



电子科技大学
University of Electronic Science and Technology of China



Inverse Variance-Based Network Inference and Clustering

Yao Yang



Data Mining Lab, Big Data Research Center, UESTC
Email: up9288yy@gmail.com

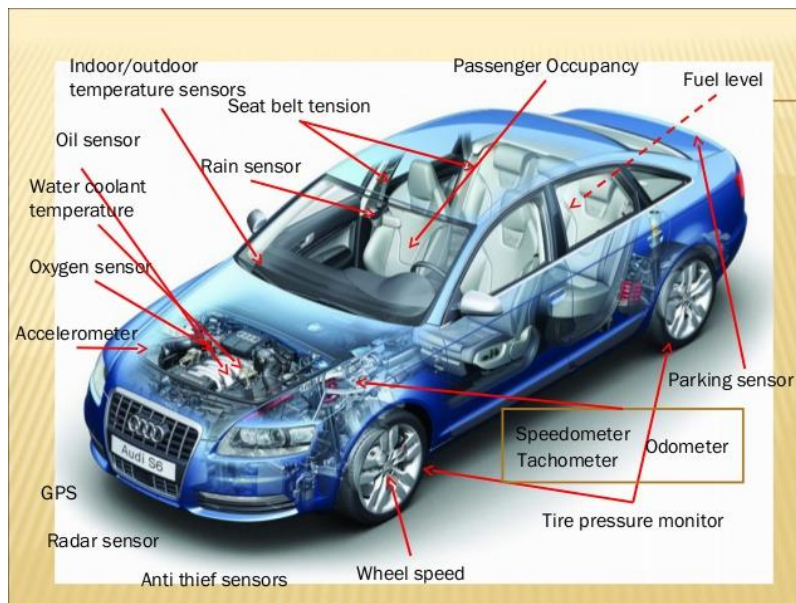
Papers

- Network Inference via the Time-Varying Graphical Lasso
- David Hallac, Stephen Boyd, Jure Leskovec, etc. KDD'17
- Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data
- David Hallac, Stephen Boyd, Jure Leskovec, etc. KDD'17



Sensors are everywhere

- In many applications, we generate large sequences of timestamped observations
 - — “Sensors” have a broad definition



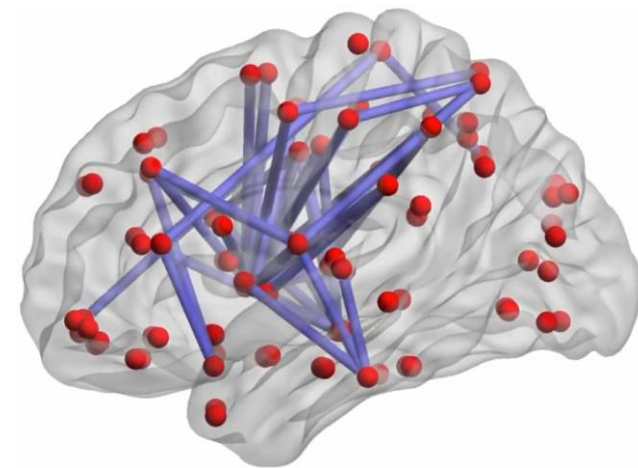
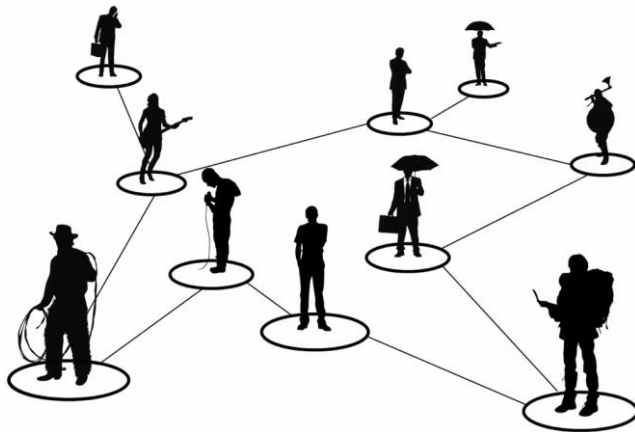


Challenges

- These data is:
 - High-dimensional
 - Unlabeled
 - High-velocity
 - Changing over time
 - Heterogeneous
- So we need a method of uncovering **interpretable structure** from the sensors in an **unsupervised** way, it must be:
 - Scalable (large amounts of raw data over long time series)
 - Robust (must apply to lots of different applications)

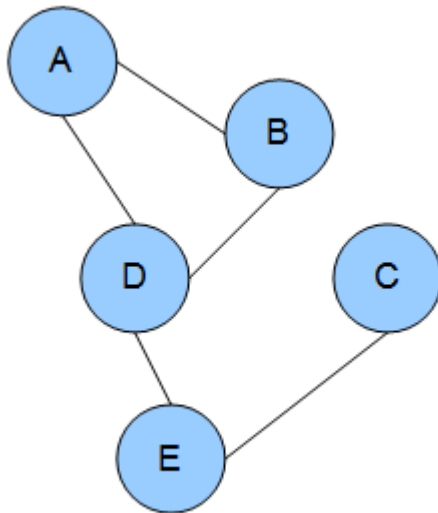
How to model

- We can often model these data as **a network of interacting entities** in these cases
- We can use these networks to discover how the **structure** of a complex system changes over time



Network encode structure

- These can be modeled as a network of interacting entities, where each entity is a node associated with a time series of data points
- An edge represents a **partial correlation, or a direct effect** (holding all other nodes constant) between two entities



Markov random fields denote conditional independencies between different entities

MRF properties:

- Pairwise Markov property
- Local Markov property
- Global Markov property

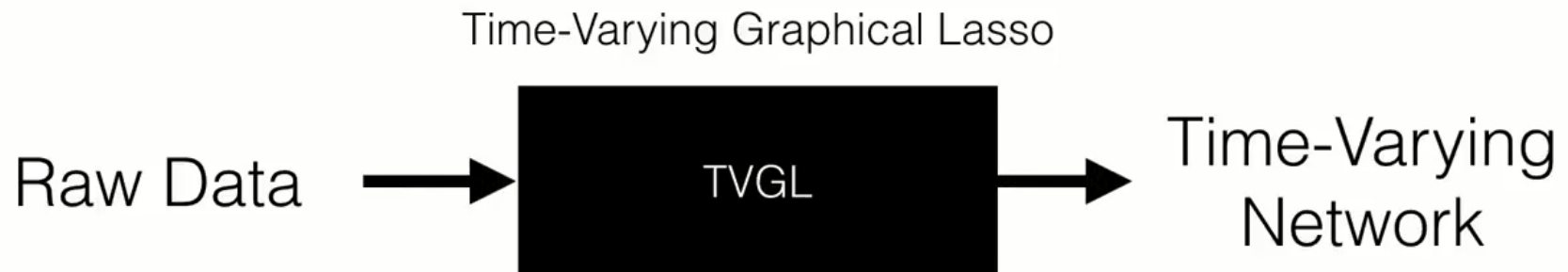
Networks can evolve over time

- These Networks do not remain constant over the course of the time series!
- Many different types of evolutions:
 - Sudden shift of the entire network structure
 - A single node rewiring all its edges
 - Smoothly varying over time
 - One or two edges changing in the whole network
- Therefore, methods must be able to uncover many types evolutionary patterns

Time-Varying Graphical Lasso

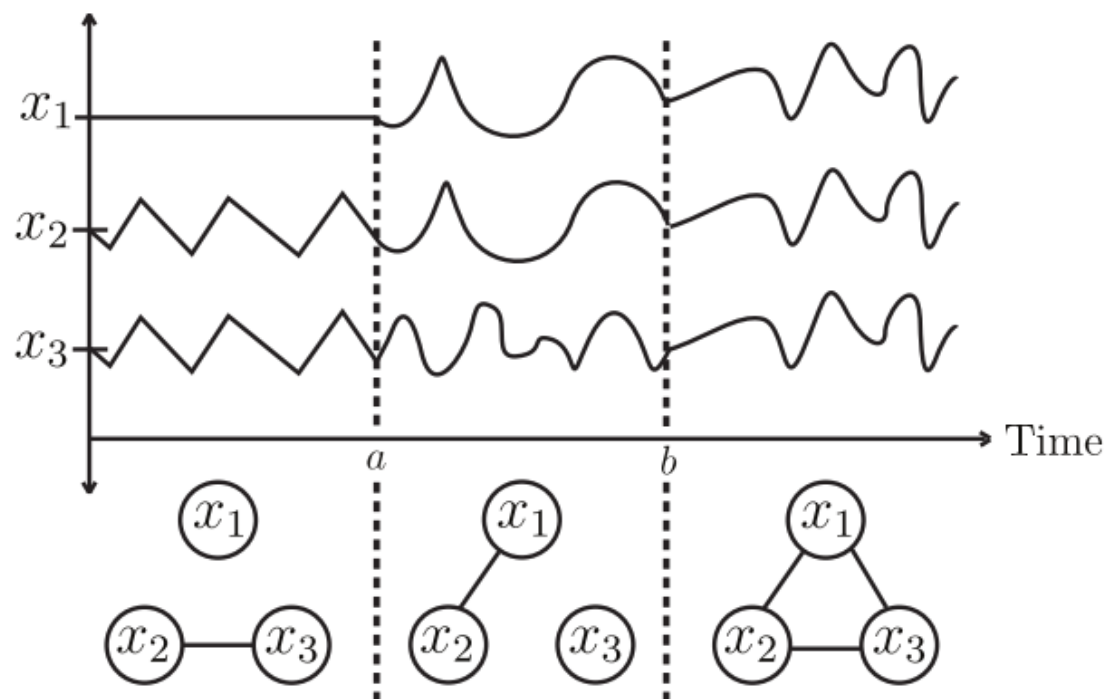
Q: So TVGL=?

A: A method takes in raw time series data and returns a **correlation network** showing how each of the sensors are related to each other and how these relationships evolve over time



Network inference from time series data

- Convert a sequence of timestamped sensor observation into a time-varying network



Tool: Inverse Covariance Matrix

- Assume observations $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We are interested in estimating inverse covariance matrix $\Theta = \boldsymbol{\Sigma}^{-1}$ to describe the Dependency Network
- Also called precision matrix, where $\Theta_{ij} = 0$ means that elements i and j are **conditionally independent**
- A **sparse** inverse covariance allows us to encode conditional independence between different variables

Inferring Static Networks

$$\text{minimize } -l(\Theta) + \lambda \|\Theta\|_{od,1}$$

Assume that $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, $i = 1, 2, 3, \dots, n$

$$L(\Theta) = \frac{1}{(2\pi)^{\frac{nD}{2}} |\Sigma|^{\frac{n}{2}}} \exp\left\{-\frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu})\right\}$$

$$\ln L(\Theta) = -\frac{1}{2} nD \ln(2\pi) + \frac{n}{2} \ln |\Sigma^{-1}| - \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu})$$

Set $\mathbf{S} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$,

$$\ln L(\Theta) = -\frac{1}{2} nD \ln(2\pi) + \frac{n}{2} \ln |\Theta| - \frac{n}{2} \text{tr}(\Theta \mathbf{S})$$

$$\propto C + n(\ln |\Theta| - \text{tr}(\Theta \mathbf{S}))$$

Because Θ is positive-definite (\mathbf{S}_{++}^p),
now our objective is:

$$\min_{\Theta \in \mathbf{S}_{++}^p} n(\text{Tr}(\Theta \mathbf{S}) - \log \det \Theta) + \lambda \|\Theta\|_{od,1}$$

Inferring Dynamic Networks - TVGL

We solve for $\Theta = (\Theta_1, \Theta_2, \dots, \Theta_T)$:

$$\min_{\Theta \in \mathcal{S}_{++}^p} \sum_{i=1}^T -l_i(\Theta_i) + \lambda \|\Theta_i\|_{od,1} + \beta \sum_{i=2}^T \psi(\Theta_i - \Theta_{i-1})$$

- Here, $l_i(\Theta_i) = n_i(\log \det \Theta_i - \text{Tr}(\mathcal{S}_i \Theta_i))$, $\beta \geq 0$,
- And $\psi(\Theta_i - \Theta_{i-1})$ is a convex penalty function, minimized at $\psi(0)$,

Simultaneously aiming to achieve three goals:

- Matching the empirical data
- Sparsity in the network
 - Provides interpretability and prevents overfitting
- Temporal consistency
 - We leverage the fact the different snapshots in time are related by imposing a penalty to limit how the network can change over time

Evolutionary dynamics

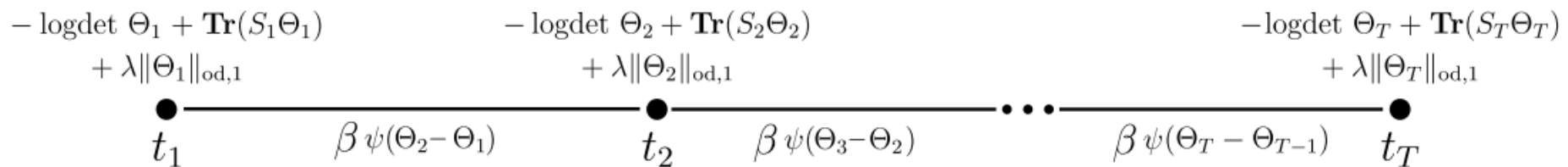
If we have an expectation about how the underlying network may change over time, we are able to encode it into ψ :

($[\mathbf{X}]_j$ refers to the j -th column of a matrix \mathbf{X})

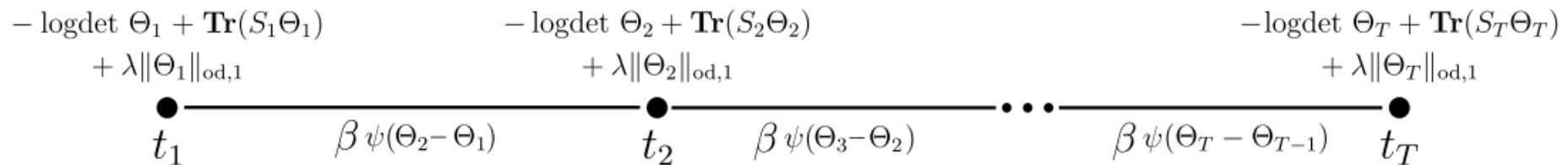
- A few edges changing at a time: $\psi(\mathbf{X}) = \sum_{i,j} |X_{i,j}|$
- Global restructuring: $\psi(\mathbf{X}) = \sum_j \|\mathbf{X}_j\|_2$
- Smoothly varying over time: $\psi(\mathbf{X}) = \sum_{i,j} X_{i,j}^2$
- Block-wise restructuring: $\psi(\mathbf{X}) = \sum_j (\max_i |X_{i,j}|)$
- Perturbed node: $\psi(\mathbf{X}) = \min_{\mathbf{V}: \mathbf{V} + \mathbf{V}^T = \mathbf{X}} \sum_j \|\mathbf{V}_j\|_2$

How to solve the problem?

- The time-varying graphical lasso often requires analyzing many sensors over long time periods
 - Standard(centralized) solvers cannot scale!
- Split the problem up into a series of subproblems on a chain graph



Alternating Direction method of multipliers(ADMM)



- ADMM is **parallelizable** and **scalable**
- Without any global coordination, this message passing algorithm quickly converges to the optimal solution
- We can derive close-form solutions for all ADMM subproblems in TVGL algorithm



Extensions

- Asynchronous Observations:

Where samples are observed at irregularly-spaced intervals ($h_i = t_i - t_{i-1}$)

$$\min_{\Theta \in \mathcal{S}_{++}^p} \sum_{i=1}^T -l_i(\Theta_i) + \lambda \|\Theta_i\|_{od,1} + \beta \sum_{i=2}^T h_i \psi\left(\frac{\Theta_i - \Theta_{i-1}}{h_i}\right)$$

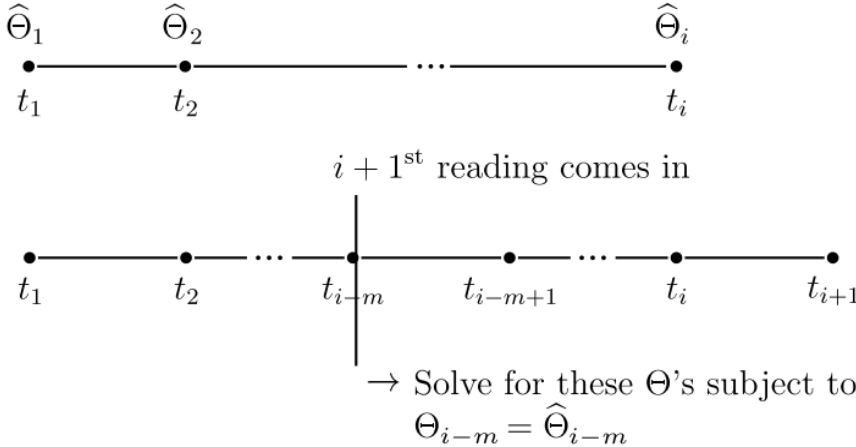
- Inferring Intermediate Networks:

We estimate Θ_s at any time s by create a dummy node connecting to the nearest observations $j - 1$ and j

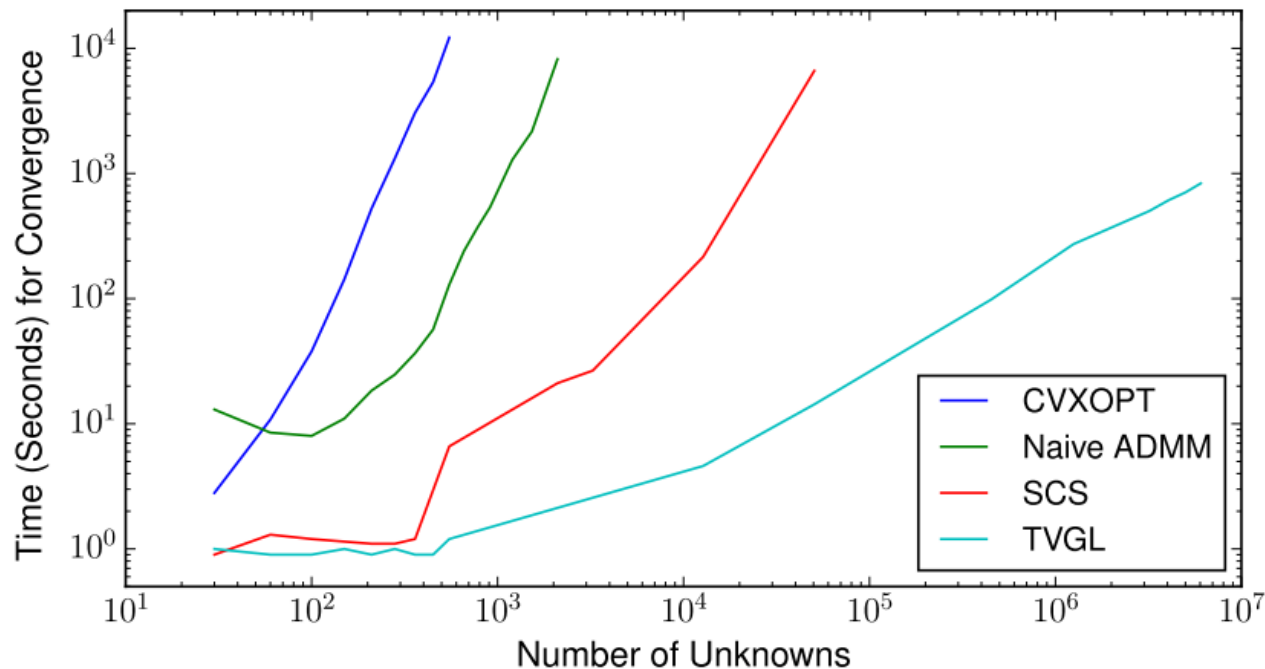
$$\min_{\Theta_s \in \mathcal{S}_{++}^p} w(s - t_{j-1})\psi(\Theta_s - \Theta_{j-1}) + w(t_j - s)\psi(\Theta_j - \Theta_s)$$

- Streaming Algorithm:

In order to guarantee computing time, we only solve for the **m most recent nodes.**

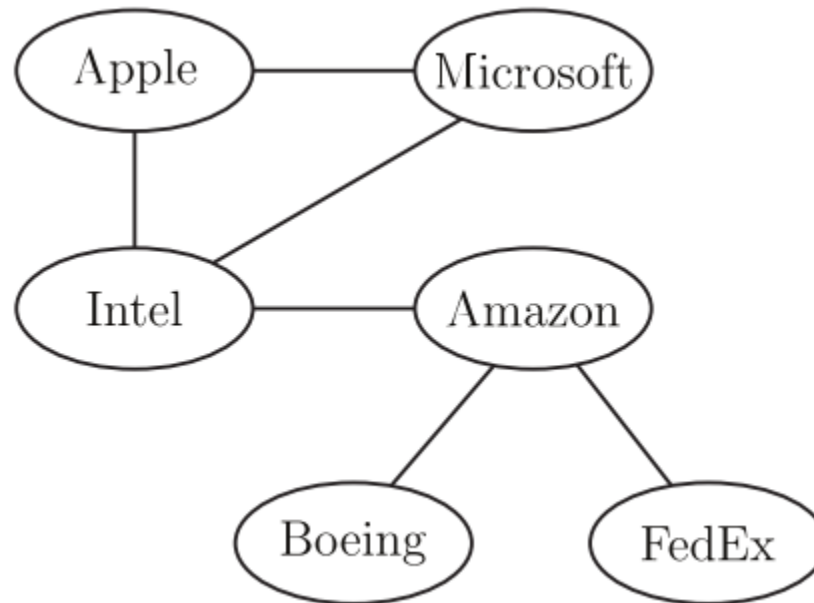


Experiments - Scalability



- **ADMM** can solve for millions of variables in just minutes!
 - Centralized solvers (CVXOPT, SCS) and naïve ADMM implementation explode computationally

Experiments – Case study

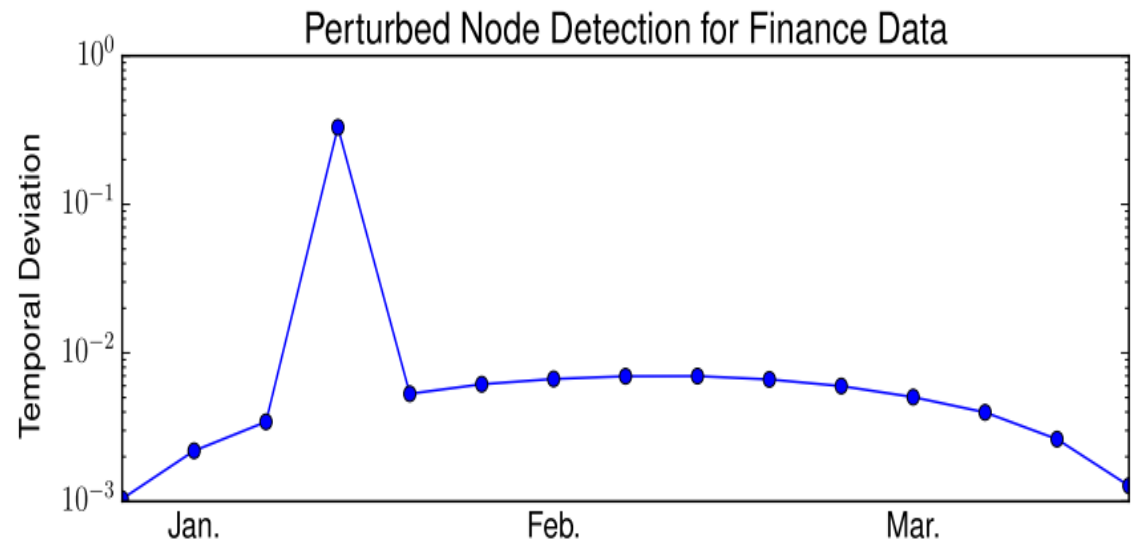


- **Dataset:** Daily stock prices of several large companies in early 2010
 - Treat the closing price of each stock as a daily sensor observation
- **Network Sparsity** shows the relationships between different companies



Experiments – Case study

- What insights can we find by looking at how the stock network evolves?
- **Perturbed-node penalty:**
 - The event that had largest single-node effect on dynamics of the network



	1	2	3	4	5	6
Apple: 1	■	■		■	■	
Microsoft: 2	■					
Amazon: 3						
Intel: 4	■					
Boeing: 5	■					
FedEx: 6						

What caused this shift?

- It occurred on the day that Apple announced the original iPad!

iPad (1st generation) +

From Wikipedia, the free encyclopedia

Not to be confused with iPad Mini (1st generation).

The **first-generation iPad** (/ˈaɪpæd/ *EYE-pad*) is a tablet computer designed and marketed by **Apple Inc.** as the first in the iPad line. The device features an Apple A4 processor, a 9.7" touchscreen display, and, on certain variants, the capability of accessing cellular networks. Using the iOS operating system, the iPad can play music, send and receive email and browse the web. Other functions, which include the ability to play games and access references, GPS navigation software and social network services can be enabled by downloading apps.

The device was announced and unveiled on January 27, 2010 at a media conference. On April 3, 2010, the Wi-Fi variant of the device was released in the **United States**, followed by the release of the Wi-Fi + Cellular variant on April 30. On May 28, it was released in Australia, Canada, France, Japan, Italy, Germany, Spain, Switzerland and the United Kingdom.

The device received primarily positive reviews from various technology blogs and publications. Reviewers praised the device for its wide range of capabilities and labelled it as a competitor to laptops and netbooks. Some aspects were criticized, including the closed nature of the operating system and the lack of support for the Adobe Flash multimedia format. During the first 80 days, three million iPads were sold. By the launch of the iPad 2, Apple sold more than 15 million iPads.

On March 2, 2011, Apple announced the iPad 2 and the discontinuation of production of the original iPad.^[6]

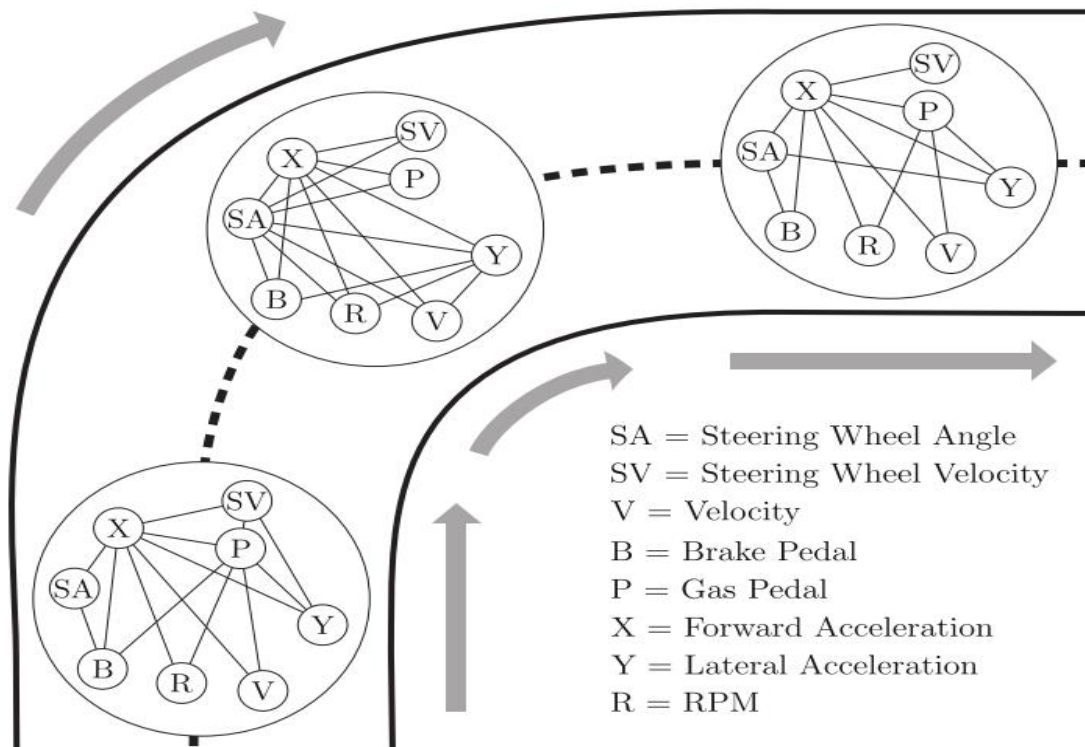
Contents [hide]

1 History



Experiments – Case study

- Dynamic sensor network can be used as a “signature”
 - Understand driving styles
 - Identify drivers
 - Detect when drivers are distracted, drowsy, drunk, etc.






Summary

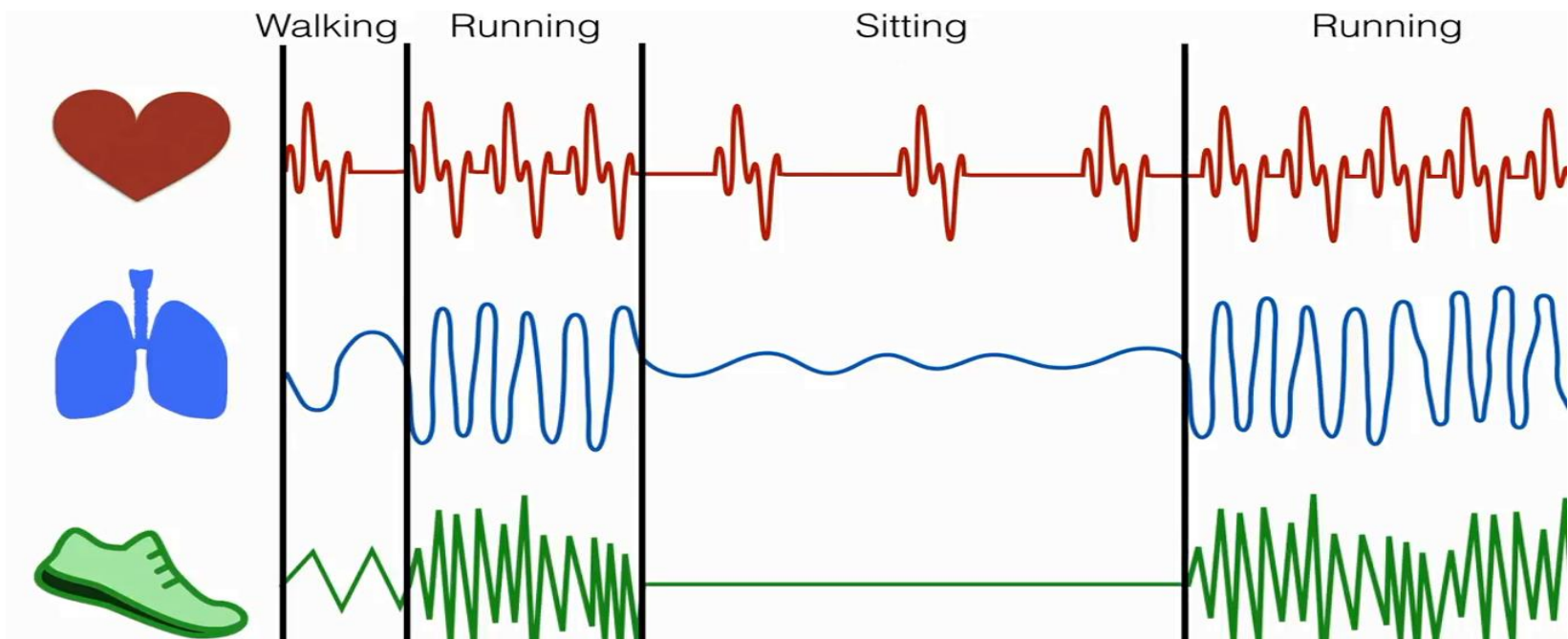
- The time-varying graphical lasso (TVGL)
- A computationally tractable method of inferring dynamic networks
- Robust and scalable solution based on ADMM
- Different temporal dependencies allow for many types of structural evolutions over time
- The same setup can be applied on a variety of different applications

Papers

- Network Inference via the Time-Varying Graphical Lasso
- David Hallac, Stephen Boyd, Jure Leskovec, etc. KDD'17
- Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data
- David Hallac, Stephen Boyd, Jure Leskovec, etc. KDD'17 

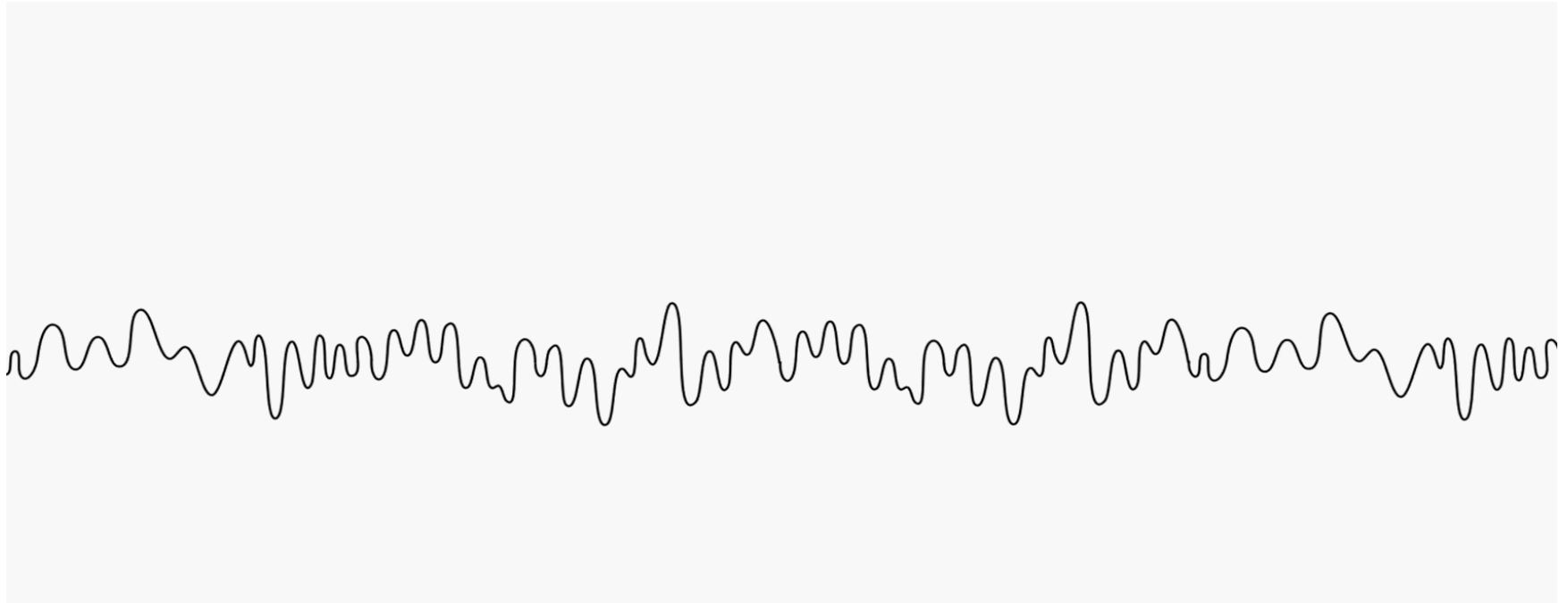
Discovering Structure in the Data

- These large amounts time-series data mentioned before can often be broken down into a sequence of states
- For example raw sensor data from a fitness tracking device can be interpreted as a temporal sequence of actions showed below



Simultaneously Segmentation and Clustering

- However in general these states are not predefined and we do not know what they are or what they refer to
- We need to learn both the states themselves and also how the time series splits into these states

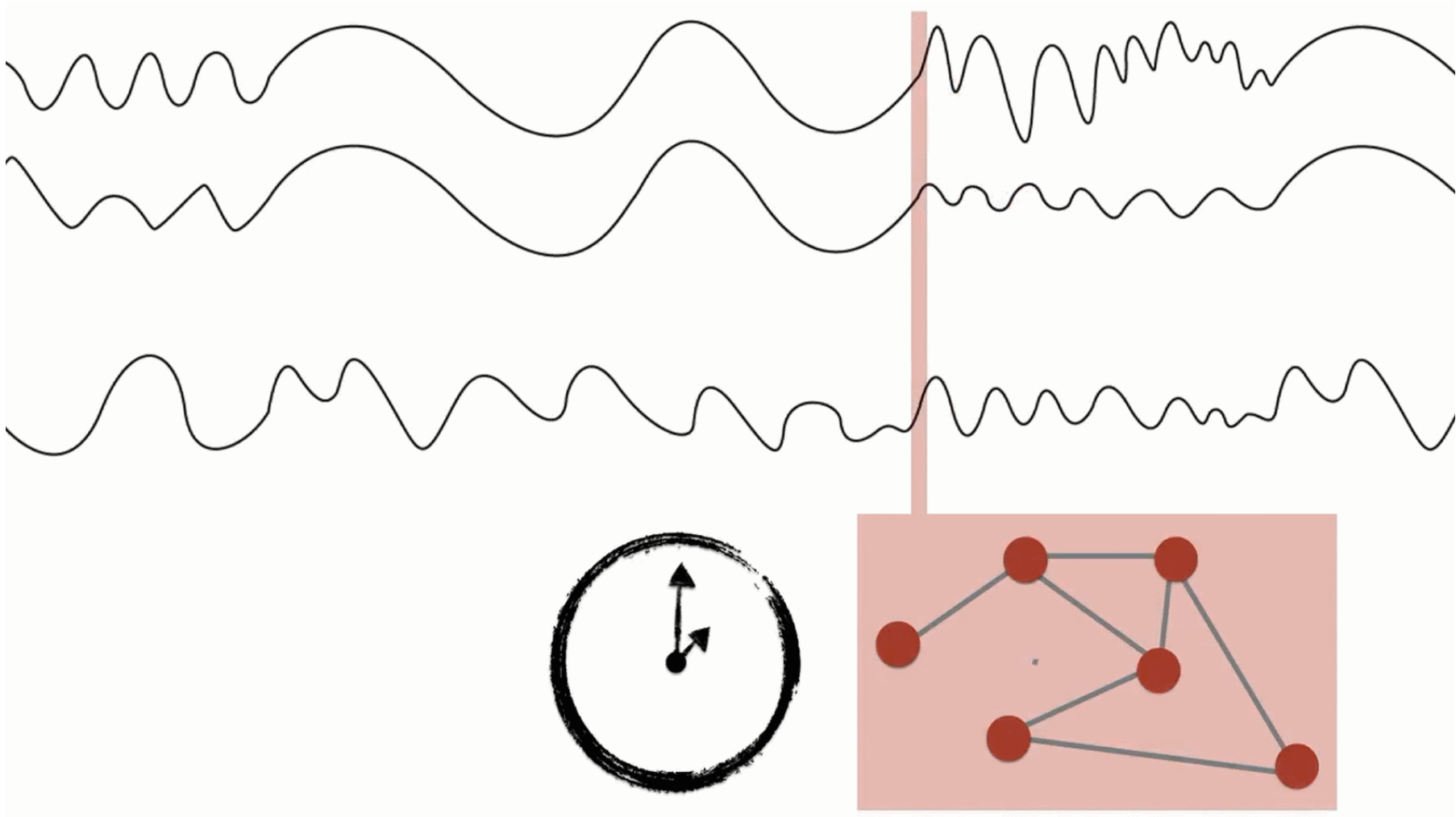




What does Clustering do?

- Given: multivariate time series data of length T
- Goal: Assign each point in the time series into one of K different states for clustering, each defined by a unique “pattern”
- Temporal consistency (= Do segmentation):
 - **Adjacent** points should be encouraged to belonging to the **same** cluster
 - This yields segments of time are intervals of time where the state of the system remains constant

Recall the Correlation Network



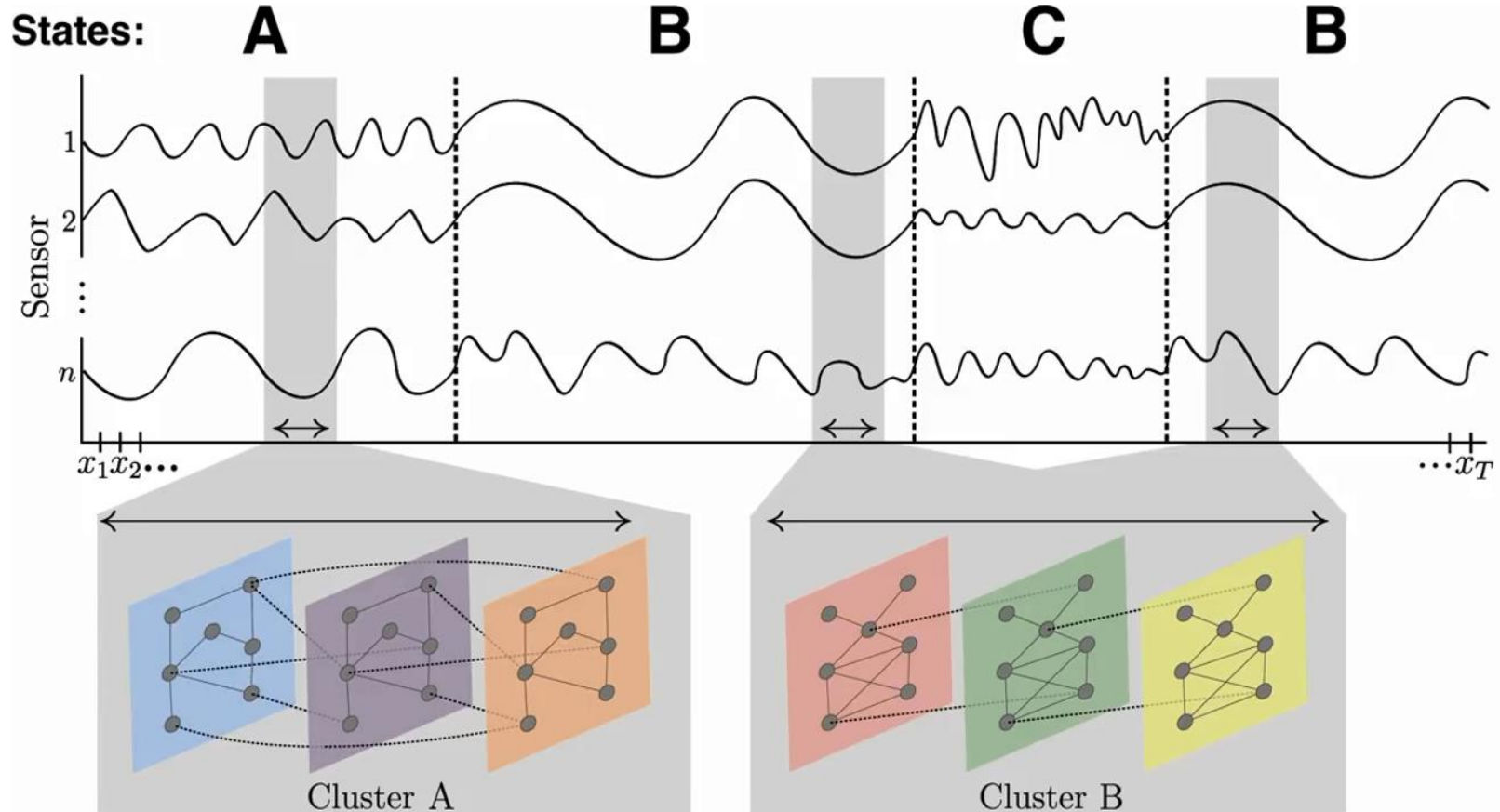


Definition of a Cluster

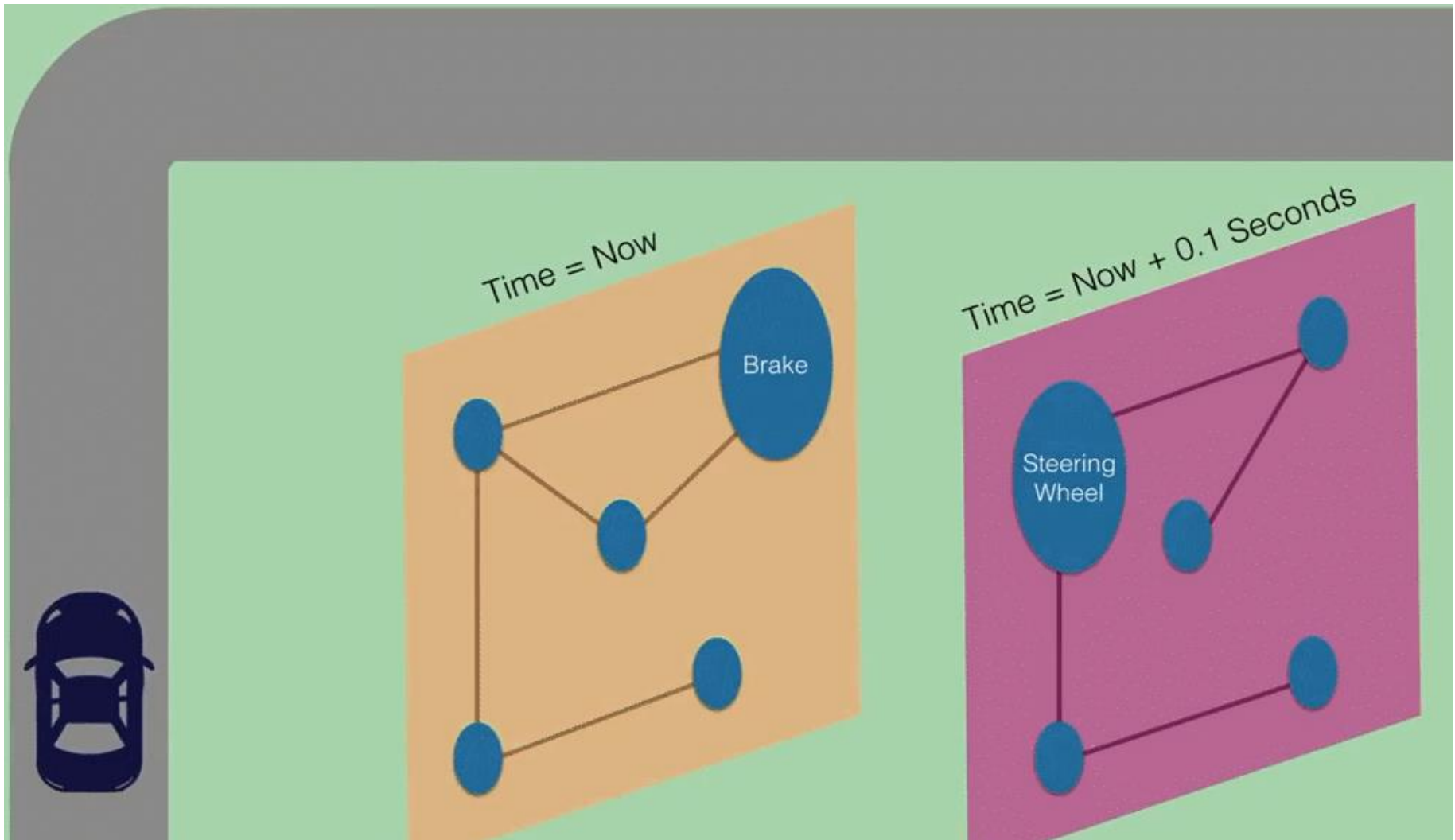
- We assign each point to a cluster by analyzing a short time window ending at the point
 - It provides **real and more context** for the data (e.g. the case of automobiles)
- Each cluster is then defined by a **multilayer** correlation network, or Markov Random Field (MRF), showing the correlation between different sensors at that point in time and over that window

Multi-layer Correlation Network

- These networks encode the conditional dependency structural relationships between the different centers across time.



An example: Turn and Slow down



Problem Setup

- Consider a time series of T sequential observations,

$$\mathbf{x}_{\text{orig}} = \begin{bmatrix} | & | & | & \dots & | \\ x_1 & x_2 & x_3 & \dots & x_T \\ | & | & | & \dots & | \end{bmatrix},$$

Our goal is to cluster these T observations into K clusters.

- Rather than just look at \mathbf{x}_t , we instead cluster a short subsequence of size $w \ll T$ that ends at t , consists of $\mathbf{x}_{t-w+1}, \dots, \mathbf{x}_t$.
- We refer to these subsequences from \mathbf{X}_1 to \mathbf{X}_T , as \mathbf{X} .
Now our goal is to cluster these subsequence $\mathbf{X}_1, \dots, \mathbf{X}_T$, and we encourage adjacent subsequences to belong the same cluster.

Toeplitz Inverse Covariance-Based Clustering(TICC)

- Define each cluster by a Gaussian inverse covariance $\Theta_i \in R^{nw \times nw}$, Our overall optimization problem is:

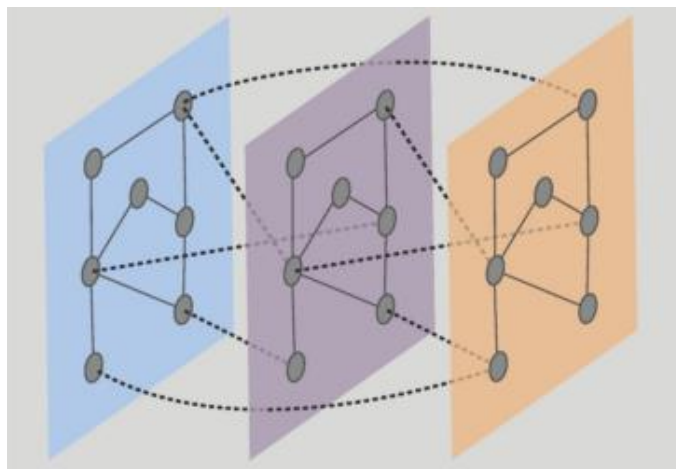
$$\operatorname{argmin}_{\Theta \in \mathcal{T}, P} \sum_{i=1}^K \left[\overbrace{\|\lambda \circ \Theta_i\|_1}^{\text{sparsity}} + \sum_{X_t \in P_i} \left(\overbrace{-\ell\ell(X_t, \Theta_i)}^{\text{log likelihood}} + \overbrace{\beta \mathbb{1}\{X_{t-1} \notin P_i\}}^{\text{temporal consistency}} \right) \right]$$

- We solve for these K inverse covariances, $\Theta = \{\Theta_1, \dots, \Theta_K\}$, and the resulting assignment sets $P = \{P_1, \dots, P_K\}$ where $P_i \subset \{1, 2, \dots, T\}$.
- \mathcal{T} is the set of symmetric block Toeplitz $nw * nw$ matrices

Block Toeplitz Matrices

- Sparsity in the Toeplitz Matrix defines the MRF edge structure

$$\Theta_i = \begin{bmatrix} A^{(0)} & (A^{(1)})^T & (A^{(2)})^T \\ A^{(1)} & A^{(0)} & (A^{(1)})^T \\ A^{(2)} & A^{(1)} & A^{(0)} \end{bmatrix}$$



- Toeplitz constraint enforces **time invariance** (Key idea!)

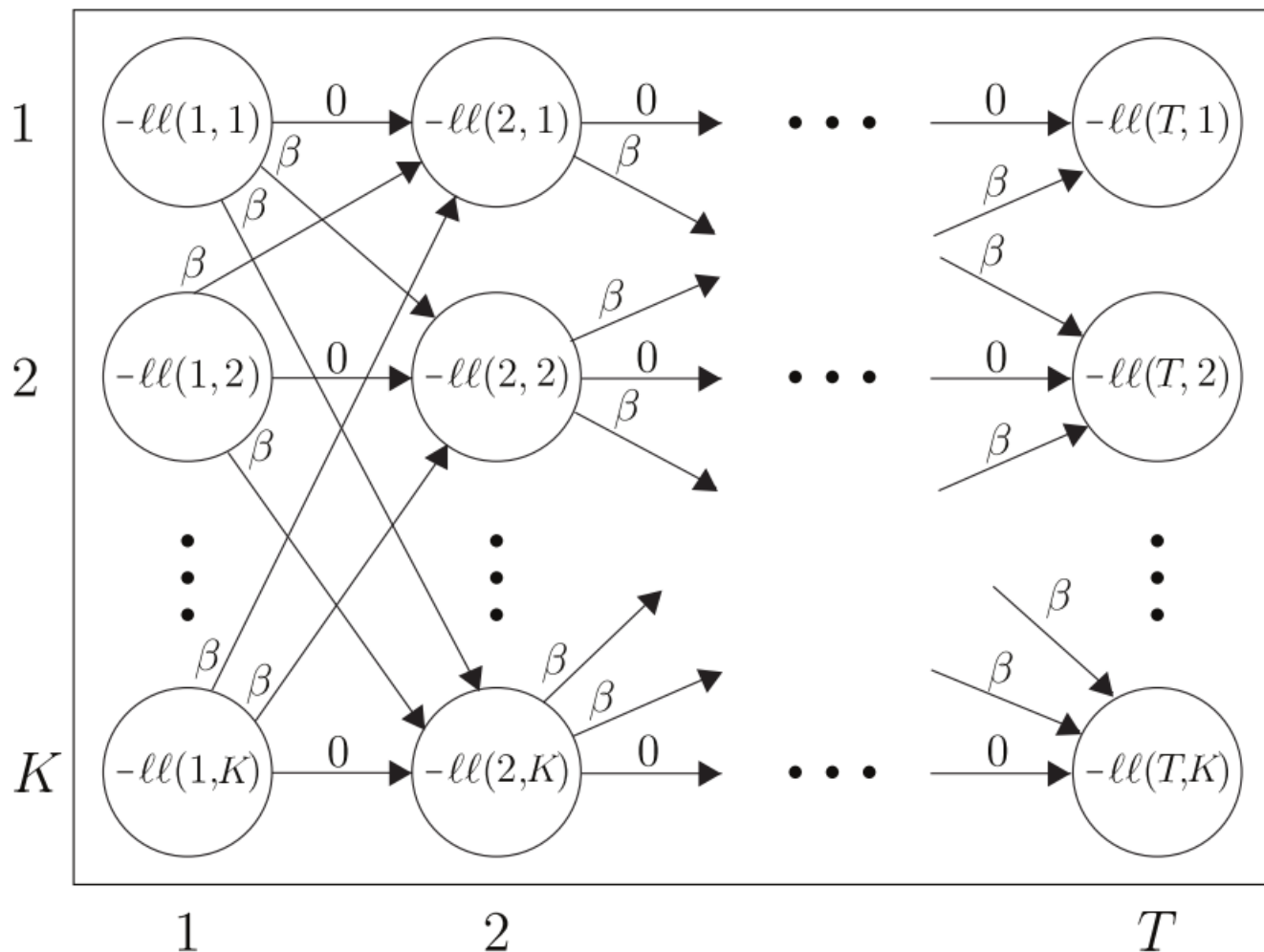


Expectation Maximization

- TICC is highly con-convex
 - But we can use a EM-like approach to solve it
- Alternate between:
 - Assigning points to clusters in a temporally consistent way
 - Updating the cluster parameters

Assigning Points to Clusters

- We can solve this with dynamic programming (e.g. Viterbi)



Updating the Cluster Parameters

- Toeplitz Graphical Lasso:

$$\text{minimize} \quad -\log \det \Theta_i + \text{tr}(\mathbf{S}_i \Theta_i) + \frac{1}{|P_i|} \|\lambda \circ \Theta_i\|_1$$

$$\text{Subject to} \quad \Theta_i \in \mathcal{T}$$

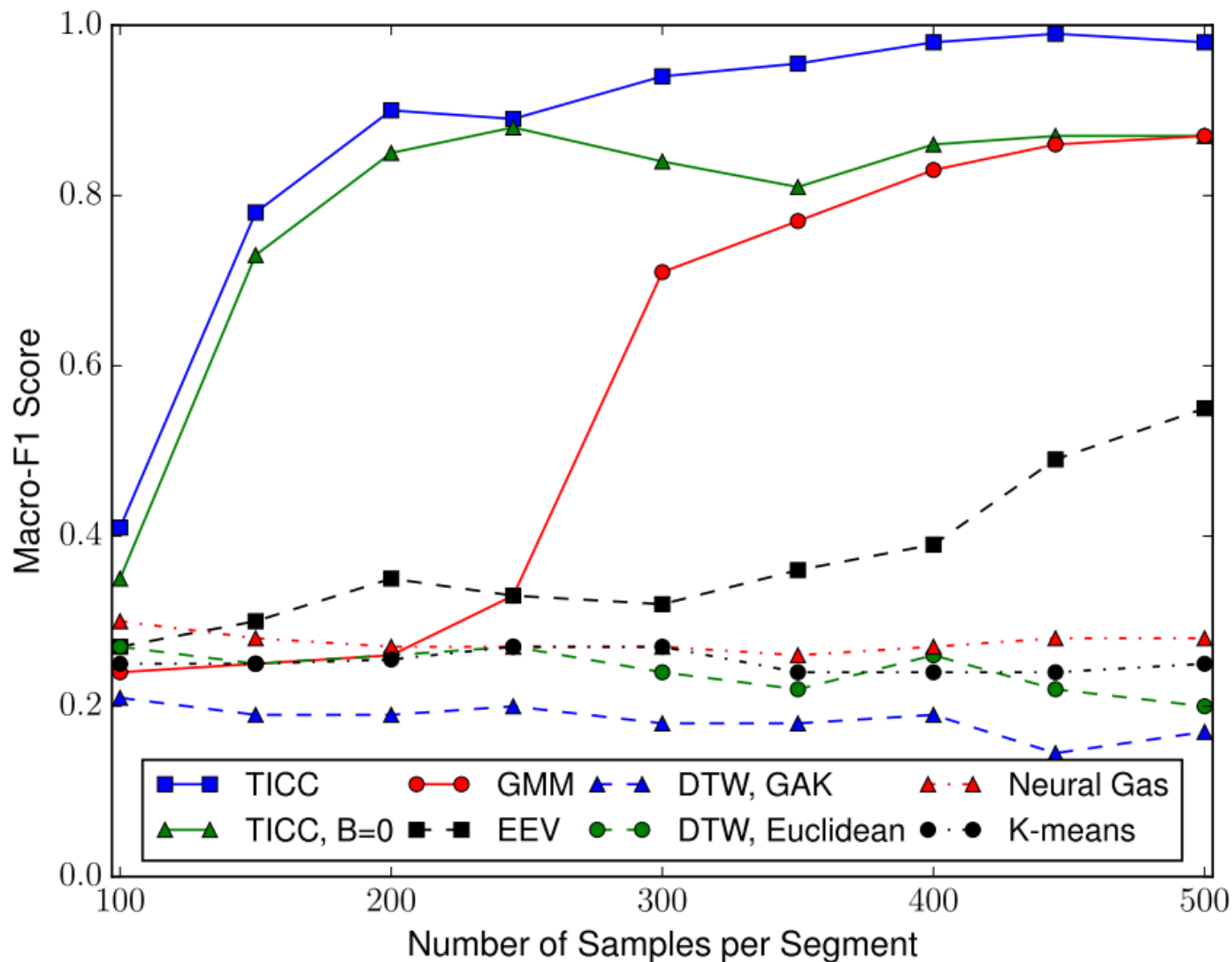
- We can derive an ADMM solution with closed-form proximal operators to solve the problem efficiently

Experiments – Clustering F1 Score

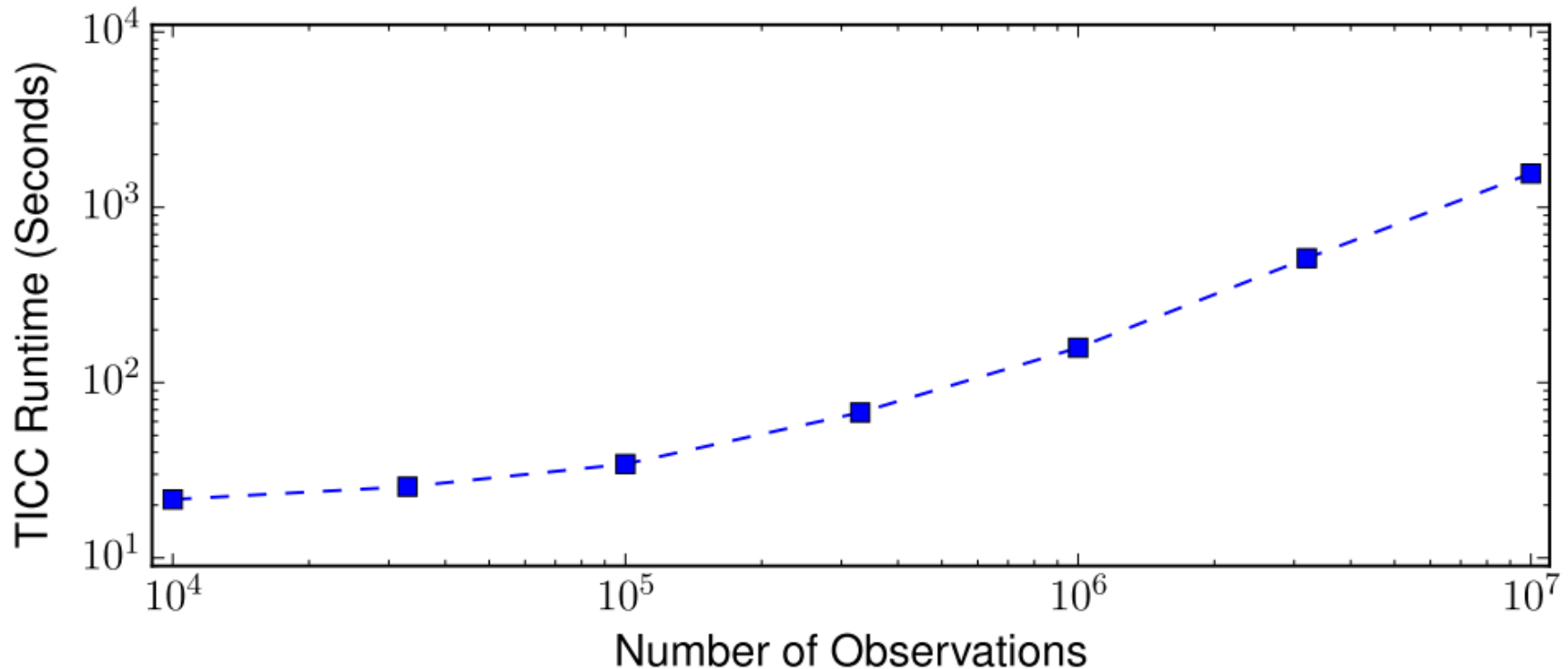
		Temporal Sequence			
Clustering Method		1,2,1	1,2,3,2,1	1,2,3,4,1,2,3,4	1,2,2,1,3,3,3,1
TICC		0.92	0.90	0.98	0.98
Model- Based	TICC, $\beta = 0$	0.88	0.89	0.86	0.89
	GMM	0.68	0.55	0.83	0.62
	EEV	0.59	0.66	0.37	0.88
Distance- Based	DTW, GAK	0.64	0.33	0.26	0.27
	DTW, Euclidean	0.50	0.24	0.17	0.25
	Neural Gas	0.52	0.35	0.27	0.34
	K-means	0.59	0.34	0.24	0.34

- Use synthetic data (with known ground truth)
- At least 41% higher F1 score than every other non-TICC method!

Experiments – Robustness



Experiments – Scalability



- 10 million observations, each 50-dimensional, in just 20 minutes



Experiments – Case study

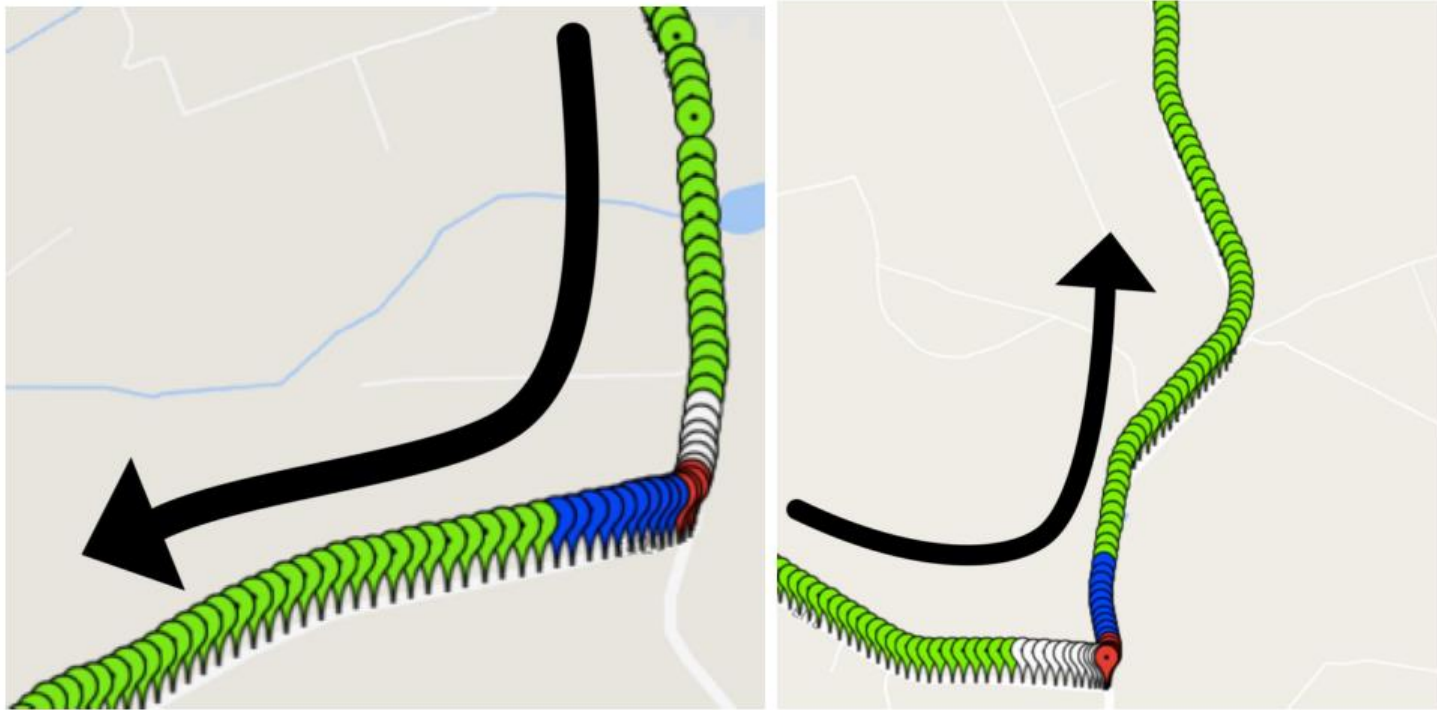
- Automobile sensor data
- 7 sensors every 0.1 seconds (1 hour of 36,000 observations):
 - Brake Pedal Position
 - Forward (X-) Acceleration
 - Lateral (Y-) Acceleration
 - Steering Wheel Angle
 - Vehicle Velocity
 - Engine RPM
 - Gas Pedal Position
- Window size of 1 second, and $K=5$ picked by using BIC

Experiments – Case study

	Interpretation	Brake	X-Acc	Y-Acc	SW Angle	Vel	RPM	Gas
#1	Slowing Down	25.64	0	0	0	27.16	0	0
#2	Turning	0	4.24	66.01	17.56	0	5.13	135.1
#3	Speeding Up	0	0	0	0	16.00	0	4.50
#4	Driving Straight	0	0	0	0	32.2	0	26.8
#5	Curvy Road	4.52	0	4.81	0	0	0	94.8

- Betweenness centrality for each sensor in each of the five clusters, show how “important” each sensor is, and more specifically how much it directly affects the other sensor values.

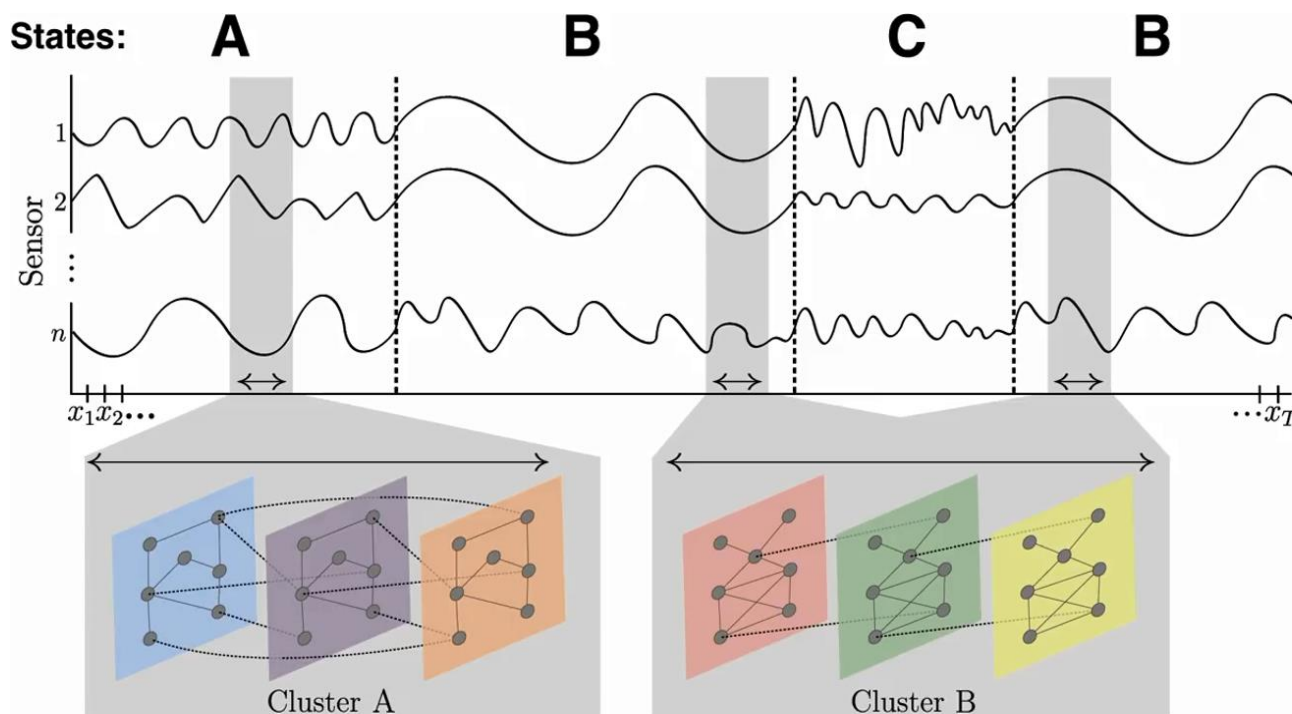
Experiments – Case study



- Going Straight, Slowing down, Turning, Speeding up
- Based on structure, rather than distance, left turn and right turn look very similar

Summary

- Toeplitz Inverse Covariance-Based Clustering (TICC)
 - Simultaneously do segmentation and clustering of multivariate time series data



Thanks

Code all available at [TVGL](#) & [TICC](#)



Yao Yang
up9288yy@gmail.com