction

Kalman filtering: Estimate the true position, velocity & acceleration based on the previous observations



Kalman smoothing: Estimate for every time tick, based on all observations

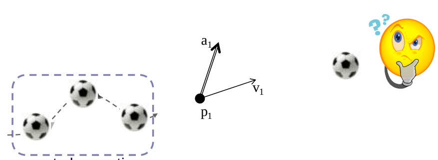


CMU SCS

Kalman Filters (intuition)

t=1, soccer with initial pos, vel and acc.

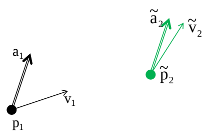
To estimate the future



CMU SCS

Kalman Filters (intuition)

t=2, according to Newton's law, it should be...

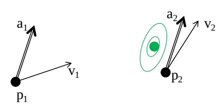


CMU SCS

Kalman Filters (intuition)

t=2, according to Newton's law, it should be...

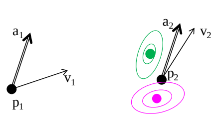
however, imperfect soccer/kick movement…



CMU SCS

Kalman Filters (intuition)

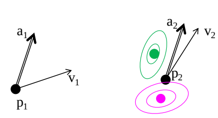
Now take a photo, due to imperfect camera…



CMU SCS

Kalman Filters (intuition)

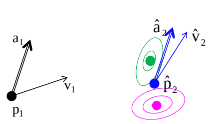
What is the best estimate for next time tick?



CMU SCS

Kalman Filters (intuition)

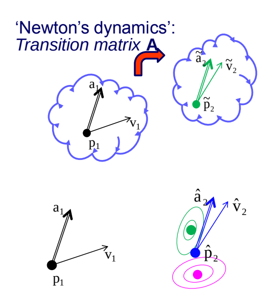
What is the best estimate for next time tick?



CMU SCS

Some math notation

	name
A	transition matrix
C	transmission/ projection/ output matrix
Q	transition covariance
R	transmission/ projection/ output covariance



CMU SCS

(details)

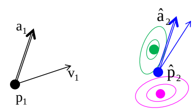
Example

hidden states $z_1=(p_1, v_1, a_1)^T$

observation $x_1=(\text{observed}_1)$

transition matrix $A=\begin{pmatrix}1,1,1/2\\0, 1, 1\\0, 0, 1\end{pmatrix}$

output matrix $C=(1, 0, 0)$



transition covariance Q

output covariance R

CMU SCS

(details)

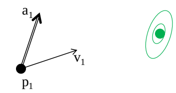
Example

hidden states $z_1=(p_1, v_1, a_1)^T$

observation $x_1=(\text{observed}_1)$

transition matrix $A=\begin{pmatrix}1,1,1/2\\0, 1, 1\\0, 0, 1\end{pmatrix}$

output matrix $C=(1, 0, 0)$



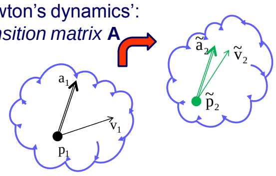
$p_2=p_1+ v_1*\Delta t + 0.5*a_1*\Delta t^2$
$v_2=v_1+a_1*\Delta t$
$a_2=a_1$

CMU SCS

Kalman Filtering (intuition)

Step1, forecast next time tick before observation

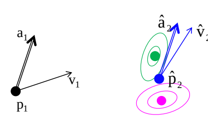
‘Newton’s dynamics’:
Transition matrix **A**



CMU SCS

Kalman Filtering (intuition)

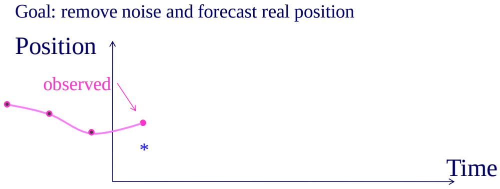
Step 2: adjust estimation after observation



CMU SCS

Example: Kalman filtering (forward)

Given:
a sequence of observations, Model parameters (A, C …)
Goal: remove noise and forecast real position



CMU SCS

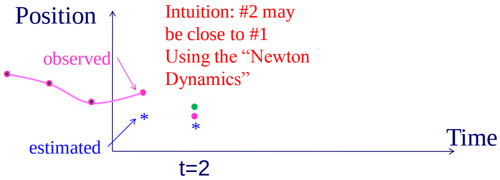
Example: Kalman filtering (forward)

[R. E. Kalman. A new approach to linear filtering and prediction problems. 1960]

CMU SCS

Example: Kalman filtering

$$\hat{z}_n = A\cdot\hat{z}_{n-1} + K_n\cdot(x_n - C\cdot A\cdot\hat{z}_{n-1})$$
$$\hat{V}_n = (I - K_n)\cdot P_{n-1}$$
$$K_n = P_{n-1}\cdot C^T\cdot(C\cdot P_{n-1}\cdot C^T + R)^{-1}$$
$$P_{n-1} = A\cdot\hat{V}_{n-1}\cdot A^T + Q$$

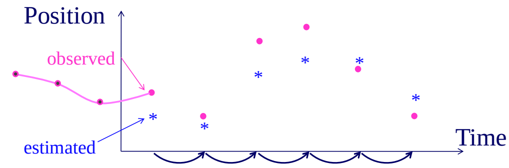


KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 19

CMU SCS

Example: Kalman filtering

$$\hat{z}_n = A\cdot\hat{z}_{n-1} + K_n\cdot(x_n - C\cdot A\cdot\hat{z}_{n-1})$$
$$\hat{V}_n = (I - K_n)\cdot P_{n-1}$$
$$K_n = P_{n-1}\cdot C^T\cdot(C\cdot P_{n-1}\cdot C^T + R)^{-1}$$
$$P_{n-1} = A\cdot\hat{V}_{n-1}\cdot A^T + Q$$



KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 20

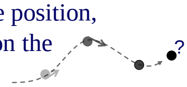
CMU SCS

Kalman Filters

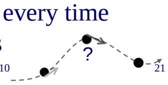
Given observations of the soccer position t=1..T, and model parameters (A, C …)

Goal: two types of prediction

Kalman filtering: Estimate the true position, velocity & acceleration based on the previous observations



Kalman smoothing: Estimate for every time tick, based on all observations



KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 21

CMU SCS

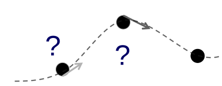
Kalman Smoothing

Given: all observation $x_1, \ldots, x_n$

Estimate: the hidden state for every time tick z_t (t=1..n)

Difference from Kalman filtering:

bring future observation back in history estimate



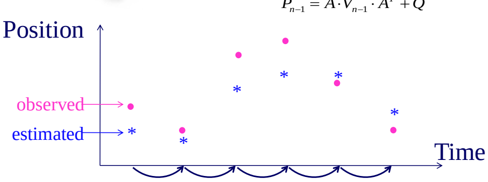
KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 22

CMU SCS

Recap: Kalman filtering

Forward

$$\hat{z}_n = A\cdot\hat{z}_{n-1} + K_n\cdot(x_n - C\cdot A\cdot\hat{z}_{n-1})$$
$$\hat{V}_n = (I - K_n)\cdot P_{n-1}$$
$$K_n = P_{n-1}\cdot C^T\cdot(C\cdot P_{n-1}\cdot C^T + R)^{-1}$$
$$P_{n-1} = A\cdot\hat{V}_{n-1}\cdot A^T + Q$$



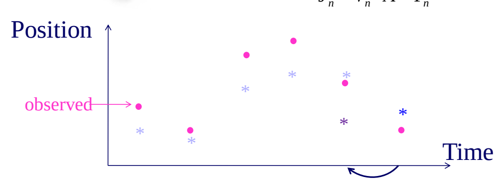
KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 23

CMU SCS

Example: Kalman Smoothing

Backward 1

$$\hat{\hat{z}}_n = \hat{z}_n + J_n\cdot(\hat{\hat{z}}_{n+1} - A\cdot\hat{z}_n)$$
$$\hat{\hat{V}}_n = \hat{V}_n + J_n\cdot(\hat{\hat{V}}_{n+1} - P_n)\cdot J_n^T$$
$$J_n = \hat{V}_n\cdot A^T\cdot P_n^{-1}$$



KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 24

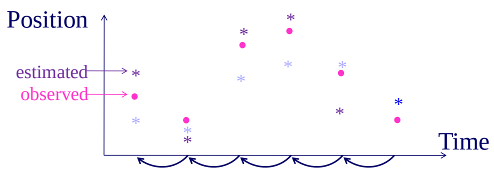
[Rauch et al, Maximum likelihood estimates of linear dynamic systems. 1965]

CMU SCS

Example: Kalman Smoothing

Backward 2

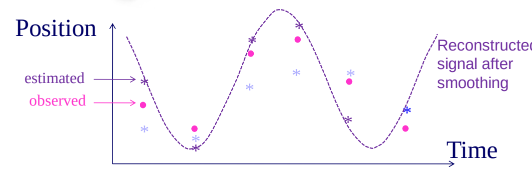
$$\hat{\hat{z}}_n = \hat{z}_n + J_n \cdot (\hat{\hat{z}}_{n+1} - A \cdot \hat{z}_n)$$
$$\hat{\hat{V}}_n = \hat{V}_n + J_n \cdot (\hat{\hat{V}}_{n+1} - P_n) \cdot J_n^T$$
$$J_n = \hat{V}_n \cdot A^T \cdot P_n^{-1}$$



CMU SCS

Example: Kalman Smoothing

Backward



CMU SCS

Outline

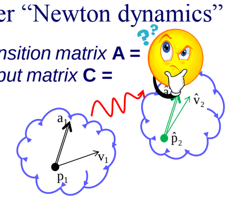
- Intuition, example, and definition
  - Original Kalman
  - Kalman filters with parameter estimation
- Extensions
- Kalman filters at work

CMU SCS

What if not know the model parameters?

- E.g. Datacenter sensor temperatures
- no longer "Newton dynamics"

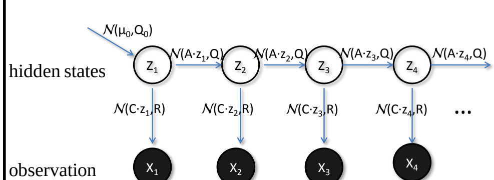
Transition matrix **A =**
output matrix **C =**



CMU SCS

(details)

Graphical Model Representation



Model parameters:
$\theta=\{\mu_0, Q_0, A, Q, C, R\}$

$$z_1 = \mu_0 + \omega_0$$
$$z_{n+1} = A \cdot z_n + \omega_n$$
$$x_n = C \cdot z_n + \varepsilon_n$$

Gaussian noise

CMU SCS

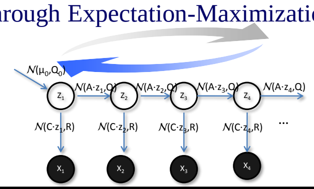
Linear Dynamical Systems: parameters

	name	meaning & example
μ_0	initial state for hidden variable	e.g. initial position, velocity & acceleration
A	transition matrix	how the states move forward, e.g. soccer flying in the air
C	transmission/ projection/ output matrix	hidden state ➔ observation, e.g. camera taking picture of the soccer
Q_0	Initial covariance	
Q	transition covariance	how precision is the soccer motion
R	transmission/ projection covariance	i.e. observation noise; e.g. how accurate is the camera

CMU SCS

Estimating parameters of LDS

- Given: a sequence of observations (e.g. car positions)
- Find: best-fit parameters
- Basic principle: maximum likelihood
- Through Expectation-Maximization Alg.

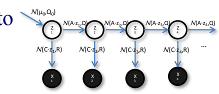


31

CMU SCS

Learning LDS: EM alg.

- E-step: Kalman filtering-smoothing
  – Estimate the hidden variables based on observation and current parameters
- M-step:
  – Update parameters (A, C…) to maximize



$$L(\theta;\mathcal{X}) = \mathbb{E}_{\mathcal{X},\vec{z}|\theta}[-D(\vec{z}_1,\vec{\mu}_0,\mathbf{Q}_0) - \sum_{t=2}^{T} D(\vec{z}_t,\mathbf{A}\vec{z}_{t-1},\mathbf{Q}) - \sum_{t=1}^{T} D(\vec{x}_t,\mathbf{C}\vec{z}_t,\mathbf{R}) - \frac{1}{2}\log|\mathbf{Q}_0| - \frac{T-1}{2}\log|\mathbf{Q}| - \frac{T}{2}\log|\mathbf{R}|]$$

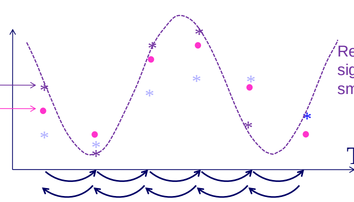
KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 32

[Shumway et al 1982, Ghahramani 1996]

CMU SCS

EM alg. Intuition

E-step: compute hidden states using Kalman filtering-smoothing (exactly as before)



KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 33

CMU SCS

EM alg.

M-step: Maximizing the log-likelihood

$$L(\theta;\mathcal{X}) = \mathbb{E}_{\mathcal{X},\vec{z}|\theta}[-D(\vec{z}_1,\vec{\mu}_0,\mathbf{Q}_0) - \sum_{t=2}^{T} D(\vec{z}_t,\mathbf{A}\vec{z}_{t-1},\mathbf{Q}) - \sum_{t=1}^{T} D(\vec{x}_t,\mathbf{C}\vec{z}_t,\mathbf{R}) - \frac{1}{2}\log|\mathbf{Q}_0| - \frac{T-1}{2}\log|\mathbf{Q}| - \frac{T}{2}\log|\mathbf{R}|]$$

by solving

$$\frac{\partial L}{\partial A} = 0$$

$$\frac{\partial L}{\partial C} = 0$$

…

KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 34

CMU SCS

Outline

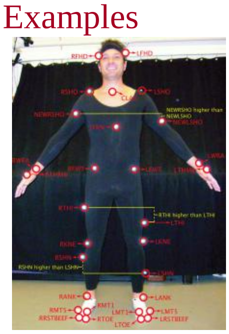
- Intuition, example, and definition
- Extensions
  – Handling missing values
  – Switching LDS
  – Particle filters
- Kalman filters at work

KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 35

CMU SCS

Motivation Examples

- 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



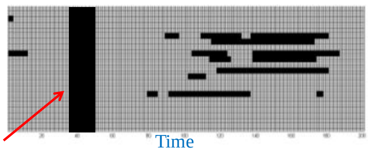
From mocap.cs.cmu.edu

KDD 2010 Copyright: C. Faloutsos & L. Li, 2010 36

CMU SCS

Illustration of data

sensor 1
sensor 2
…
$sensor_m$
blackout

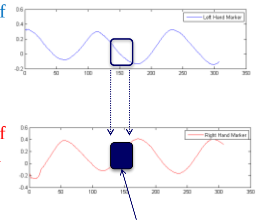


CMU SCS

DynaMMo: Intuition

Position of Left hand marker

Position of right hand marker



Recover using Correlation among multiple sequences

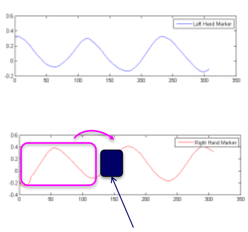
[Lei Li et al DynaMMo. KDD 2009]

CMU SCS

DynaMMo: Intuition

Position of Left hand marker

Position of right hand marker



Recover using Dynamics temporal moving pattern

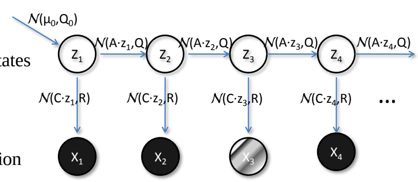
[Lei Li et al DynaMMo. KDD 2009]

CMU SCS

LDS with Missing Value

hidden states

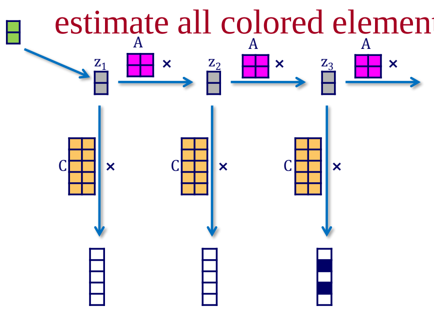
observation



partially observed

CMU SCS

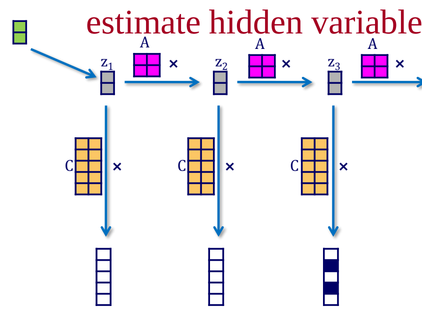
DynaMMo Illustration: estimate all colored elements



Details in [Li+2009]

CMU SCS

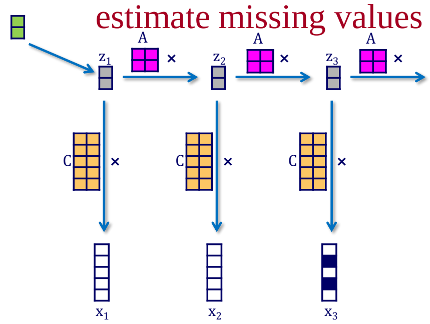
DynaMMo Illustration: step 1 estimate hidden variables



Details in [Li+2009]

CMU SCS

DynaMMo Illustration: step 2 estimate missing values

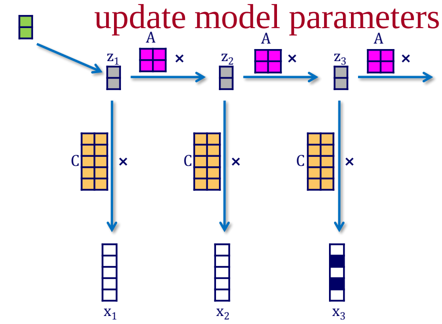


Details in [Li+2009]

43

CMU SCS

DynaMMo Illustration: step 3 update model parameters



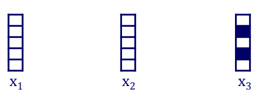
Details in [Li+2009]

44

CMU SCS

DynaMMo Intuition, forward-backward algorithm

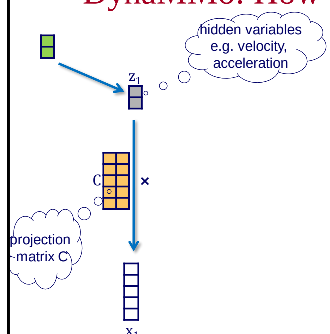
- How to recover the missing values?



45

CMU SCS

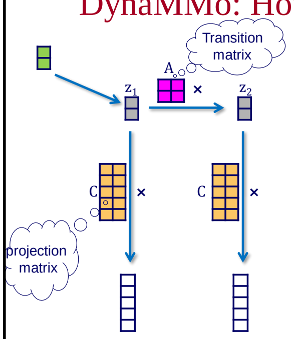
DynaMMo: How to Recover?



46

CMU SCS

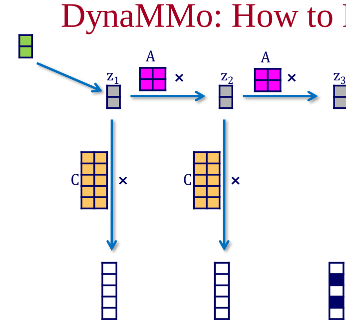
DynaMMo: How to Recover?



47

CMU SCS

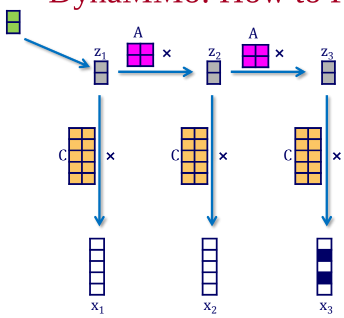
DynaMMo: How to Recover?



48

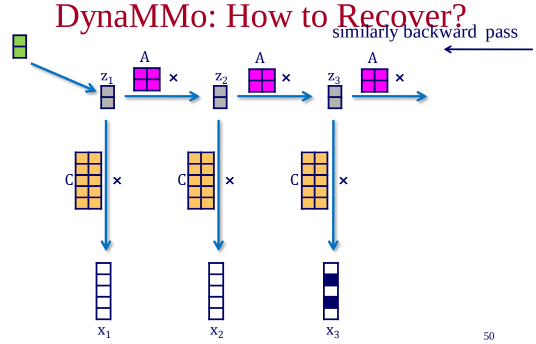
CMU SCS

DynaMMo: How to Recover?



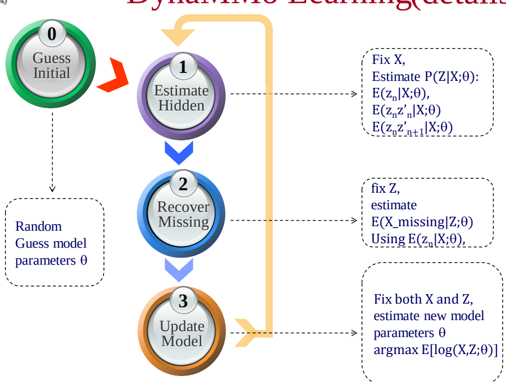
CMU SCS

DynaMMo: How to Recover?



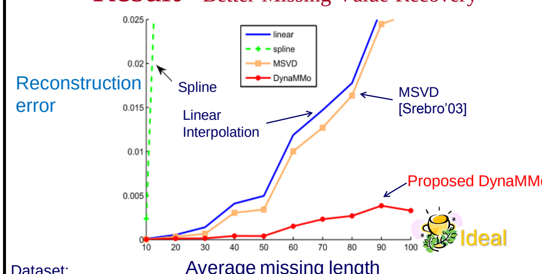
CMU SCS

DynaMMo Learning(details)



CMU SCS

Result – Better Missing Value Recovery



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

more results in [Li+2009]

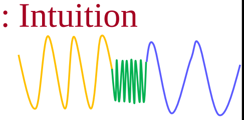
CMU SCS

Outline

- Intuition, example, and definition
- Extensions
  - Handling missing values
  - Switching LDS
  - Particle filters
- Kalman filters at work

CMU SCS

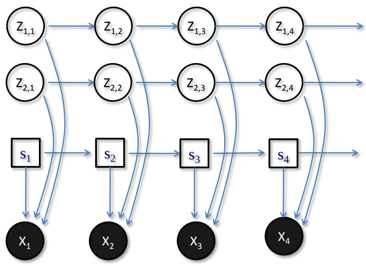
Switching LDS: Intuition



- Tracking human motion
  - Start from slow walking
  - Transition to running for a while
  - Gradually stop
- Each part corresponds to a different state & dynamics

Switching LDS

S={running, walking}



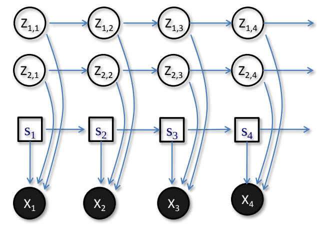
Switching LDS: example

S=1, walking

$A=A_1$, $C=C_1$

S=2, running

$A=A_2$, $C=C_2$



Switching LDS in Penalty Kick

Before touching ball

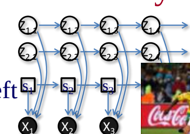
S=1, running towards left

$A=A_1$, $C=C_1$

After hitting ball

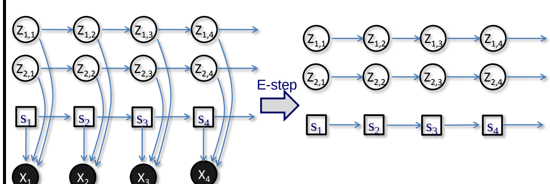
S=2, kicking towards right

$A=A_2$, $C=C_2$



From FIFA 2010

Estimating Parameters for SLDS



Approximate inference: Variational EM

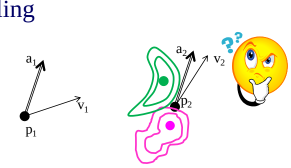
[Ghahramani&Hinton. Variational learning for switching state-space models,2000]

Outline

- Intuition, example, and definition
- Extensions
  - Handling missing values
  - Switching LDS
  - Particle filters
- Kalman filters at work

Particle Filters

- What if non-Gaussian noise?
- Inference using Markov chain Monte Carlo sampling



[Gordon et al, Nonlinear/non-gaussian bayesian state estimation, 1993]

CMU SCS

Outline

- Intuition, example, and definition
- Extensions
- Kalman filters at work
  - Segmentation & Compression
  - Parallel learning on Multi-core
  - Motion Stitching

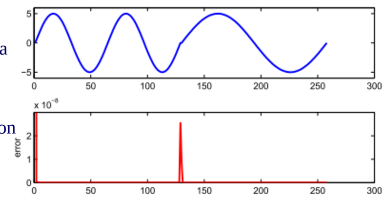
CMU SCS

How to Segment

- Segment by threshold on reconstruction error

original data

reconstruction error



[Lei Li et al DynaMMo. KDD 2009]

CMU SCS

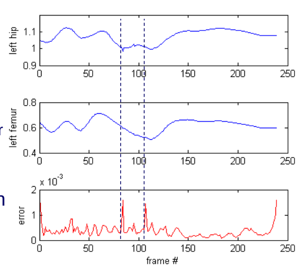
Results – Segmentation

- Find the *transition* during "running" to "stop".

left hip

left femur

reconstruction error



CMU SCS

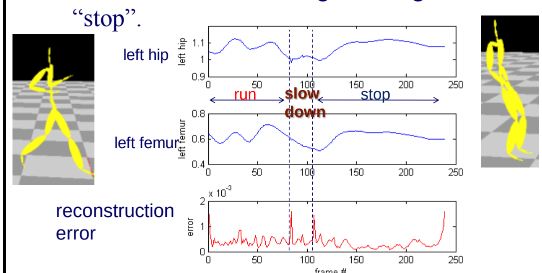
Results – Segmentation

- Find the *transition* during "running" to "stop".

left hip

left femur

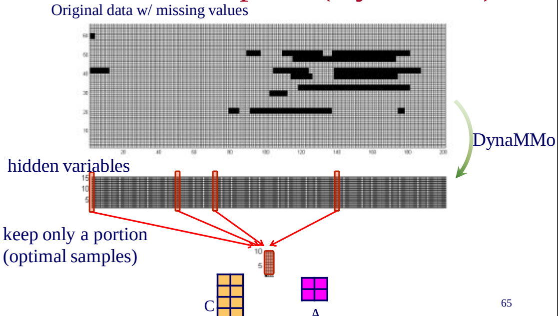
reconstruction error



CMU SCS

How to Compress (DynaMMo)

Original data w/ missing values



CMU SCS

Outline

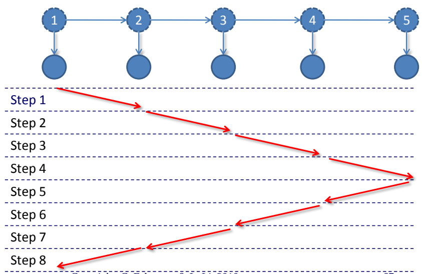
- Intuition, example, and definition
- Extensions
- Kalman filters at work
  - Segmentation & Compression
  - Parallel learning on Multi-core
  - Motion Stitching

CMU SCS

Challenge illustration Expectation-Maximization Alg.

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

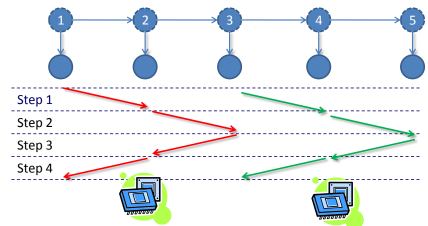
EM can only use single CPU due to data dependency



CMU SCS

Parallel learning by Cut-And-Stitch Method

Goal: with 2 CPUs

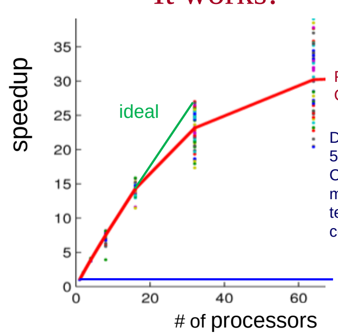


[Li et al 2008b]

[Li et al Cut-And-Stitch. KDD 2008]

CMU SCS

It works!



Dataset:
58 motion sequences
CMU Mocap #16
mocap.cs.cmu.edu,
tested on NCSA super computer,

CMU SCS

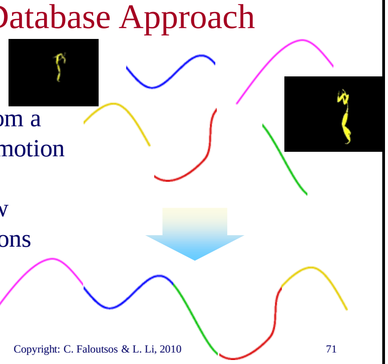
Outline

- Intuition, example, and definition
- Extensions
- Kalman filters at work
  - Segmentation & Compression
  - Parallel learning on Multi-core
  - Motion Stitching

CMU SCS

Motion Stitching A Database Approach

- Select *best stitchable* segments from a set of basic motion pieces and generate new natural motions

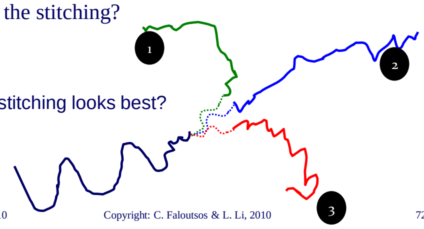


CMU SCS

Problem Definition

- Given two motion-capture sequences that are to be stitched together, how can we assess the goodness of the stitching?

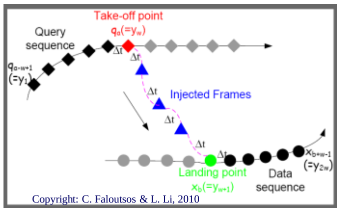
Which stitching looks best?



CMU SCS

Proposed Method: Laziness Score [Li+2008a]

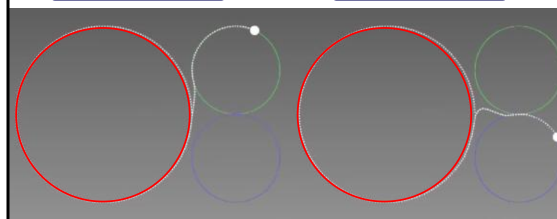
- Conjecture: *less human effort → more natural*
- Proposed: use Kalman filters to estimate position, velocity, acceleration → Compute effort/ energy



CMU SCS

Which continues to? Green or Blue?

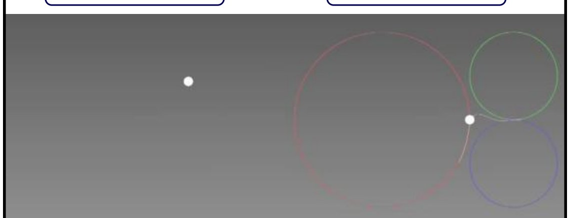
straight moving | U-Turn



CMU SCS

Result – Laziness-score prefers straightforward moving

straight moving | U-Turn



[Li et al Laziness Score. Eurographics 2008]

CMU SCS

Conclusion

- Intuition, example, and definition
  - Original kalman filter (known parameters)
    - Kalman filtering
    - Kalman smoothing
  - Kalman filters with parameter estimation (EM)
- Extensions
  - Handling missing values
  - Switching linear dynamical
  - Particle filters (MCMC sampling)
- Kalman filters at work
  - Segmentation & compression
  - Parallel learning
  - Motion stitching

CMU SCS

References

- Zoubin Ghahramani and Geoffrey E. Hinton. Parameter estimation for linear dynamical systems. 1996
- Zoubin Ghahramani and Geoffrey E. Hinton. Variational learning for switching state-space models. Neural Computation, 12(4):831–864, 2000.
- N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. IEE Proceedings F Radar and Signal Processing, 140(2):107–113, 1993.
- R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the em algorithm. Journal of Time Series Analysis, 3:253–264, 1982.
- R. E. Kalman. A new approach to linear filtering and prediction problems. Transactions of the ASME ASC Journal of Basic Engineering, (82 (Series D)):35–45, 1960.

CMU SCS

References (cont')

- H. Rauch, F Tung, and C T Striebel. Maximum likelihood estimates of linear dynamic systems. AIAA Journal, pages 1445–1450, 1965.
- Lei Li, Wenjie Fu, Fan Guo, Todd C Mowry, and Christos Faloutsos. Cut-and-stitch: efficient parallel learning of linear dynamical systems on smps. In Proceedings of the 14th ACM SIGKDD, pages 471–479, 2008.
- Lei Li, James McCann, Christos Faloutsos, and Nancy Pollard. Laziness is a virtue: Motion stitching using effort minimization. In Short Papers Proceedings of EUROGRAPHICS, 2008.
- Lei Li, James McCann, Nancy S Pollard, and Christos Faloutsos. DynaMMo: mining and summarization of coevolving sequences with missing values. In Proceedings of the 15th ACM SIGKDD,, pages 507–516, 2009.

Software

- DynaMMo code (matlab) for missing value, compression & segmentation.
- Parallel learning (in C) for LDS
- http://www.cs.cmu.edu/~leili/
- http://www.cs.cmu.edu/~leili/pubs/dynammo.2.1.2.zip
- http://www.cs.cmu.edu/~leili/paralearn/paralearn.0.1.zip (running on gcc 4.2.0 above)