Learning Linear Dynamical Systems without Sequence Information

T.-K. Huang and Jeff Schneider

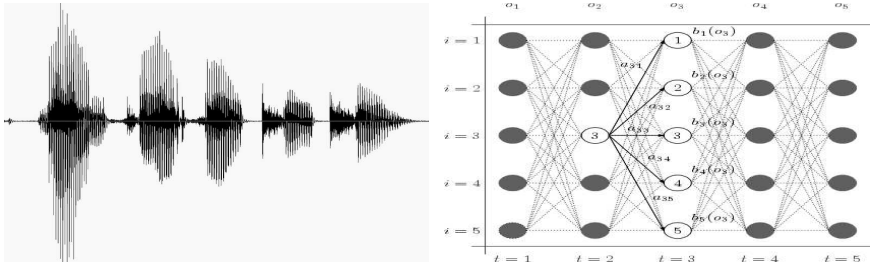
Auton Lab & Machine Learning Department & Robotics Institute
Carnegie Mellon University

Learning Dynamic Models

- Useful for analyzing time-evolving data

Hidden Markov Models

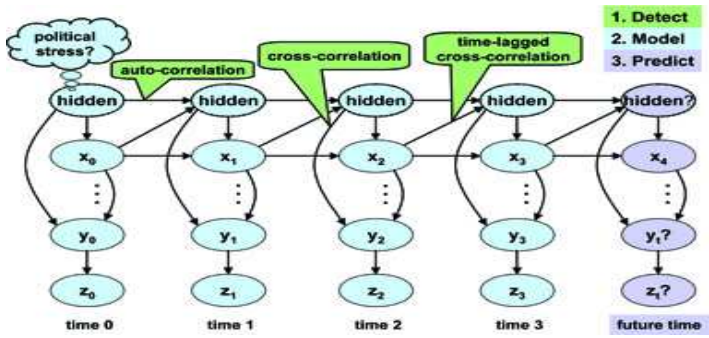
ex. Speech Recognition



[Source: Wikimedia Commons]

Dynamic Bayesian Networks

ex. Protein Interaction



[Source: SISL ARLUT]

System Identification

ex. Automatic Control

[Bagnell & Schneider, 2001]

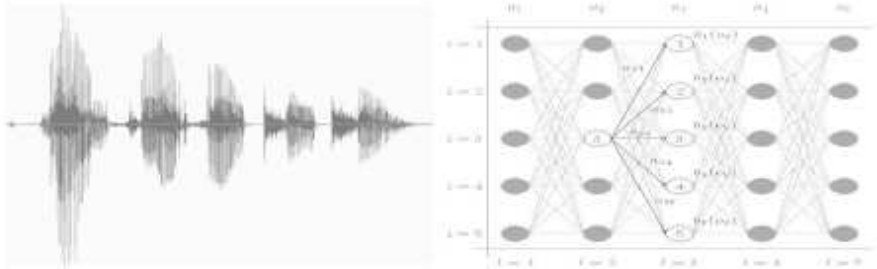
[Source: UAV ETHZ]

Learning Dynamic Models

- Useful for analyzing time-evolving data

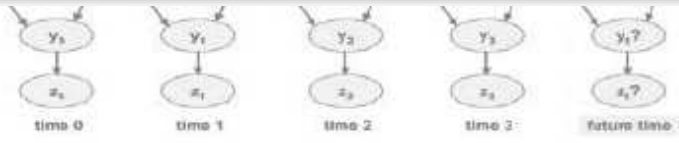
Hidden Markov Models

ex. Speech Recognition



- Key Assumption: SEQUENCED observations
- What if observations are NOT SEQUENCED?

ex. Protein Interaction



[Source: SISL ARLUT]

System Identification

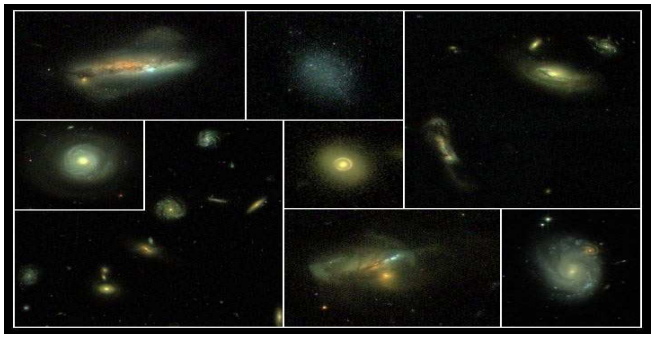
ex. Automatic Control

[Bagnell & Schneider, 2001]

[Source: UAV ETHZ]

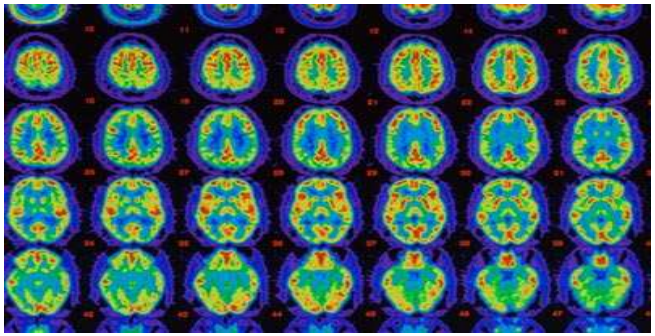
When are observations not sequenced?

Galaxy evolution (many snapshots, no ordering)



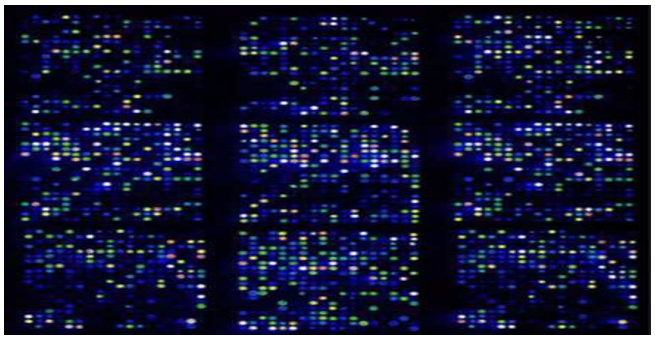
[Source: STAGES]

Slow-developing diseases, ex. Alzheimer's and Parkinson's



[Source: Getty Images]

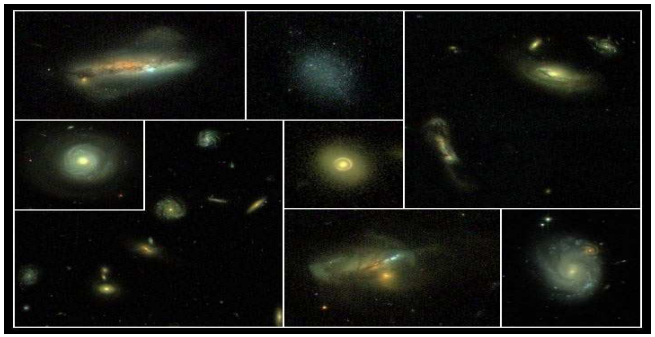
Destructive measurement for biological processes



[Source: Bryan Neff Lab, UWO]

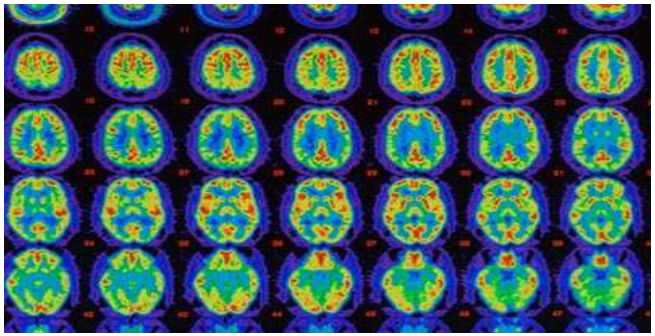
When are observations not sequenced?

Galaxy evolution (many snapshots, no ordering)



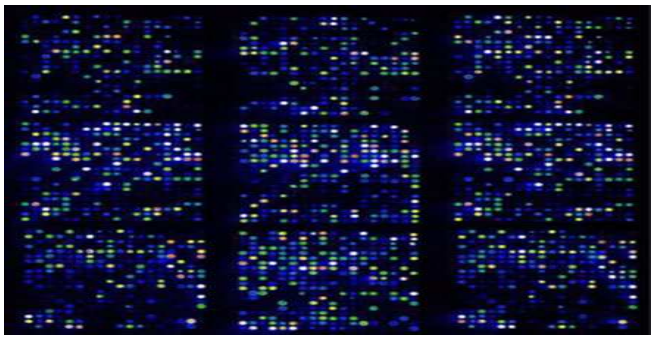
[Source: STAGES]

Slow-developing diseases, ex. Alzheimer's and Parkinson's



[Source: Getty Images]

Destructive measurement for biological processes



[Source: Bryan Neff Lab, UWO]

How can we learn dynamic models?

Formal Definition

- Consider *linear*, *discrete-time*, *continuous-state*, and *fully observable* systems:

$$\mathbf{x}^t \leftarrow A\mathbf{x}^{t-1} + \epsilon, \ \epsilon \sim N(\mathbf{0}, \sigma^2 I)$$

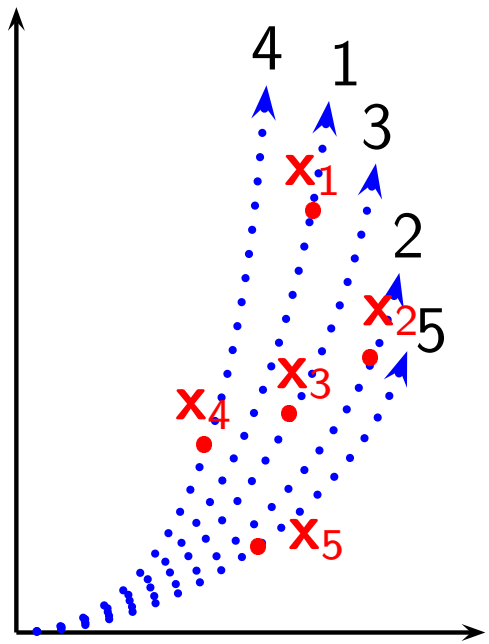
Formal Definition

- Consider *linear*, *discrete-time*, *continuous-state*, and *fully observable* systems:

$$\mathbf{x}^t \leftarrow A\mathbf{x}^{t-1} + \epsilon, \ \epsilon \sim N(\mathbf{0}, \sigma^2 I)$$

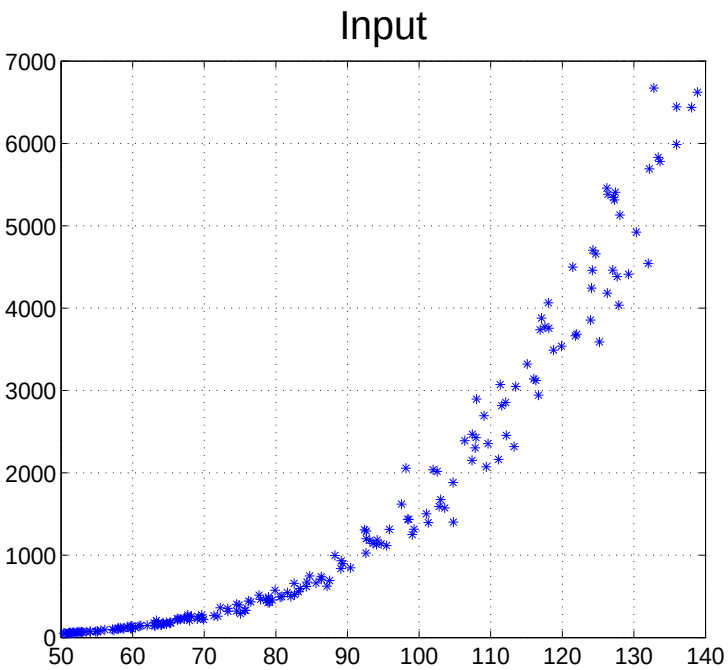
- Exactly one observation from each trajectory

for $i = 1$ **to** N **do**
  Randomly pick $T_i \in \{1, \ldots, T_{\max}\}$
  for $t = 1$ **to** T_i **do**
    $\mathbf{x}^t \leftarrow A\mathbf{x}^{t-1} + \epsilon$
  end for
  Set $\mathbf{x}_i = \mathbf{x}^{T_i}$
end for
Output: A sample $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$

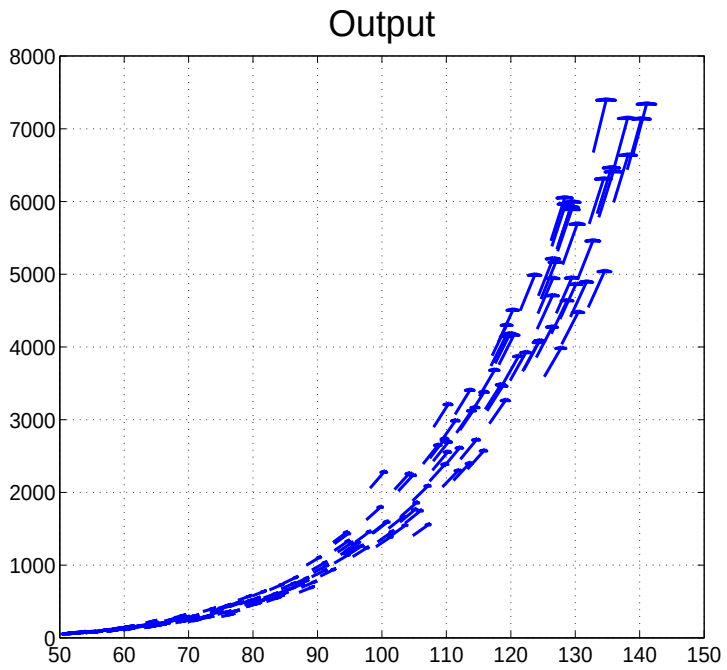


Goal

- Estimate A (and σ^2) from non-sequenced sample



(a) Sample points

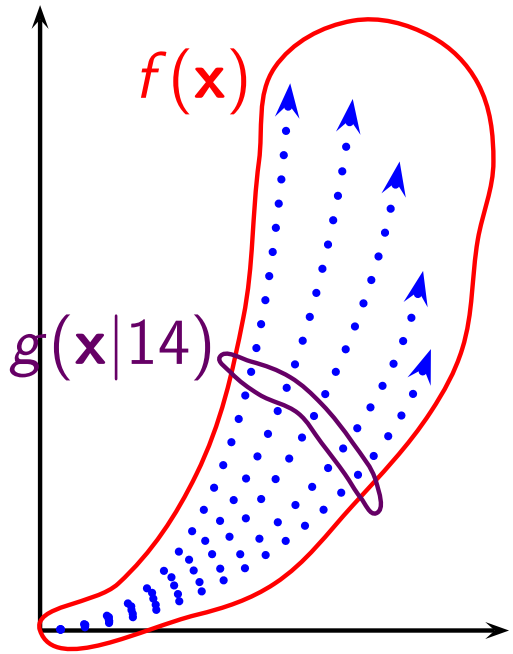


(b) Estimated gradients

Estimated gradient: $\hat{A}\mathbf{x}_i - \mathbf{x}_i$

True Likelihood: Notation

- Let $\mathbf{x}$ be a point in the state space of the system, and $t(\mathbf{x})$ be its time index
- Let $f(\mathbf{x})$ be the state space density induced by the system
- Let $g(\mathbf{x}|j)$ be the state space density at time j



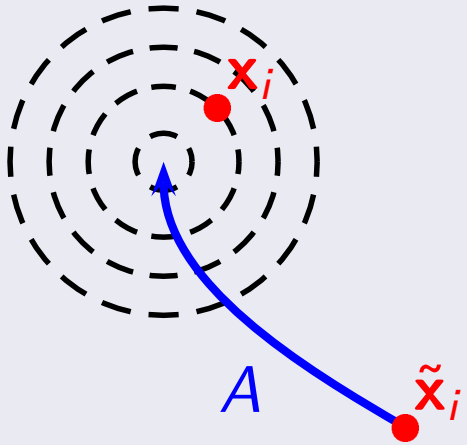
True Likelihood

- If the true predecessor $\tilde{\mathbf{x}}_i$ of $\mathbf{x}_i$ is known, we may write the joint likelihood:

$$I(\mathbf{x}_1, \ldots, \mathbf{x}_N | \tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_N) = \prod_{i=1}^{N} \frac{\exp\left(-\frac{\|\mathbf{x}_i - A\tilde{\mathbf{x}}_i\|^2}{2\sigma^2}\right)}{(2\pi\sigma^2)^{\frac{n}{2}}}$$

Recall the system equation:

$$\mathbf{x}_i \leftarrow A\tilde{\mathbf{x}}_i + \epsilon, \ \ \epsilon \sim N(\mathbf{0}, \sigma^2 I)$$



True Likelihood (Cont.)

- If the true predecessor $\tilde{\mathbf{x}}_i$ of $\mathbf{x}_i$ is known, we may write the joint likelihood:

$$l(\mathbf{x}_1, \ldots, \mathbf{x}_N | \tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_N) = \prod_{i=1}^{N} \frac{\exp\left(-\frac{\|\mathbf{x}_i - A\tilde{\mathbf{x}}_i\|^2}{2\sigma^2}\right)}{(2\pi\sigma^2)^{\frac{n}{2}}}$$

- But $\tilde{\mathbf{x}}_i$ is unknown, so integrate it out w.r.t the density at time $t(\mathbf{x}_i) - 1$:

$$l(\mathbf{x}_1, \ldots, \mathbf{x}_N | t(\mathbf{x}_1), \ldots, t(\mathbf{x}_N))$$

$$= \prod_{\substack{i=1, \\ t(\mathbf{x}_i)>0}}^{N} \left(\int \frac{\exp\left(-\frac{\|\mathbf{x}_i - A\mathbf{x}\|^2}{2\sigma^2}\right)}{(2\pi\sigma^2)^{\frac{n}{2}}} g(\mathbf{x} | t(\mathbf{x}_i) - 1) d\mathbf{x} \right)$$

Approximate Likelihood Approaches

- Maximizing $l(\mathbf{x}_1, \ldots, \mathbf{x}_N | t(\mathbf{x}_1), \ldots, t(\mathbf{x}_N))$ on A and σ^2 is hard
  - $t(\mathbf{x}_i)$ is missing
  - $g(\mathbf{x}|t(\mathbf{x}_i) - 1)$ contains high-order terms in A
- Instead, maximize *approximate* likelihood
  - An Unordered Model (UM)
  - A Partial-ordered Model (PM)

Unordered Model (UM)

- Assume $t(\mathbf{x}_i)$ uniformly sampled from $\{1, \ldots, T_{\max}\}$
- Marginalize over the missing $t(\mathbf{x}_i)$:

$$I(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{\substack{i=1, \\ t(\mathbf{x}_i)>0}}^{N} \sum_{t(\mathbf{x}_i)=1}^{T_{\max}} \left(\int \frac{\exp(-\frac{\|\mathbf{x}_i - A\mathbf{x}\|^2}{2\sigma^2})}{(2\pi\sigma^2)^{\frac{n}{2}}} \frac{g(\mathbf{x}|t(\mathbf{x}_i)-1)}{T_{\max}} d\mathbf{x} \right)$$

$$\approx \prod_{\substack{i=1, \\ t(\mathbf{x}_i)>0}}^{N} \left(\int \frac{\exp(-\frac{\|\mathbf{x}_i - A\mathbf{x}\|^2}{2\sigma^2})}{(2\pi\sigma^2)^{\frac{n}{2}}} f(\mathbf{x}) d\mathbf{x} \right)$$

Unordered Model (UM)

- Assume $t(\mathbf{x}_i)$ uniformly sampled from $\{1, \ldots, T_{\max}\}$
- Marginalize over the missing $t(\mathbf{x}_i)$:

$$
\begin{aligned}
I(\mathbf{x}_1, \ldots, \mathbf{x}_N) &= \prod_{\substack{i=1, \\ t(\mathbf{x}_i)>0}}^{N} \sum_{t(\mathbf{x}_i)=1}^{T_{\max}} \left(\int \frac{\exp(-\frac{\|\mathbf{x}_i - A\mathbf{x}\|^2}{2\sigma^2})}{(2\pi\sigma^2)^{\frac{n}{2}}} \frac{g(\mathbf{x}|t(\mathbf{x}_i)-1)}{T_{\max}} d\mathbf{x} \right) \\
&\approx \prod_{\substack{i=1, \\ t(\mathbf{x}_i)>0}}^{N} \left(\int \frac{\exp(-\frac{\|\mathbf{x}_i - A\mathbf{x}\|^2}{2\sigma^2})}{(2\pi\sigma^2)^{\frac{n}{2}}} f(\mathbf{x}) d\mathbf{x} \right)
\end{aligned}
$$

Approximate $f(\mathbf{x})$ with the empirical density

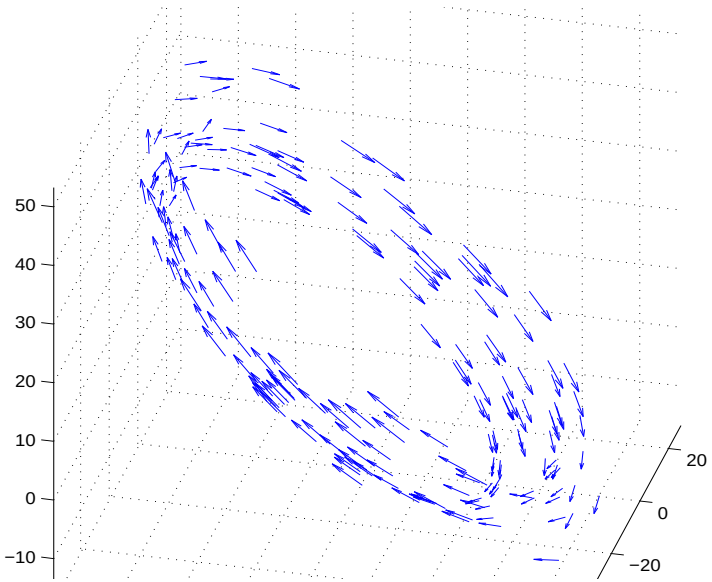
$$
\hat{I}_1(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{i=1}^{N} \left(\sum_{j \neq i} \frac{\exp(-\frac{\|\mathbf{x}_i - A\mathbf{x}_j\|^2}{2\sigma^2})}{(N-1)(2\pi\sigma^2)^{\frac{n}{2}}} \right) \qquad \text{(UM)}
$$

UM: Estimation

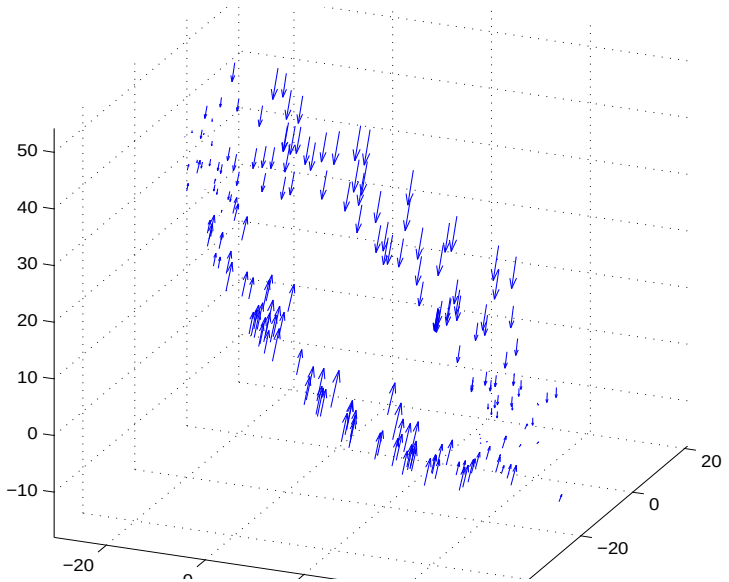
- Expectation Maximization
- Introduce latent predecessor variable $Z \in \{0,1\}^{N \times N}$
  Z_{ij} indicates whether $\mathbf{x}_j$ comes from $\mathbf{x}_i$
- E-step:
  Similar to that of Gaussian Mixture Models
- M-step (for A and σ^2):
  Least square linear regression

UM: Pros and Cons

- Pros: reasonable approximation, simple estimation
- Cons: marginalizing over $t(\mathbf{x}_i)$ obscures the underlying order in time $\Rightarrow$ degenerate estimates, lacking globally evolving dynamics



True gradients



Estimated gradients by UM

Partial-ordered Model (PM)

- Idea: instead of marginalizing over $t(\mathbf{x}_i)$, try to estimate it
- Estimating $t(\mathbf{x}_i)$ directly is hard, may involve:
  - ◇ Finding a total order of sample points
  - ◇ Maximization over permutations
- Solution: break total order into pairwise relationships, then seek a partial order

PM: Approximate likelihood

- In true likelihood, approximate $g(\mathbf{x}|\cdot)$ by data:

$$
\begin{aligned}
& l(\mathbf{x}_1, \ldots, \mathbf{x}_N | t(\mathbf{x}_1), \ldots, t(\mathbf{x}_N)) \\
= & \prod_{i=1, t(\mathbf{x}_i)>0}^{N} \left(\int \frac{\exp(-\frac{\|\mathbf{x}_i - A\mathbf{x}\|^2}{2\sigma^2})}{(2\pi\sigma^2)^{\frac{n}{2}}} g(\mathbf{x}|t(\mathbf{x}_i) - 1) d\mathbf{x} \right) \\
\approx & \prod_{\substack{i=1, \\ i \notin S}}^{N} \sum_{j=1}^{N} \left(\frac{\exp(-\frac{\|\mathbf{x}_i - A\mathbf{x}_j\|^2}{2\sigma^2})}{(2\pi\sigma^2)^{\frac{n}{2}}} \hat{g}(\mathbf{x}_j|t(\mathbf{x}_i) - 1) \right)
\end{aligned}
$$

S: the set of start states

PM: Approximate likelihood

- In true likelihood, approximate $g(\mathbf{x}|\cdot)$ by data:

$$
\begin{aligned}
& I(\mathbf{x}_1, \ldots, \mathbf{x}_N | t(\mathbf{x}_1), \ldots, t(\mathbf{x}_N)) \\
= & \prod_{i=1, t(\mathbf{x}_i)>0}^{N} \left(\int \frac{\exp(-\frac{\|\mathbf{x}_i - A\mathbf{x}\|^2}{2\sigma^2})}{(2\pi\sigma^2)^{\frac{n}{2}}} g(\mathbf{x}|t(\mathbf{x}_i) - 1) d\mathbf{x} \right) \\
\approx & \prod_{\substack{i=1, \\ i \notin S}}^{N} \sum_{j=1}^{N} \left(\frac{\exp(-\frac{\|\mathbf{x}_i - A\mathbf{x}_j\|^2}{2\sigma^2})}{(2\pi\sigma^2)^{\frac{n}{2}}} \hat{g}(\mathbf{x}_j|t(\mathbf{x}_i) - 1) \right)
\end{aligned}
$$

S: the set of start states

Rename $\hat{g}(\mathbf{x}_j|t(\mathbf{x}_i) - 1)$ as ω_{ij} (pairwise params)

$$
\hat{I}_2(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{\substack{i=1, \\ i \notin S}}^{N} \sum_{j=1}^{N} \left(\frac{\exp(-\frac{\|\mathbf{x}_i - A\mathbf{x}_j\|^2}{2\sigma^2})}{(2\pi\sigma^2)^{\frac{n}{2}}} \omega_{ij} \right) \qquad \text{(PM)}
$$

PM: Constraints

- $\omega_{ij} \equiv \hat{g}(\mathbf{x}_j | t(\mathbf{x}_i) - 1) \approx$ Prob. that $\mathbf{x}_j$ precedes $\mathbf{x}_i$

  Matrix $\omega \geq 0$, each row sums to 1 or 0 (C1)

- We want time direction to be consistent:

  As an adjacency matrix,
  ω represents a directed acyclic graph

- Modified constraints

  $\omega \in \{0, 1\}^{N \times N}$ represents a directed TREE (C2)

  Efficient computation

PM Estimation: Alternating Optimization

- Constrained maximization:

$$\max_{\substack{A,\sigma^2,\omega,\\ r\in\{1,\ldots,N\}}} \sum_{\substack{i=1,\\ i\neq r}}^{N} \log \sum_{j=1}^{N} \left(\frac{\exp\left(-\frac{\|\mathbf{x}_i - A\mathbf{x}_j\|^2}{2\sigma^2}\right)}{(2\pi\sigma^2)^{\frac{n}{2}}} \omega_{ij} \right)$$

$$\text{s.t. } \omega_{ij} \in \{0,1\}, \ \sum_{j=1}^{N} \omega_{ij} = 1, \ i \neq r, \ \sum_{j=1}^{N} \omega_{rj} = 0,$$

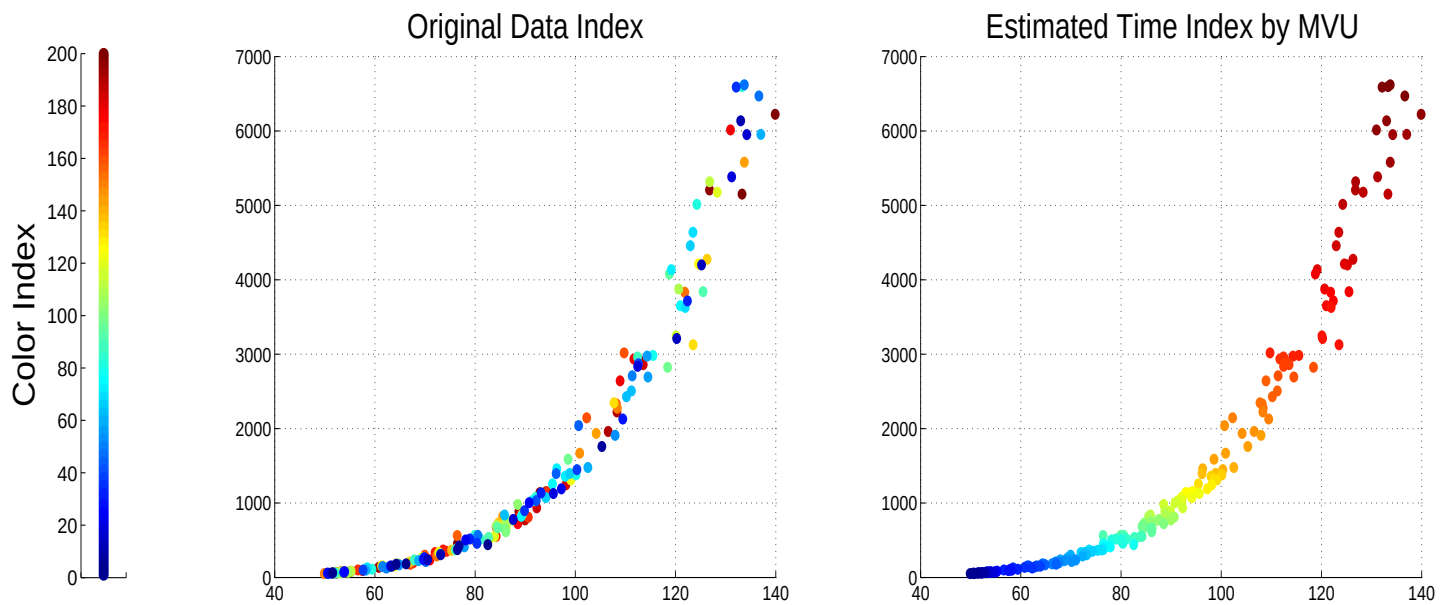
$$\omega \text{ forms a tree with root } \mathbf{x}_r$$

- Maximize ω under fixed A and σ^2:
  Maximum spanning tree on directed graph: $O(N^2)$
- Maximize A and σ^2 under fixed ω:
  Least-square linear regression

Initialization for UM and PM

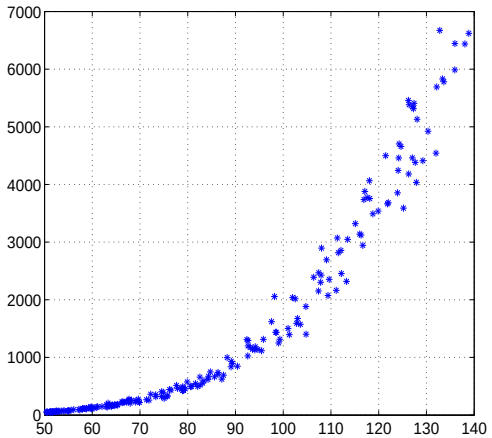
- Random initialization
- Manifold Learning/Dimensionality Reduction
  (1) Project data points into a one-dimensional space
  (2) Sort data points by their 1-D projections
  (3) Learn a linear dynamic model based on sorted data
  (4) Initialize UM and PM with the learned model parameters

Ordering by Maximum Variance Unfolding [Weinberger et al., 2004]:

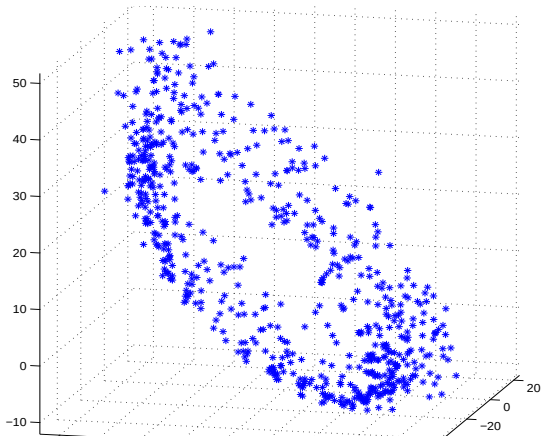


Experiment Setting: Data sets

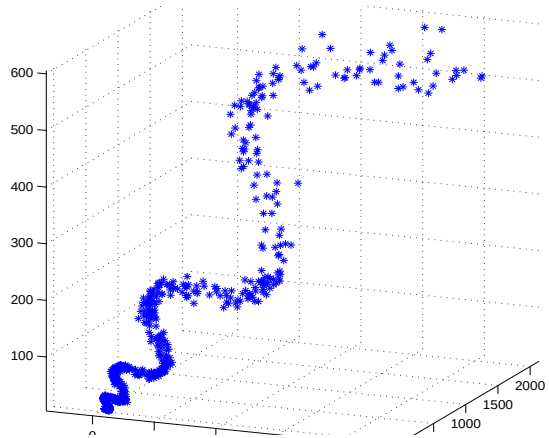
- 2D: 40 random samples, 200 points each, $\sigma = 0.2$.
- 3D-1 and 3D-2:
  - Small-sized experiments: 40 random samples, 200 points each, $\sigma = 0.2, 0.4, 0.6, 0.8$.
  - Large-sized experiments: 20 random samples, 2,000 points each, $\sigma = 0.2, 0.4, 0.6, 0.8$.



2D



3D-1



3D-2

Experiment Setting: Methods

Compare six methods:

- Maximum Variance Unfolding (MVU), only on small samples
- PM, UM: multiple random initializations, choose the best estimate (largest likelihood)
- PM+MVU, UM+MVU: PM, UM initialized by MVU, only applied to small samples
- Rand: Random guess of A and σ^2

Evaluation Criteria

- Rate-adjusted *matrix error*

$$\mathrm{ME}(A, \hat{A}) \equiv \min_{t \in Q} \|A - \hat{A}^t\|_F$$

$$Q = \{\pm 1, \pm 2, \ldots, \pm 10, \pm 1/2, \pm 1/3, \ldots, \pm 1/10\}$$

Smaller is better

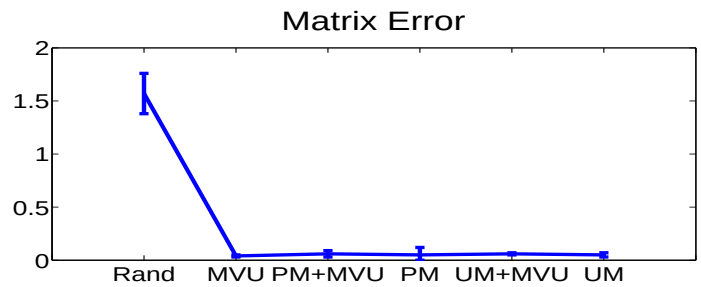
- Normalized gradient *cosine score*

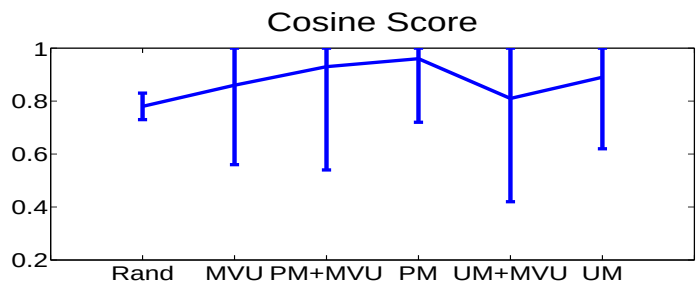
$$\mathrm{CS}(A, \hat{A}) \equiv \frac{1}{N} \left| \sum_{i=1}^{N} \frac{(A\mathbf{x}_i - \mathbf{x}_i)'(\hat{A}\mathbf{x}_i - \mathbf{x}_i)}{\|A\mathbf{x}_i - \mathbf{x}_i\| \|\hat{A}\mathbf{x}_i - \mathbf{x}_i\|} \right| \in [0, 1]$$

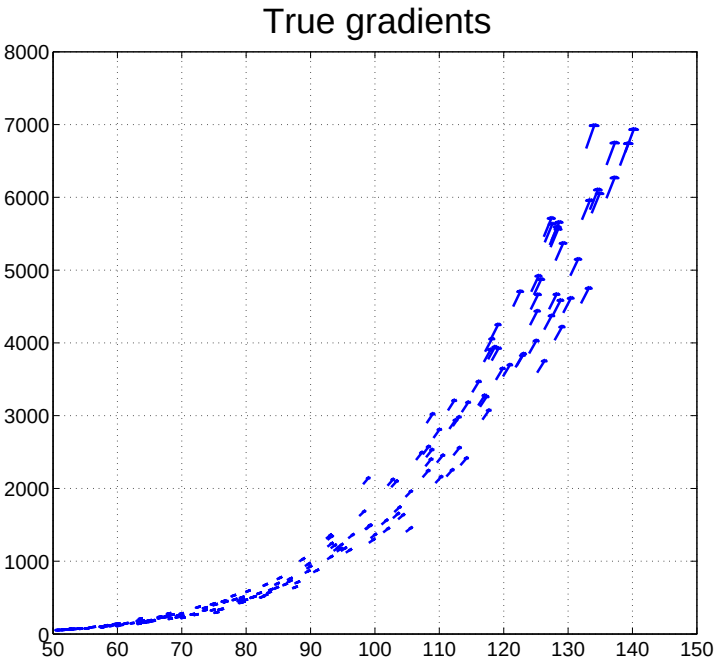
Larger is better

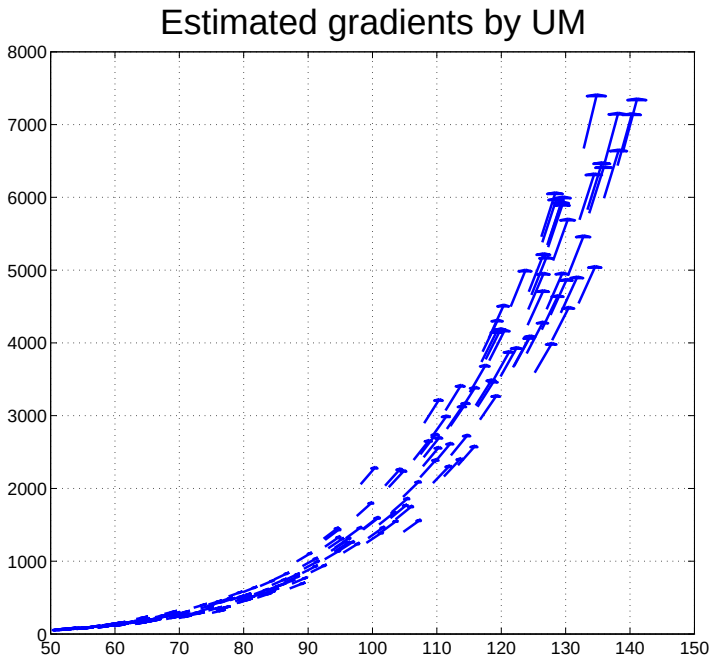
Results: 2D

Table: Results on 40 samples with standard deviations, $\sigma = 0.2$

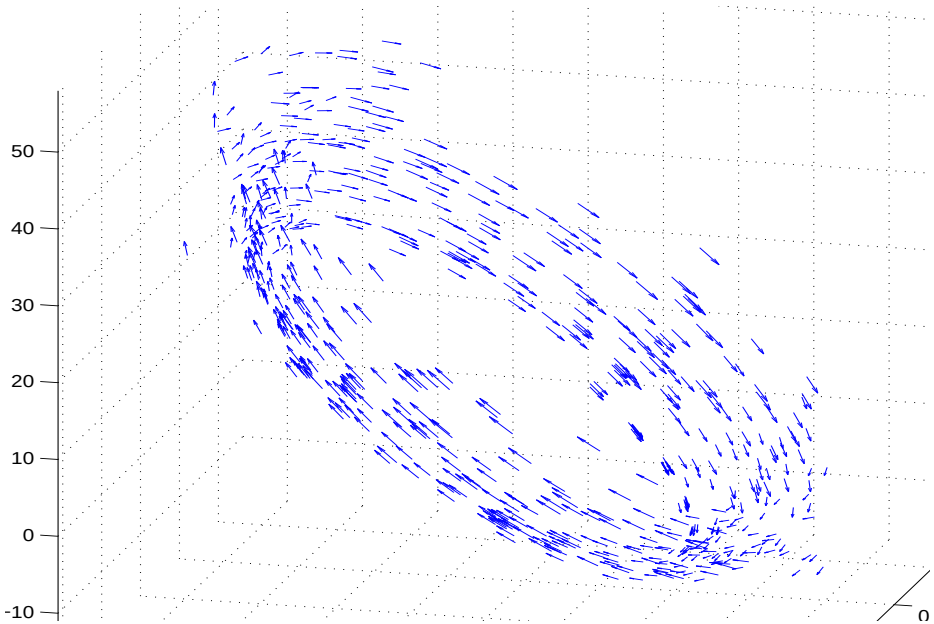




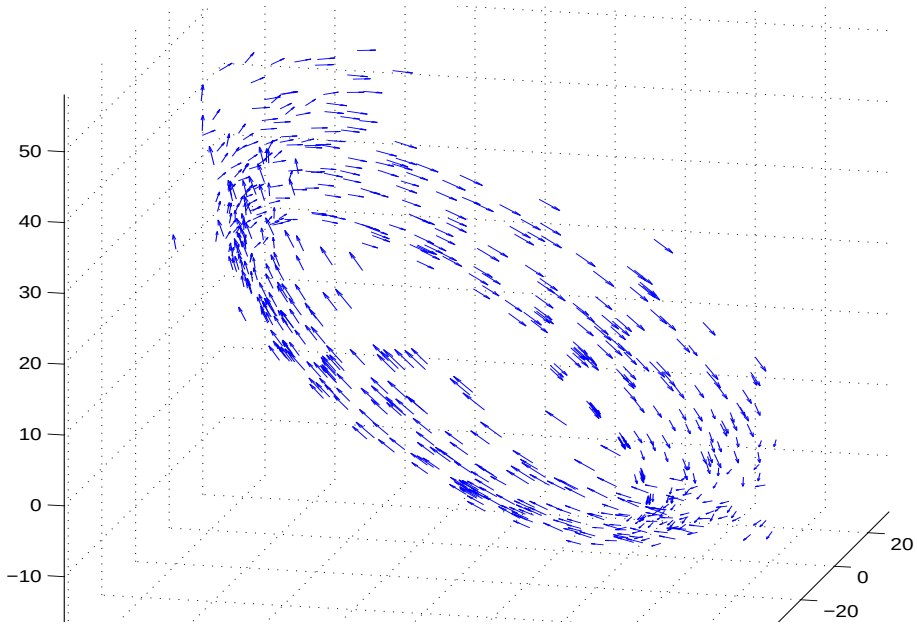




Results: 3D-1



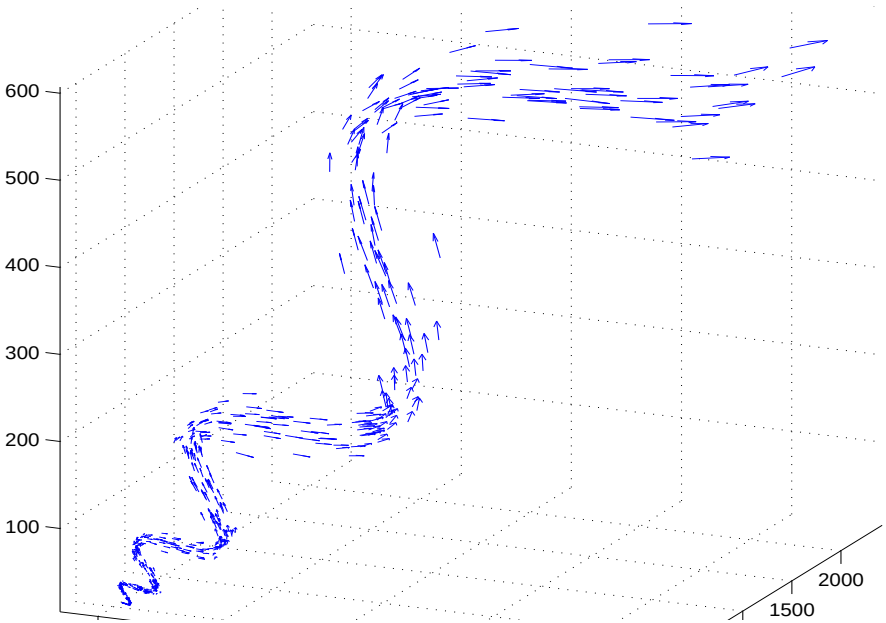
(c) True gradients



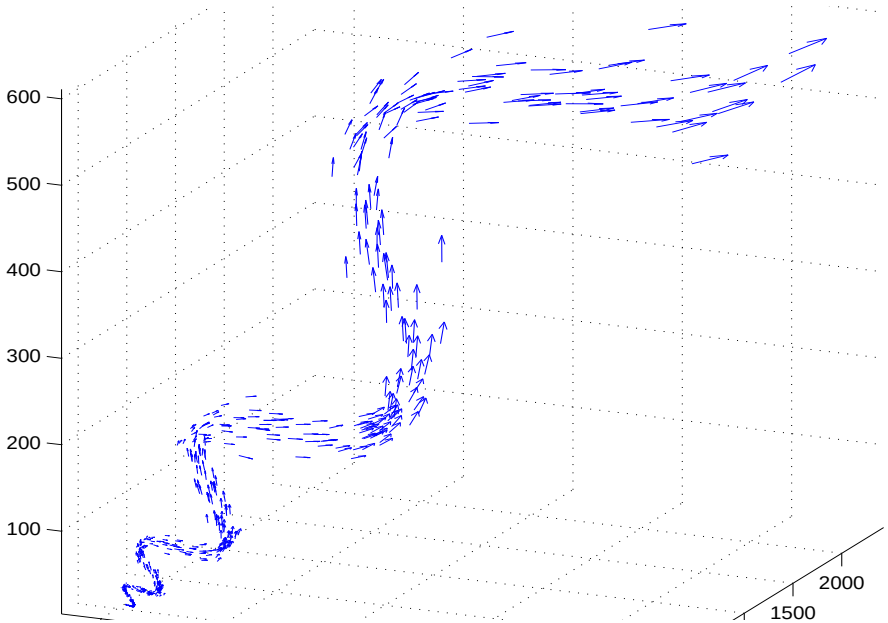
(d) Estimated gradients by PM, cosine score = 0.9921

3D-1, $\sigma = 0.2$, 2,000-points sample

Results: 3D-2



(e) True gradients



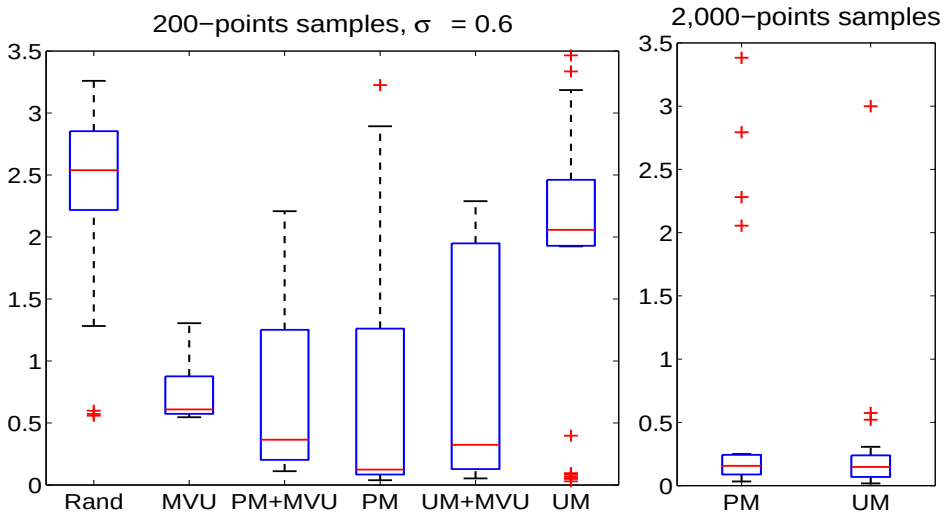
(f) Estimated gradients by PM, cosine score = 0.9574

3D-2, $\sigma = 0.2$, 2,000-points sample

3D-1: Boxplots of ME and CS, $\sigma = 0.6$

ME

better



200-points:

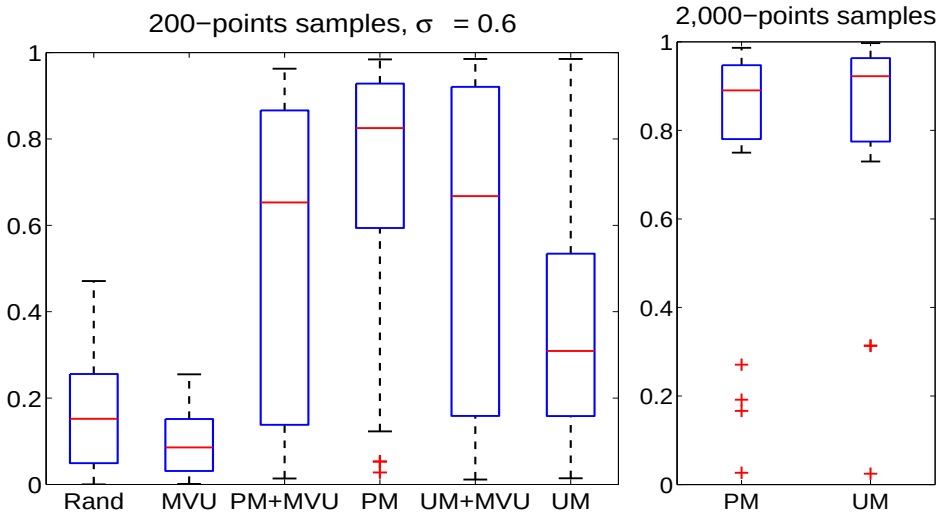
Rand,UM are worse than others

2,000-points:

PM $\approx$ UM

CS

better



200-points:

Rand,MVU,UM are worse than others

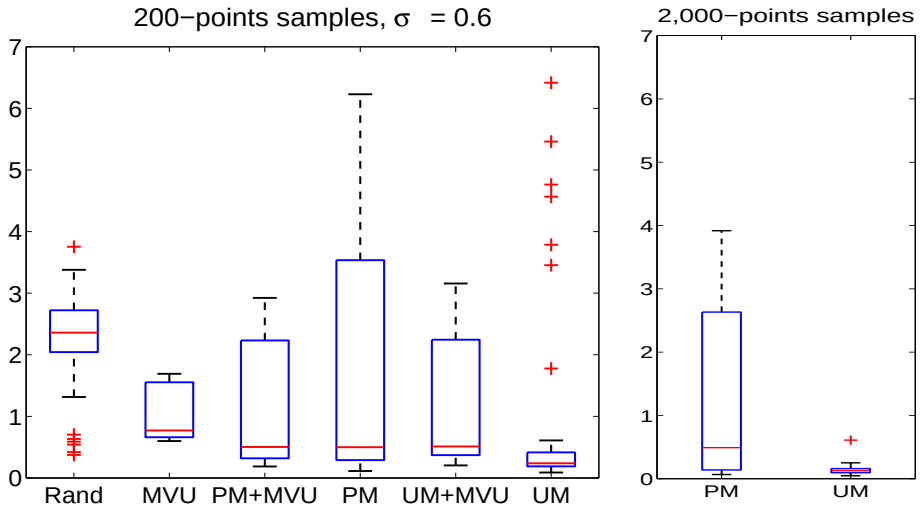
2,000-points:

PM $\approx$ UM

3D-2: Boxplots of ME and CS, $\sigma = 0.6$

ME

better



200-points:

all are good

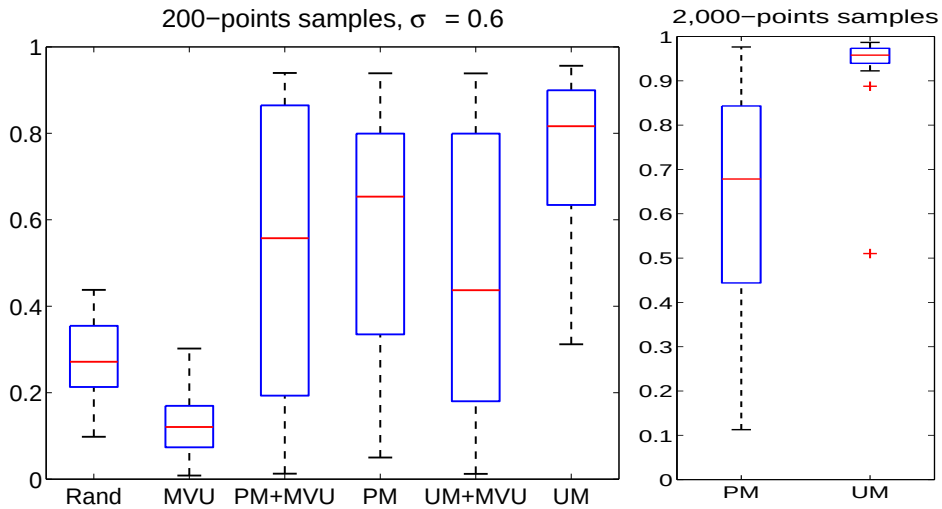
except Rand

2,000-points:

UM better than PM

CS

better



200-points:

Rand,MVU are bad,

others are good,

UM is the best

2,000-points:

UM better than PM

Findings and Issues

- MVU less useful in 3D than 2D

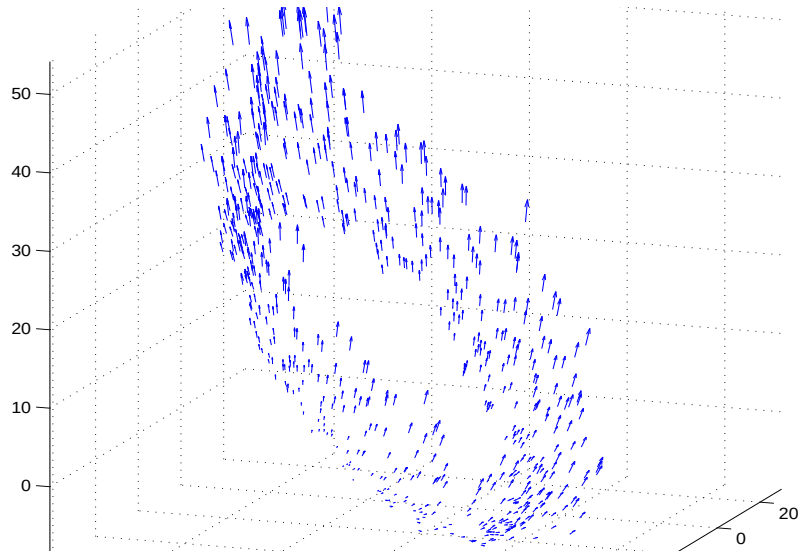
Findings and Issues

- MVU less useful in 3D than 2D
- In 3D-1, PM better than UM on small samples, but UM improves more than PM as sample size increases
  Directionality constraints introduce some bias

Findings and Issues

- MVU less useful in 3D than 2D
- In 3D-1, PM better than UM on small samples, but UM improves more than PM as sample size increases
  Directionality constraints introduce some bias

- UM can be almost as poor as Rand even if samples are large
  - What are the limitations of it?
  - Under what conditions and to what extent the problem can be solved?



UM (3D-1, $\sigma = 0.2$)

Conclusions

- Propose the problem of learning fully observable linear dynamical systems from non-sequenced data
- Propose two approximate likelihood approaches
  - Unordered Model
  - Partial-ordered Model

  Work well on synthetic data
- Many interesting future directions
  - Real data: astronomical, medical, biological
  - Theoretical properties of the problem
  - Nonlinear dynamical systems
  - Partial observability