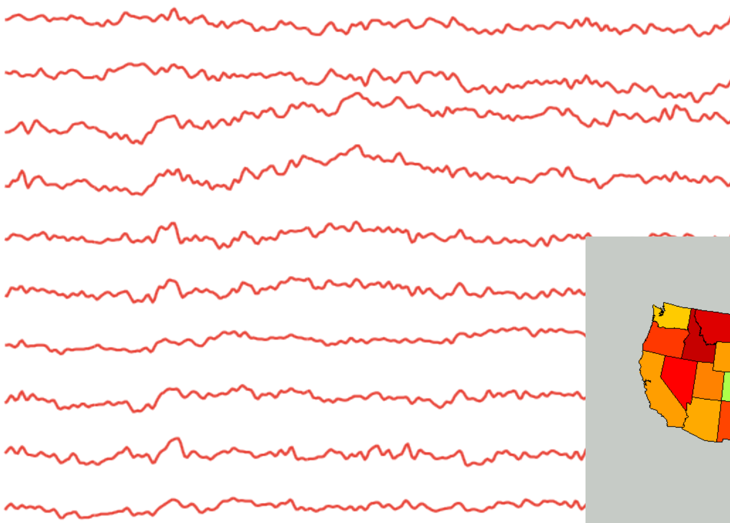
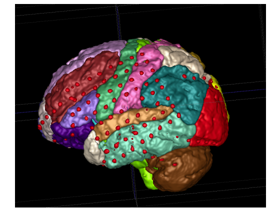
Bayesian Time Series: Structured Representations for Scalability

Emily Fox

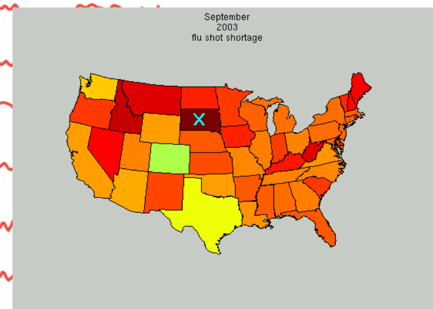
UNIVERSITY OF WASHINGTON

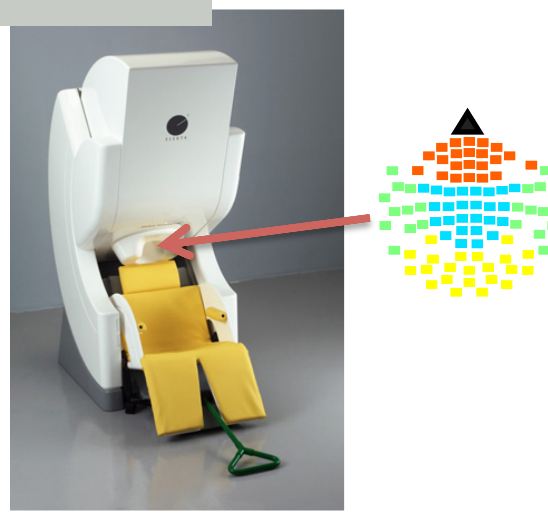
Time Series

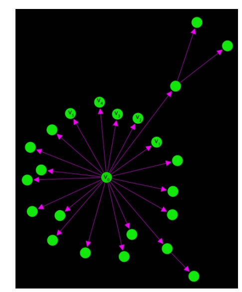




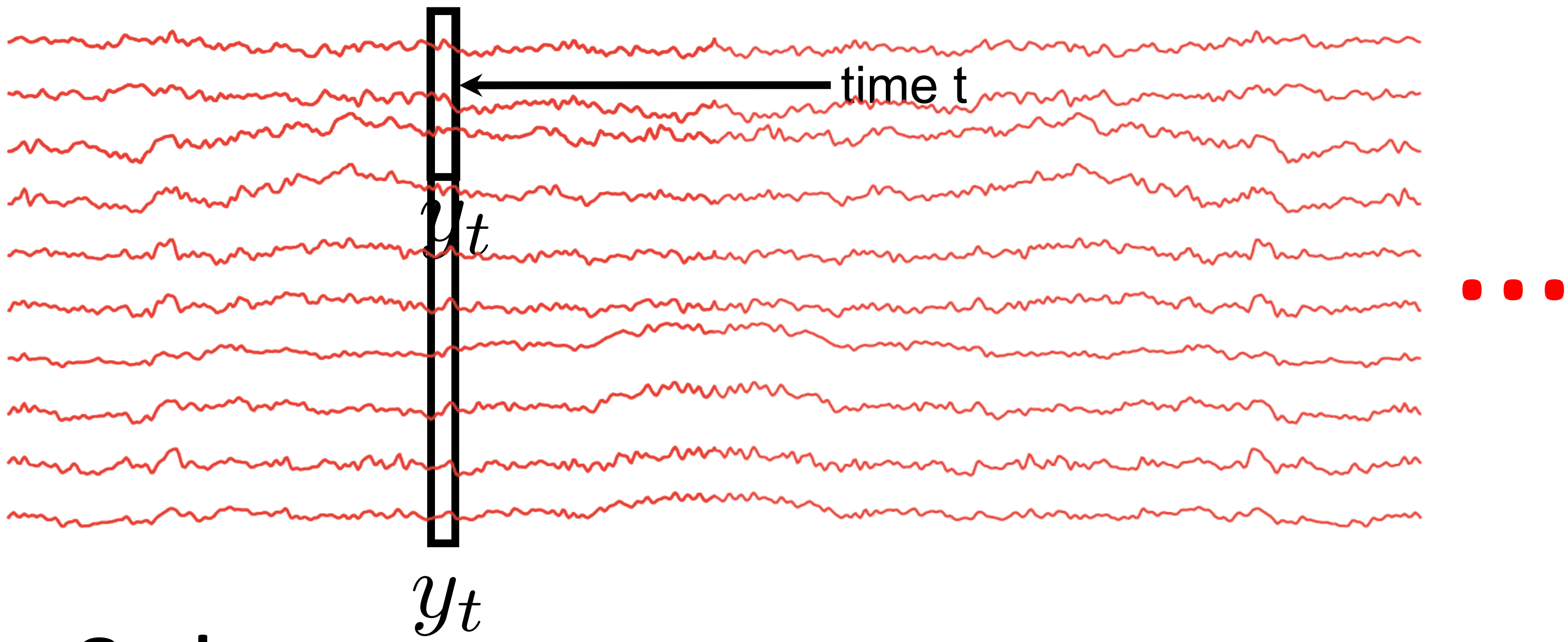

INDU +191.92
INDP 14087.55


September
2003
flu shot shortage






HUMAN GENOME, IN NUMBERS
6 billion
DNA LETTERS
46 CHROMOSOMES
22,000
GENES
$9,500
COST TO SEQUENCE
500GB
SIZE ON DISK



Goals:

- Evolution – Dynamics across time
- Relational structure – Dependencies between series

Modeling challenges:

- Large p – Many dimensions/series
- Irregular grid of observations
- Missing values
- Heterogeneous data sources
- …

Computational challenges:

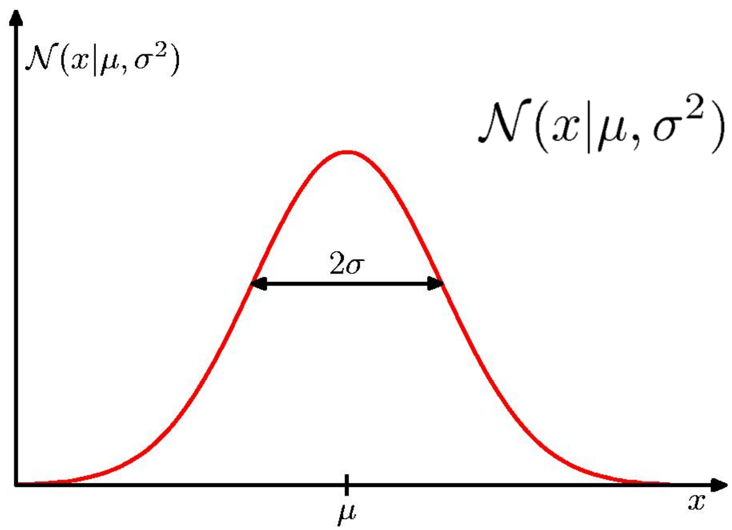
- Large n – Long time series
- Streaming data – Continuum of observations

Preliminaries/Review

- Multivariate Gaussians
- Hidden Markov models (HMMs)
- Vector autoregressive (VAR) processes
  - Stability/stationarity
- Gaussian state space models
  - Identifiability

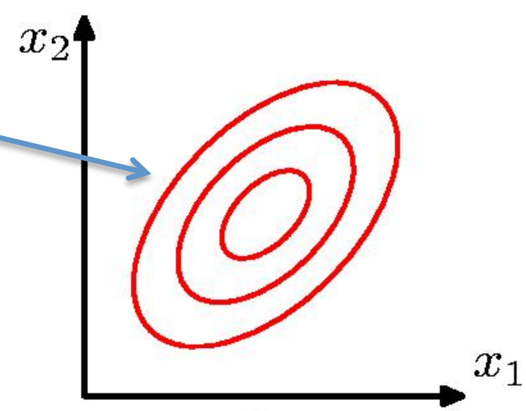
Quick Review of Gaussians

- Univariate and multivariate Gaussians



$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

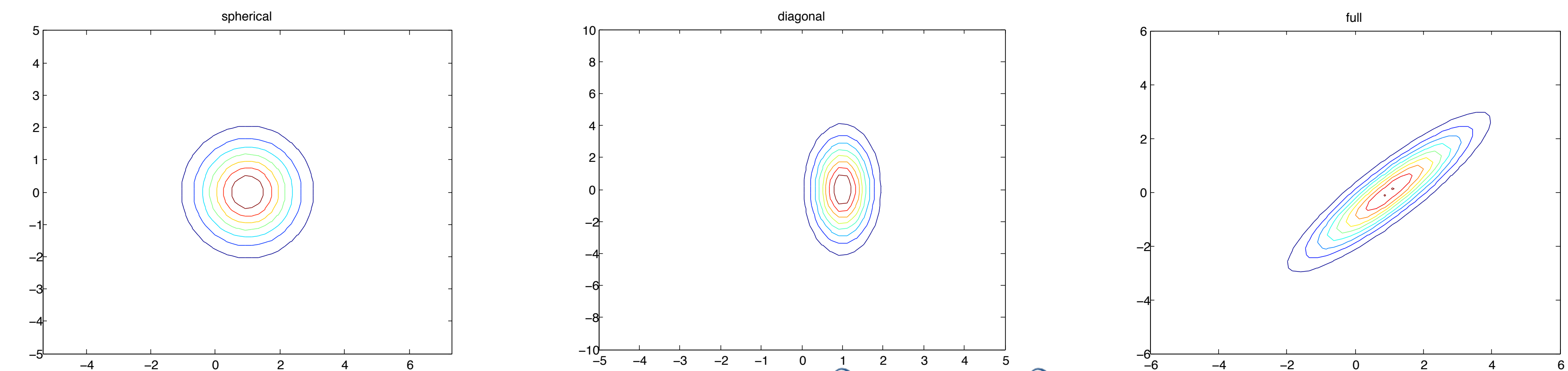
Covariance defines shape (eigvecs+eigvals)



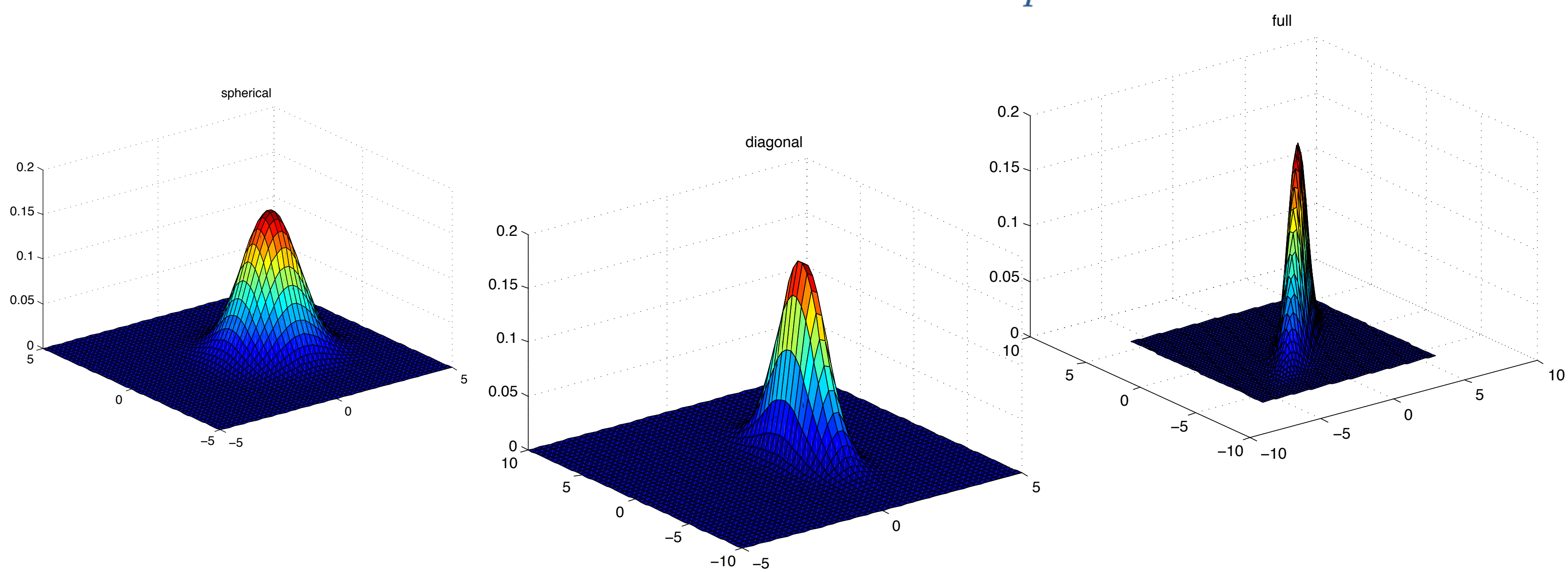
$$\Sigma = \mathrm{cov}(\mathbf{x})$$

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

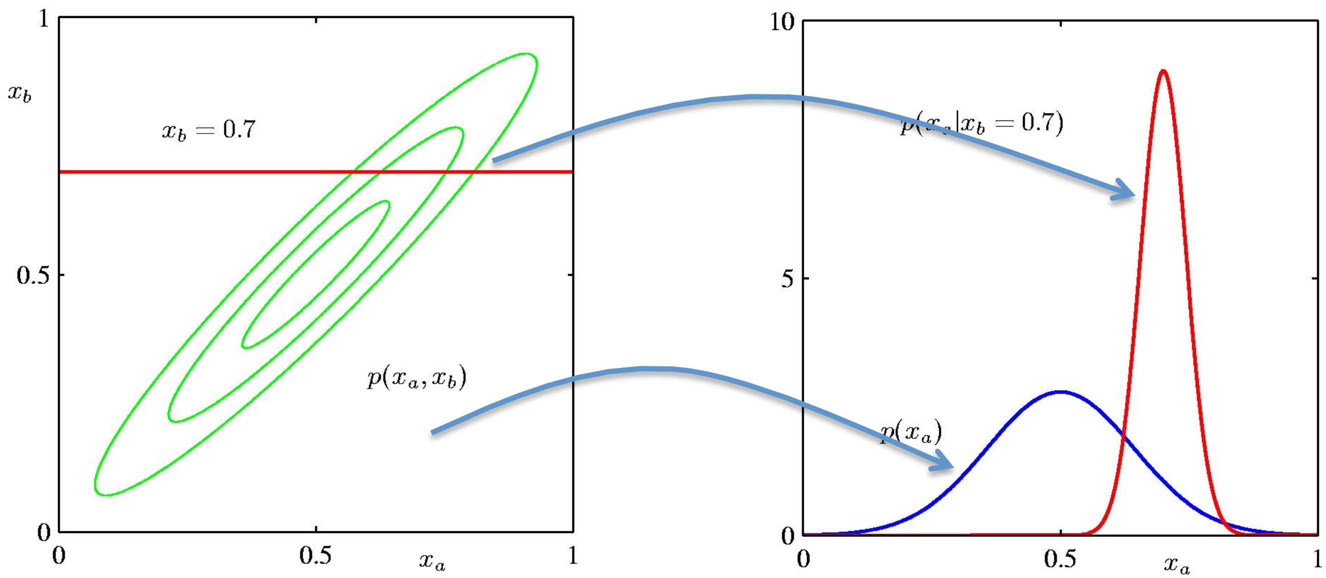
Two-Dimensional Gaussians



$$\Sigma = \sigma^2 I \qquad \Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2) \qquad \Sigma \text{ general}$$



Conditional & Marginal Distributions



$$\begin{pmatrix} x_a \\ x_b \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ab}^T & \Sigma_{bb} \end{bmatrix} \right)$$

Marginally: $x_a \sim N(\mu_a, \Sigma_{aa})$

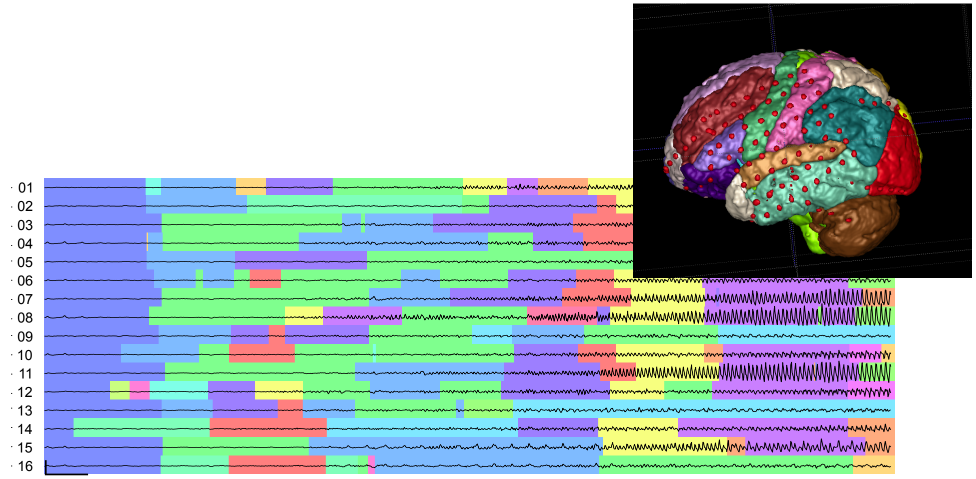
Conditionally:

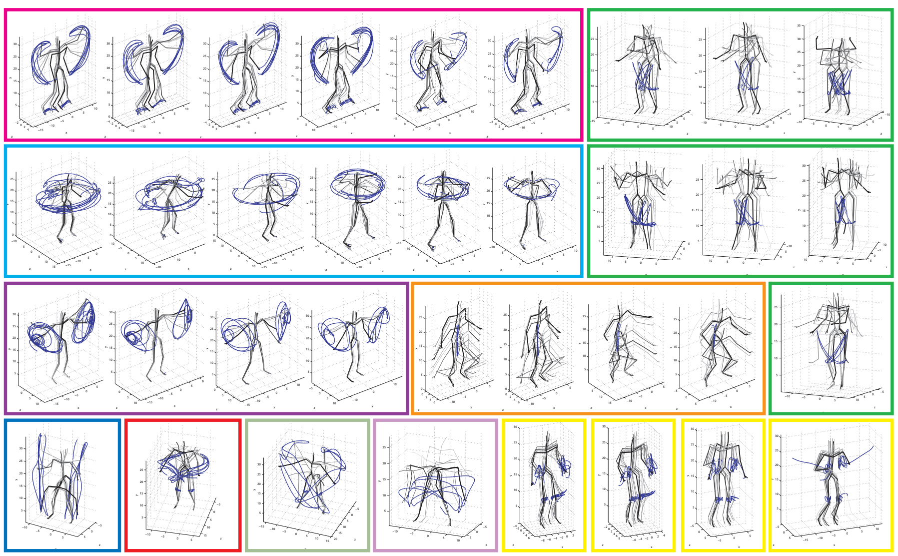
$$x_a \mid x_b \sim N(\mu_a - \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b), \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ab}^T)$$

Hidden Markov Models

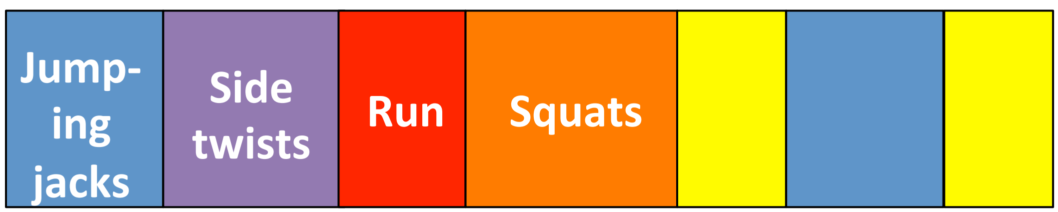
Example applications:

- Parsing EEG recordings
- Discovering behaviors in videos
- Speech segmentation
- Volatility regimes in financial time series
- Genomics
- …

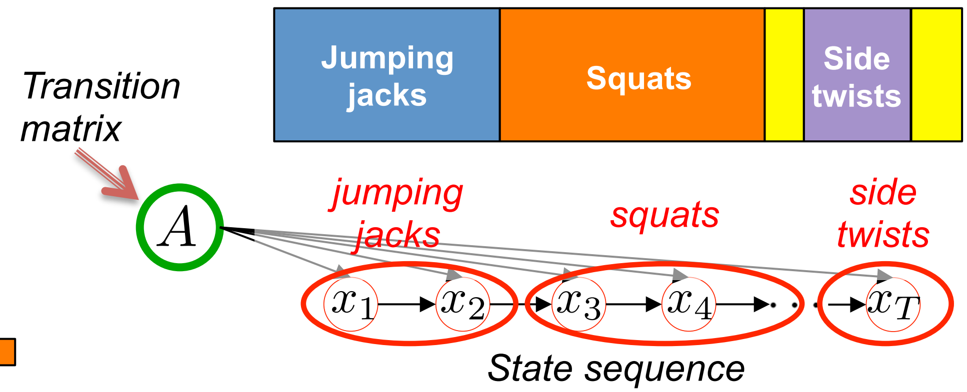




Example: Motion Capture Segmentation

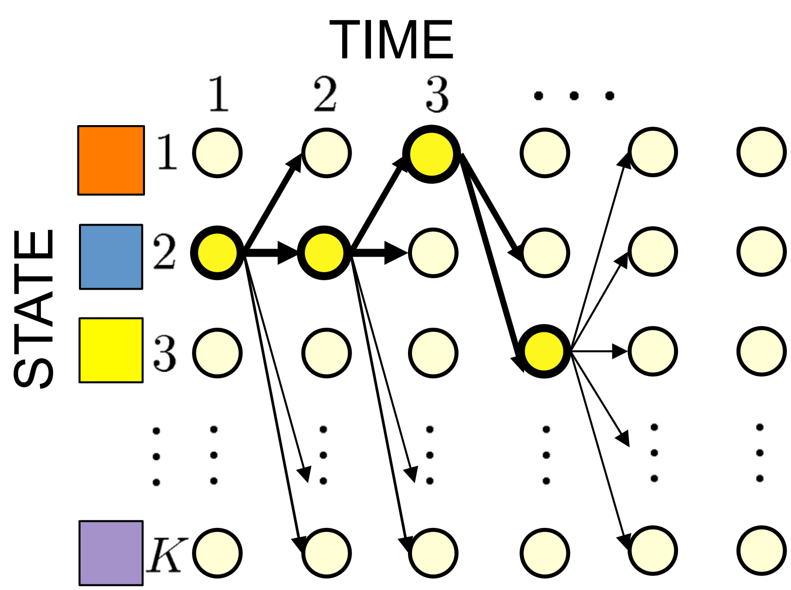


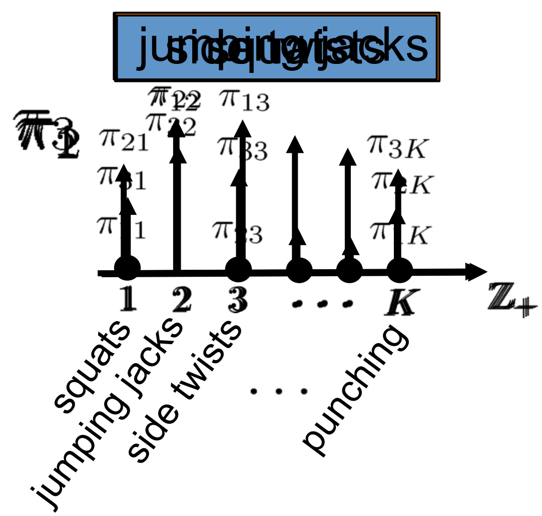
Hidden Markov Model

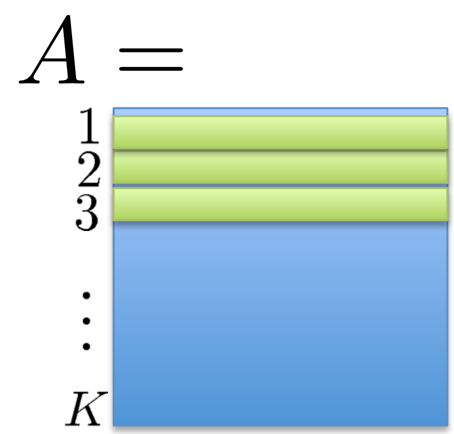


- Markov transition dynamics:

$$\Pr(x_t = \square \,|\, x_{t-1} = \square) = A_{\square\square}$$

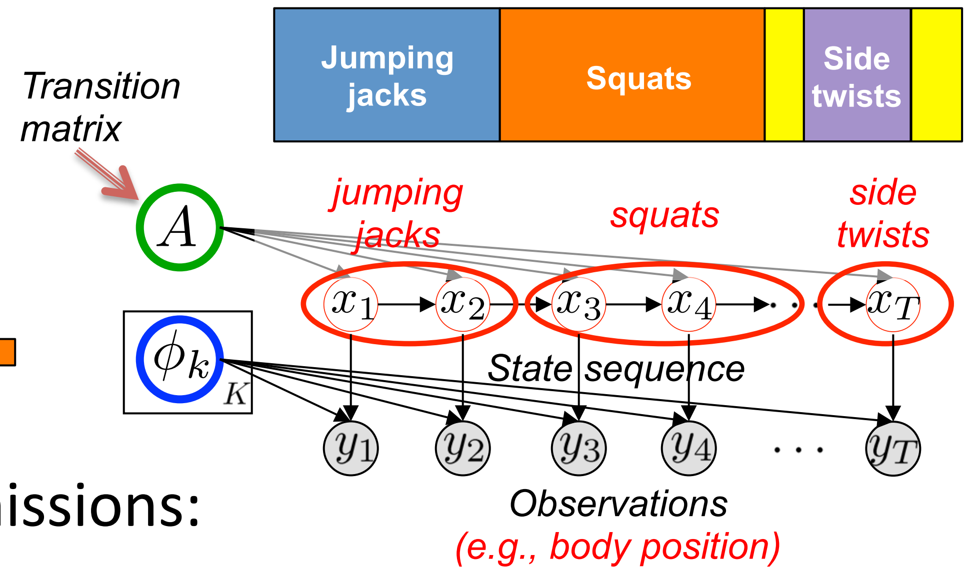






Tutorial:
Rabiner, *Proc. IEEE* 1989

Hidden Markov Model



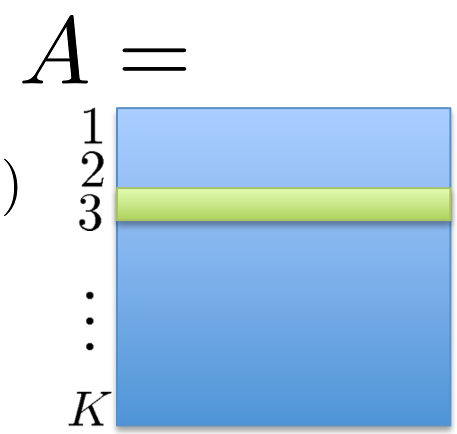
- Markov transition dynamics:

$$\Pr(x_t = \blacksquare \mid x_{t-1} = \blacksquare) = A_{\blacksquare\blacksquare}$$

- Conditionally independent emissions:

$$y_t \mid x_t = \blacksquare \sim F(\phi_{\blacksquare})$$

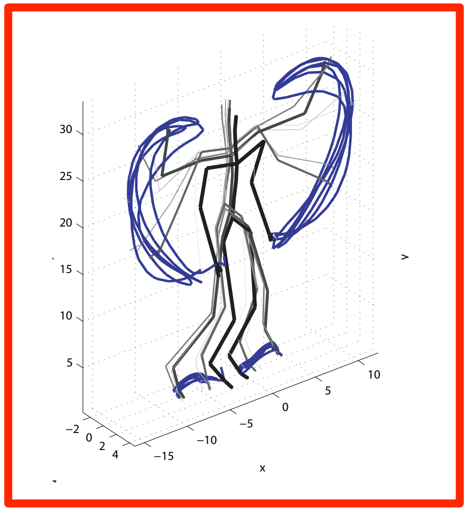
- Latent Markov chain structure enables
  - Efficient computation of marginals $p(x_t \mid y_1, \ldots, y_T)$ using the *forward-backward* algorithm
  - Most-probable sequence via *Viterbi*
  - Parameter learning using *Baum-Welch* (EM for HMMs)



Tutorial:
Rabiner, *Proc. IEEE* 1989

Motivating Other Time Series Models

Jumping Jacks



Side Twists

Arm Circles

. . .

Vector autoregressive (VAR) process:

$$y_t = \sum_{i=1}^{r} A_i y_{t-i} + e_t \quad e_t \sim N(0, \Sigma)$$

Stationary VAR Processes

$$y_t = \sum_{i=1}^{r} A_i y_{t-i} + e_t \quad e_t \sim N(0, \Sigma)$$

- If the *companion matrix* has eigenvalues λ with |λ|<1, then the process is stable

$$\begin{bmatrix} A_1 & A_2 & \cdots & A_r \\ I & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & \cdots & I & 0 \end{bmatrix}$$

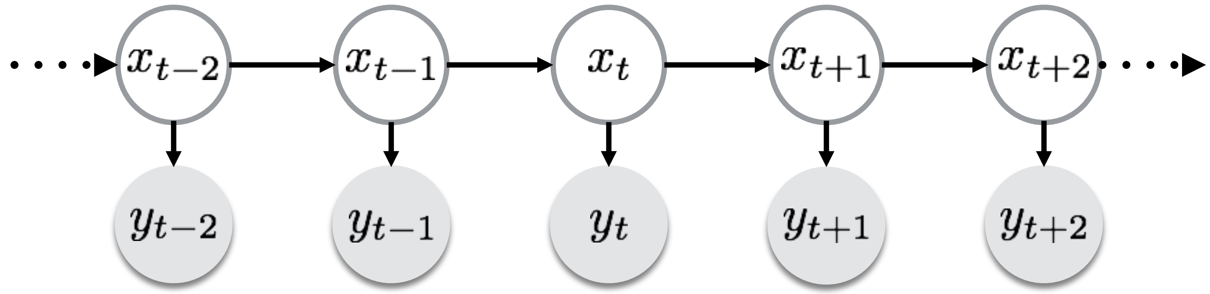
- If initialized infinitely in the past, then stationary

$$E[y_t] = \mu = 0 \qquad \mathrm{cov}(y_t, y_{t+h}) = \Gamma(h)$$

- For VAR(1) process, marginal covariance satisfies

$$\Gamma(0) = A_1 \Gamma(0) A_1' + \Sigma$$

State Space Models



- Like HMMs, but continuous-valued latent state sequence

$$x_t = Ax_{t-1} + e_t \quad e_t \sim N(0, \Sigma)$$
$$y_t = Cx_t + w_t \quad w_t \sim N(0, R)$$

- Entire class of equivalent systems from input/output perspective by changing latent space via $x_t \to Tx_t$:

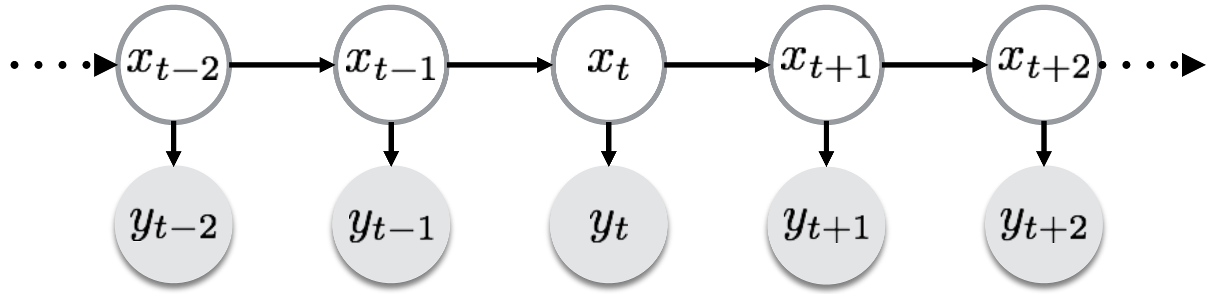
$T^{-1}AT$

$$x_t = \tilde{A}x_{t-1} + e_t \quad e_t \sim N(0, \Sigma)$$
$$y_t = \tilde{C}x_t + w_t \quad w_t \sim N(0, R)$$

CT

Constrain A, Σ, or C

State Space Models

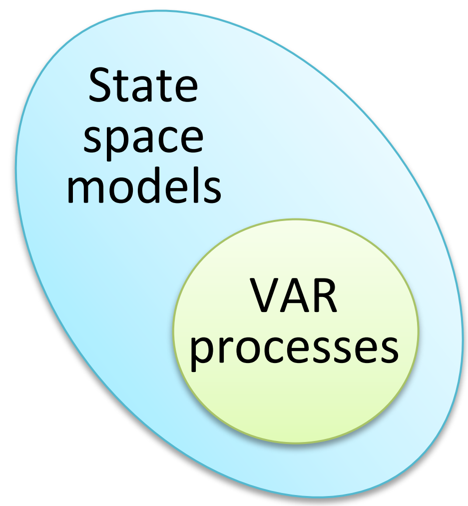


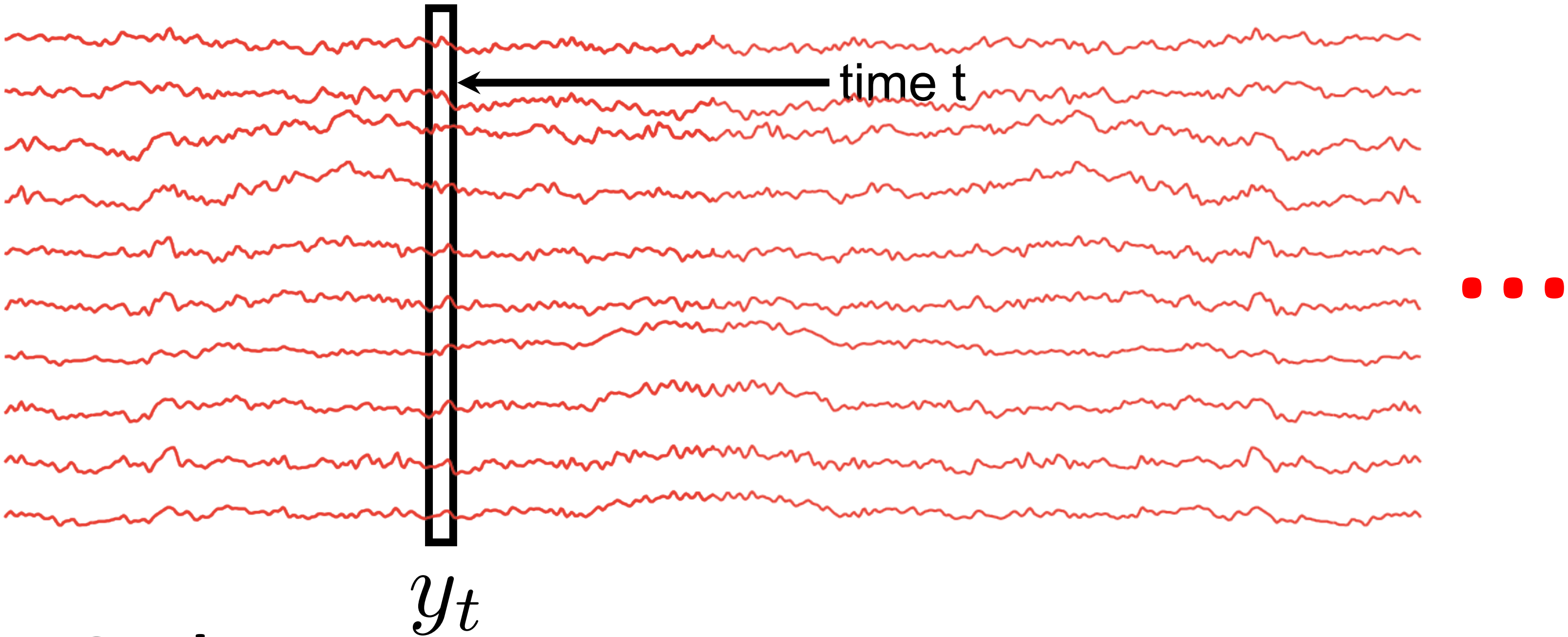
- Can write a VAR(*r*) process in state space form via

$$y_t = \sum_{i=1}^{r} A_i y_{t-i} + e_t \quad e_t \sim N(0, \Sigma)$$

$$x_t = \begin{bmatrix} A_1 & A_2 & \cdots & A_r \\ I & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & \cdots & I & 0 \end{bmatrix} x_{t-1} + \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} e_t$$

$$y_t = \begin{bmatrix} I & 0 \cdots & 0 \end{bmatrix} x_t$$





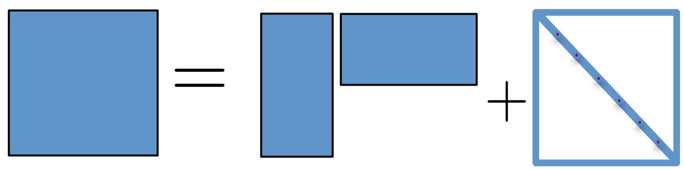
Goals:
- Evolution – Dynamics across time
- Relational structure – Dependencies between series

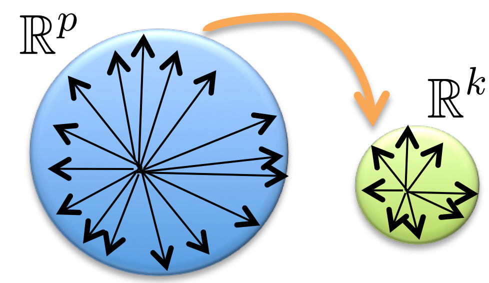
Modeling challenges:
- Large p – Many dimensions/series
- Irregular grid of observations
- Missing values
- Heterogeneous data sources
- ...

Computational challenges:
- Large n – Long time series
- Streaming data – Continuum of observations

Methods for Scaling to High Dimensions

Low Rank $\Sigma = \Lambda\Lambda' + \Sigma_0$

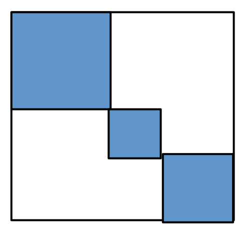




k << p

Low-dimensional embedding

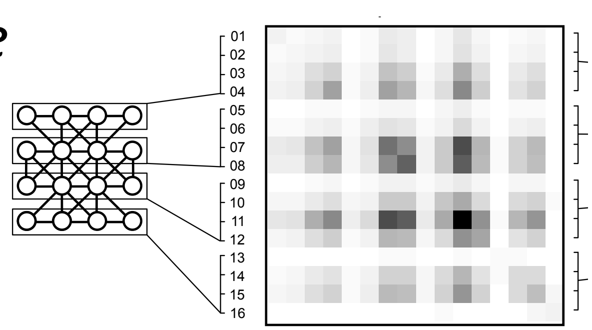
Σ *sparse*



Independent groups of nodes

Σ^{-1} *sparse*

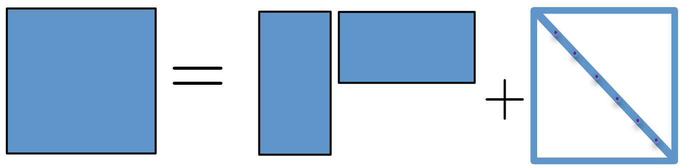
Gaussian
Graphical
Model

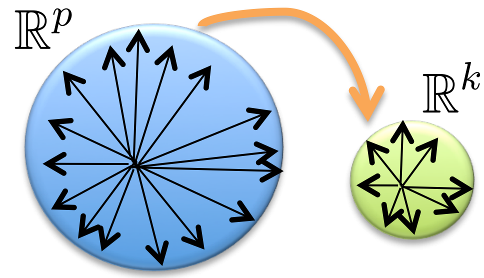


Zeros = no edge in graph,
Cond. ind. between nodes

Methods for Scaling to High Dimensions

Low Rank $\Sigma = \Lambda\Lambda' + \Sigma_0$





k << p

Low-dimensional embedding

Σ *sparse*

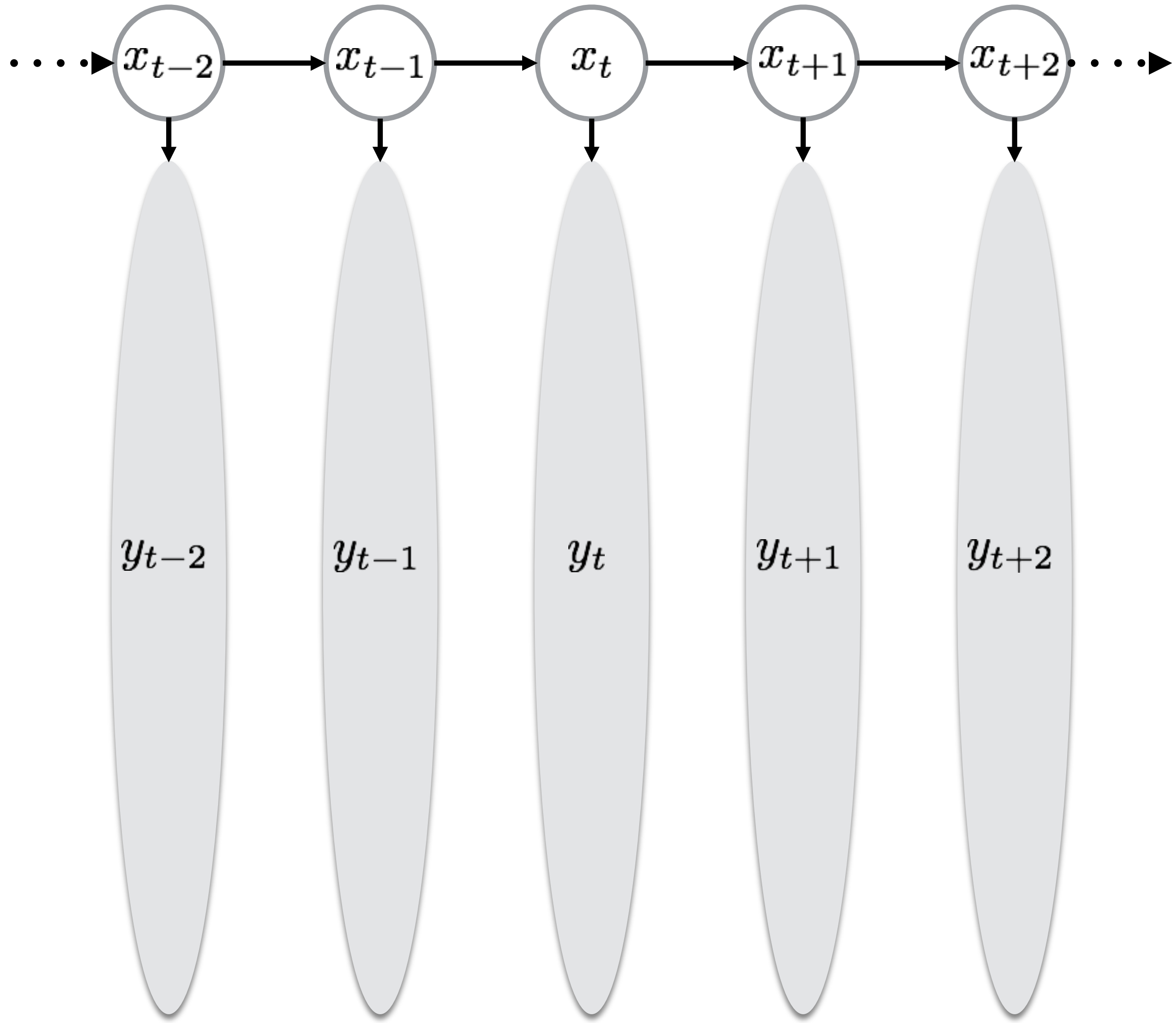
Independent groups of nodes

Σ^{-1} *sparse*

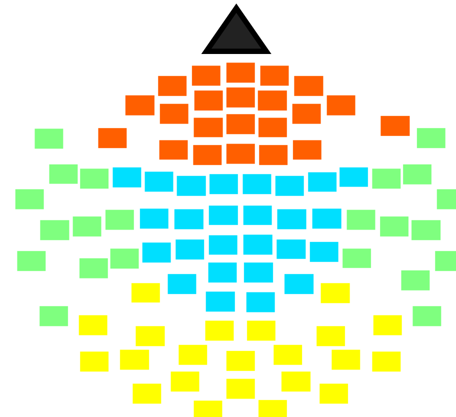
Gaussian Graphical Model

Zeros = no edge in graph, Cond. ind. between nodes

Modeling High-Dimensional Time Series



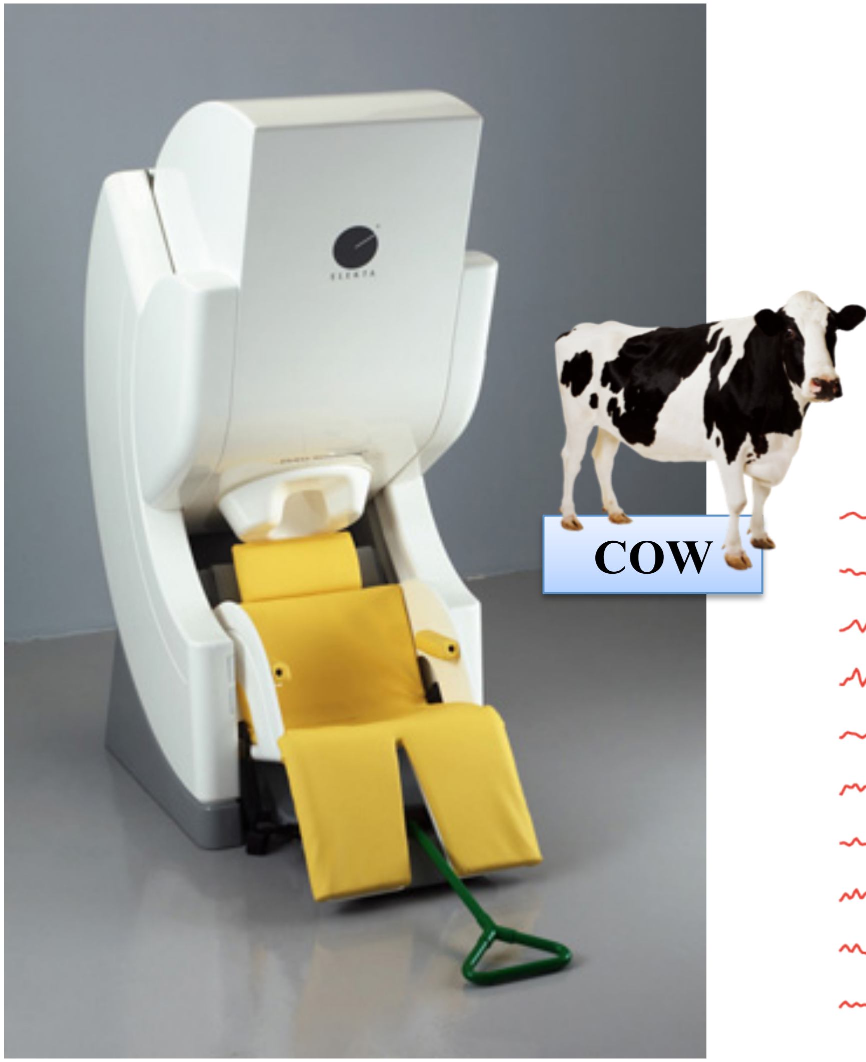
Magnetoencephalography (MEG)

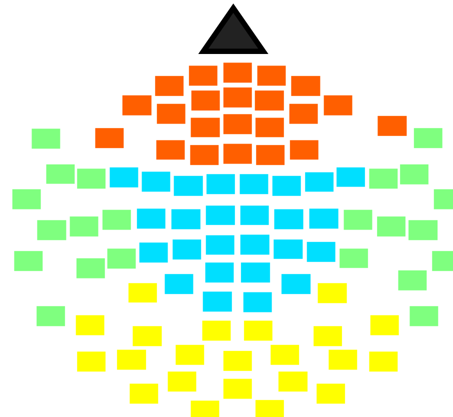


Helmet with
102 sensors

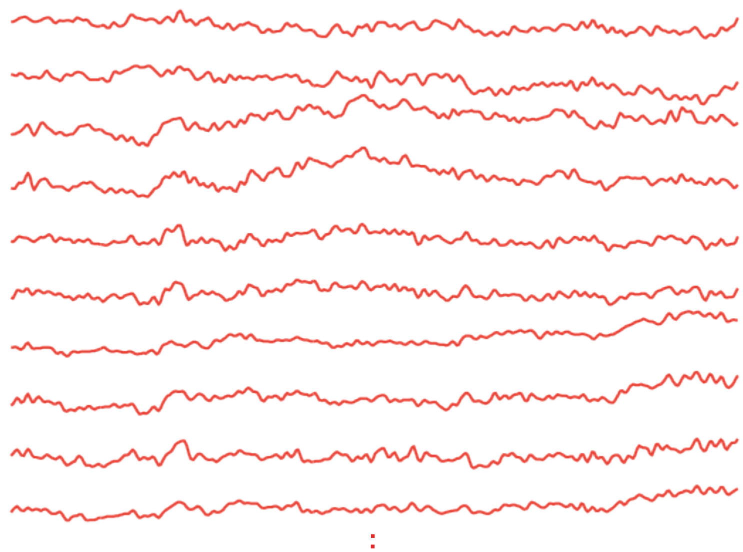
- How does the brain code concepts?
  - e.g. animals, food…

Magnetoencephalography (MEG)

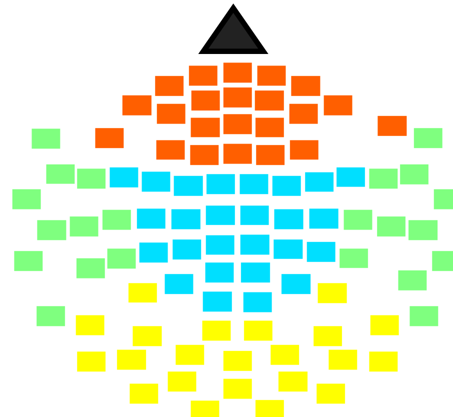




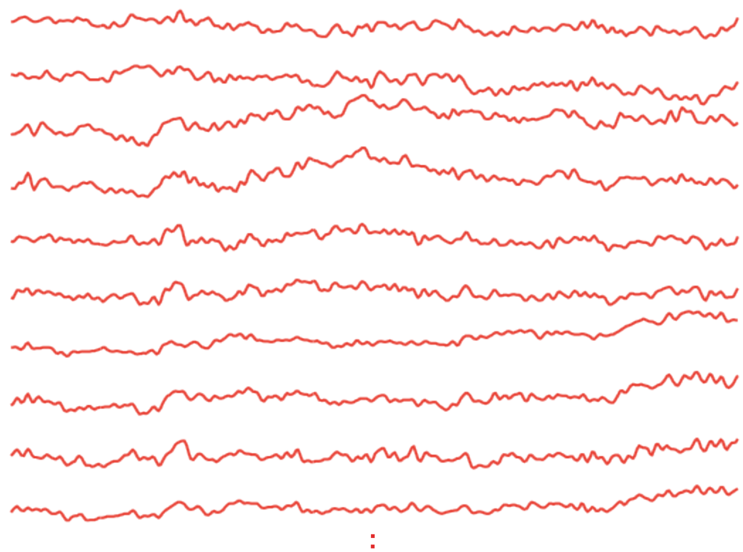
Helmet with
102 sensors



Magnetoencephalography (MEG)

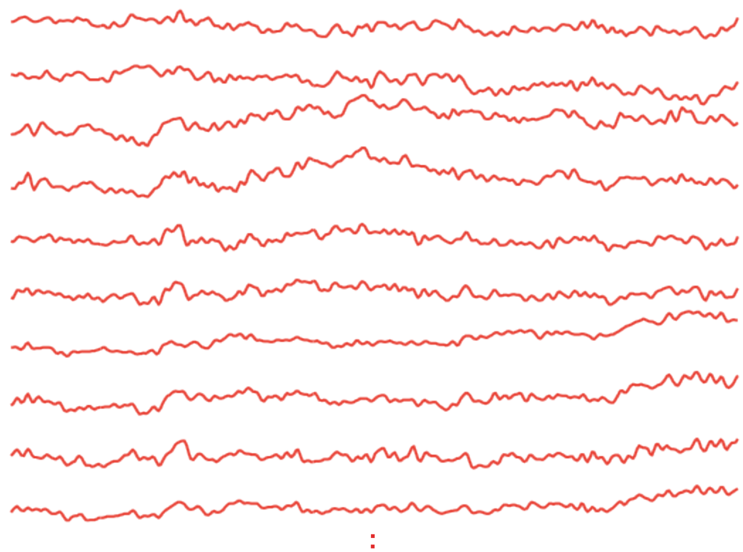


Helmet with
102 sensors



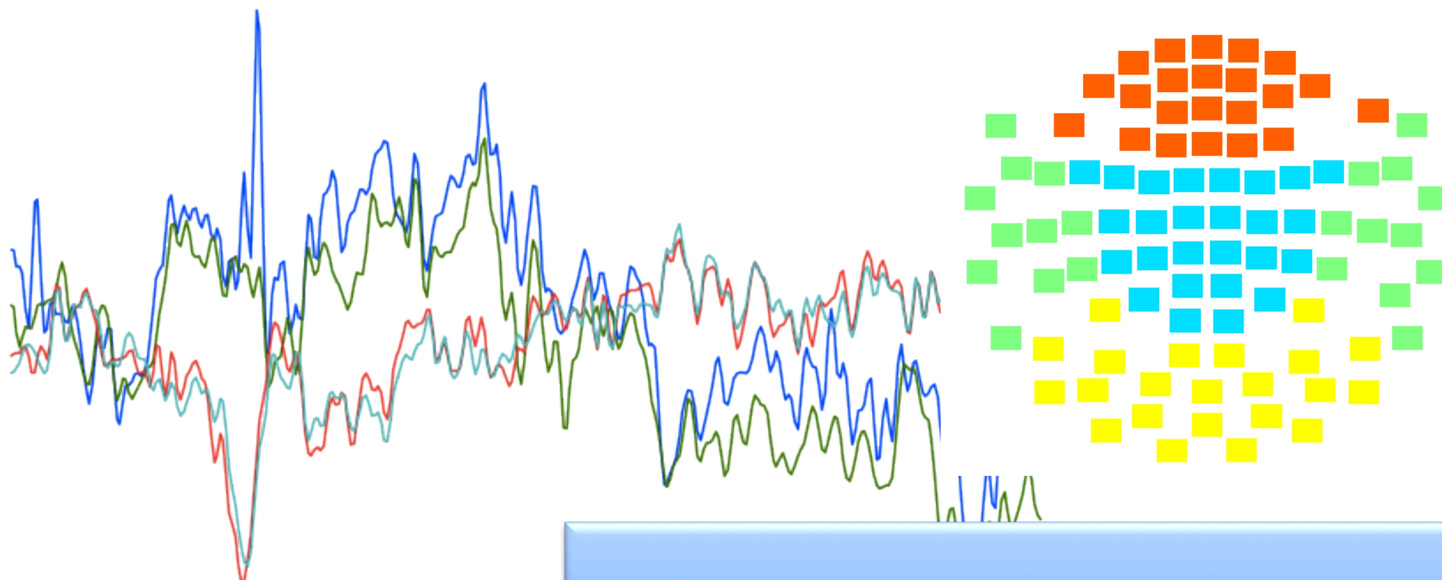
Magnetoencephalography (MEG)

- High dimensional
- Time-varying correlations
→ *Functional connectivity*



Coping with Dimensionality

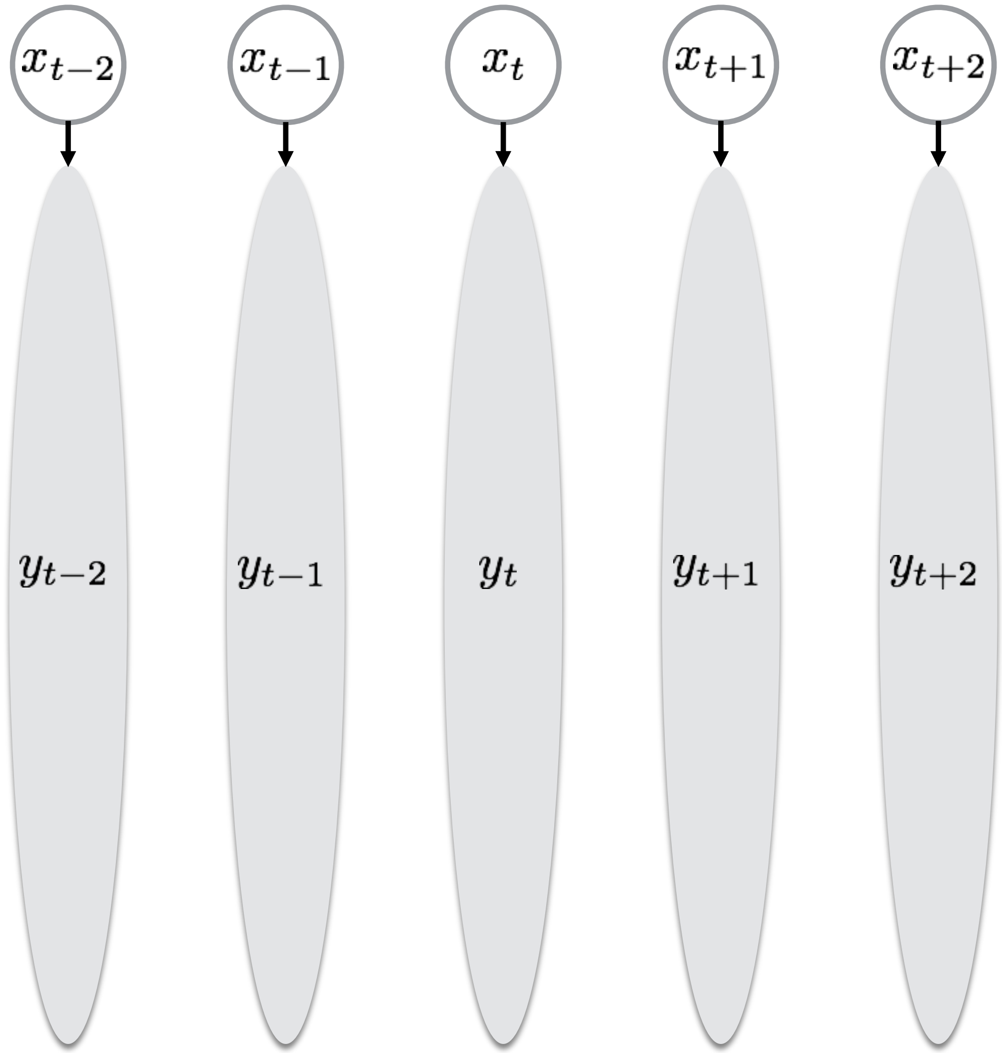
- Observation: Sensors are *redundant*



Dynamic Latent Factor Models

- Goal:
  - Harness *low-dimensional embedding* of dynamics

High-Dim i.i.d. Data



Latent Factor Model

- Assume normally distributed data

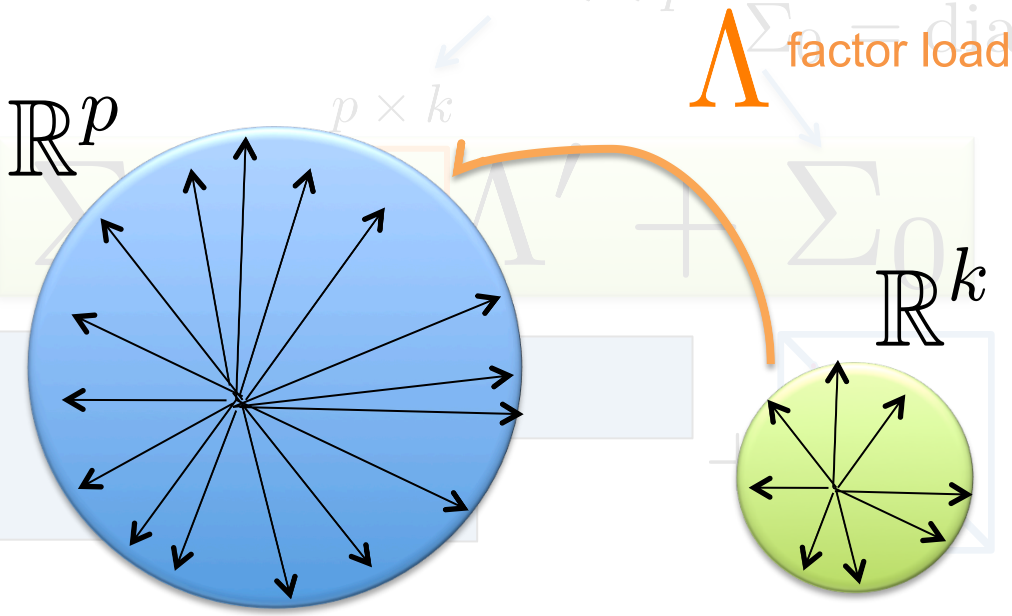
$$y_i \sim N_p(0, \Sigma)$$

Modeling statistical uncertainty in
low-dim subspace

$k << p$ $\quad \Sigma_0 = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2)$

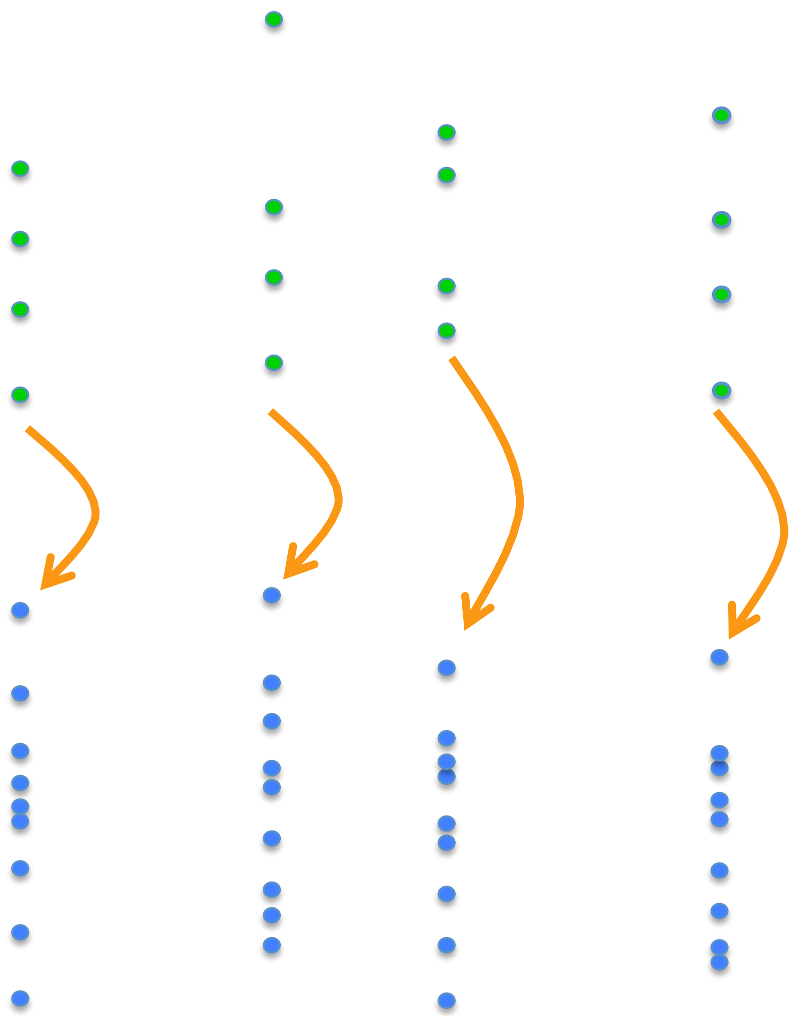
$p \times k$

$$\Sigma = \Lambda\Lambda' + \Sigma_0$$



- Number of parameters: $pk + p = p(k+1) << \frac{p(p+1)}{2}$

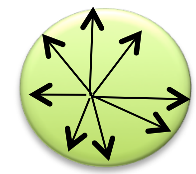
Latent Factor Model



latent factors

$\eta_i \sim N_k(0, I)$

$\mathbb{R}^k$

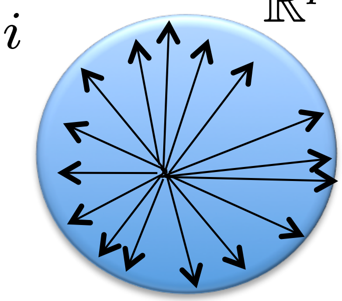


factor loadings

$\epsilon_i \sim N_p(0, \Sigma_0)$

$y_i = \boxed{\Lambda}\eta_i + \epsilon_i$

$\mathbb{R}^p$



Marginalize latent factors

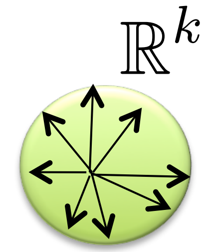
$$\boxed{\Sigma = \Lambda\Lambda' + \Sigma_0}$$

Derivation of Marginal Distribution

- Marginal mean:

$$\begin{aligned} E[y_i] &= E[\Lambda \eta_i + \epsilon_i] \\ &= \Lambda E[\eta_i] + E[\epsilon_i] = 0 \end{aligned}$$

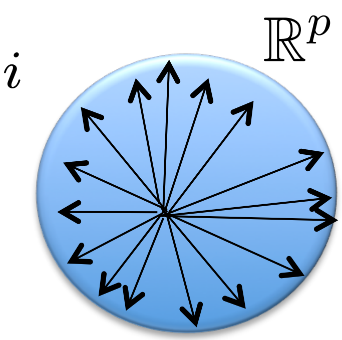
$$\eta_i \sim N_k(0, I)$$



- Marginal covariance:

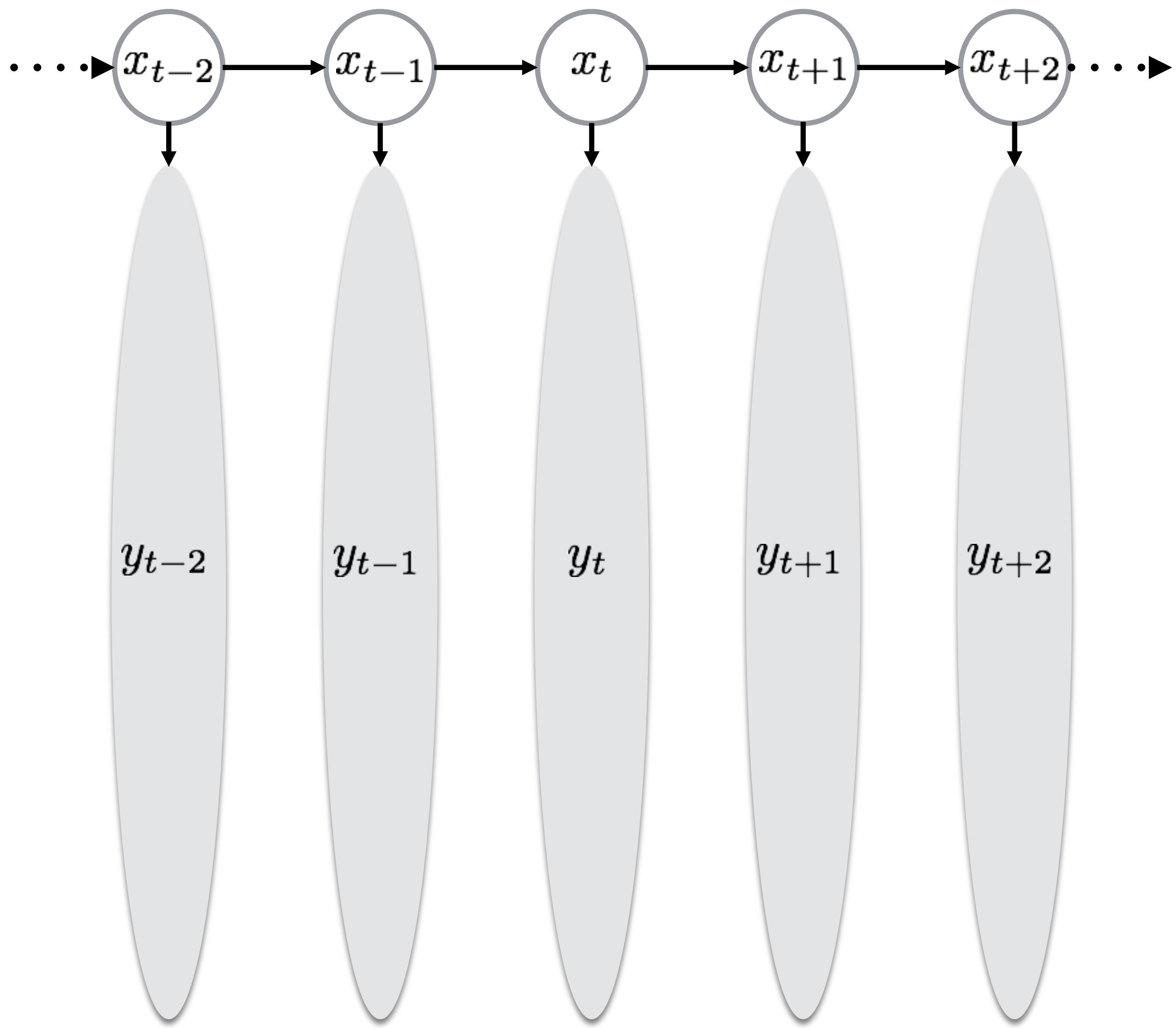
$$\epsilon_i \sim N_p(0, \Sigma_0)$$

$$y_i = \boxed{\Lambda} \eta_i + \epsilon_i$$

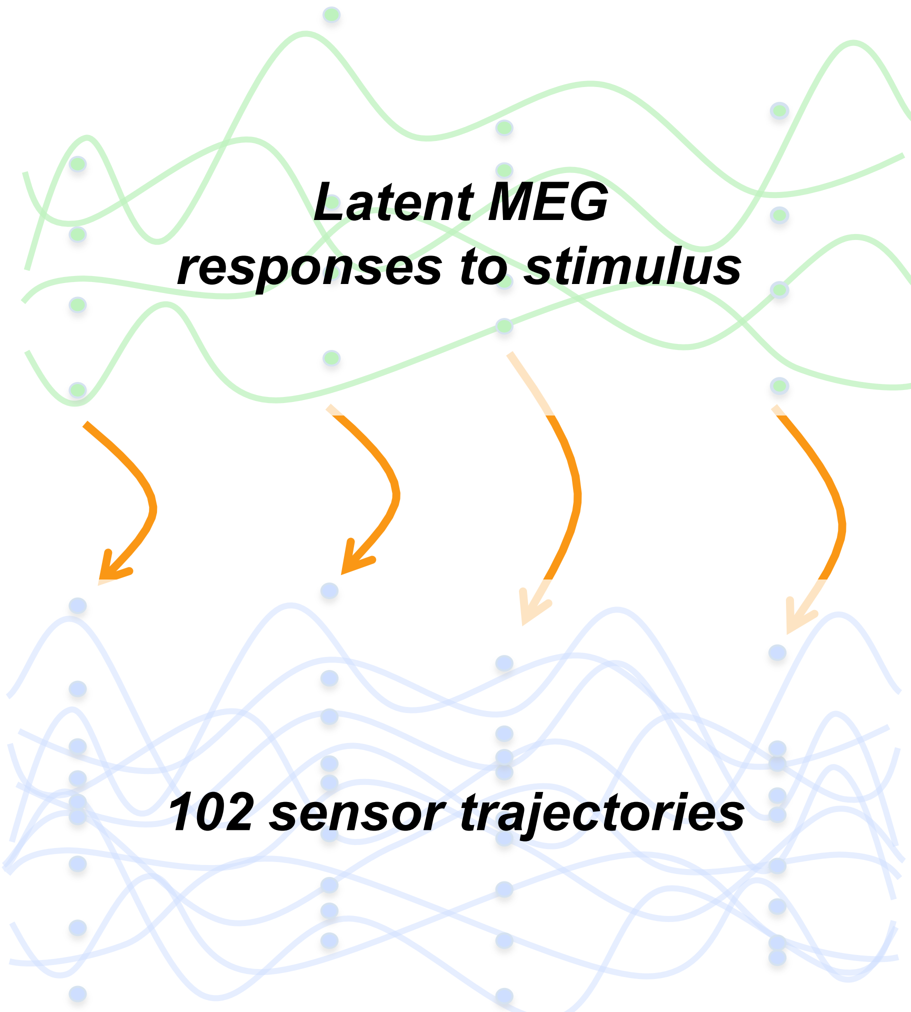


$$\begin{aligned} \mathrm{cov}(y_i) &= E[(y_i - E[y_i])(y_i - E[y_i])'] \\ &= E[y_i y_i'] \\ &= E[(\Lambda \eta_i + \epsilon_i)(\Lambda \eta_i + \epsilon_i)'] \\ &= \Lambda E[\eta_i \eta_i'] \Lambda' + 2\Lambda E[\eta_i \epsilon_i] + E[\epsilon_i \epsilon_i'] \\ &= \Lambda I \Lambda' + 0 + \Sigma_0 \\ &= \Lambda \Lambda' + \Sigma_0 \end{aligned}$$

Adding Dynamics



Dynamic Latent Factor Model



$$\eta_t = \boxed{\Phi \eta_{t-1}} + \nu_t \qquad N_k(0, I)$$

Evolution of latent factors

$$y_t = \boxed{\Lambda}\eta_t + \epsilon_t \qquad N_p(0, \Sigma_0)$$

Dynamic Latent Factor Model

- State-space model with low-dim state and high-dim observations
- Originally developed by Geweke (1977)
  - Other early work: Sargent and Sims (1977) Watson and Engle (1983)
- Very popular in econometrics
- Most foundational dynamic model of high-dimensional time series

$$\eta_t = \underbrace{\boxed{\Phi\eta_{t-1}}}_{\text{Evolution of latent factors}} + \overbrace{\nu_t}^{N_k(0, I)}$$

$$y_t = \boxed{\Lambda}\eta_t + \underbrace{\epsilon_t}_{N_p(0, \Sigma_0)}$$

Dynamic Latent Factor Model

- Assuming latent process is *stable*, marginally

$$y_t \sim N(0, \Sigma)$$

$$\Sigma = \Lambda \underbrace{\Sigma_\eta}_{\Gamma_\eta(0)} \Lambda' + \Sigma_0$$

- Though, still a dynamic process with lag covariance

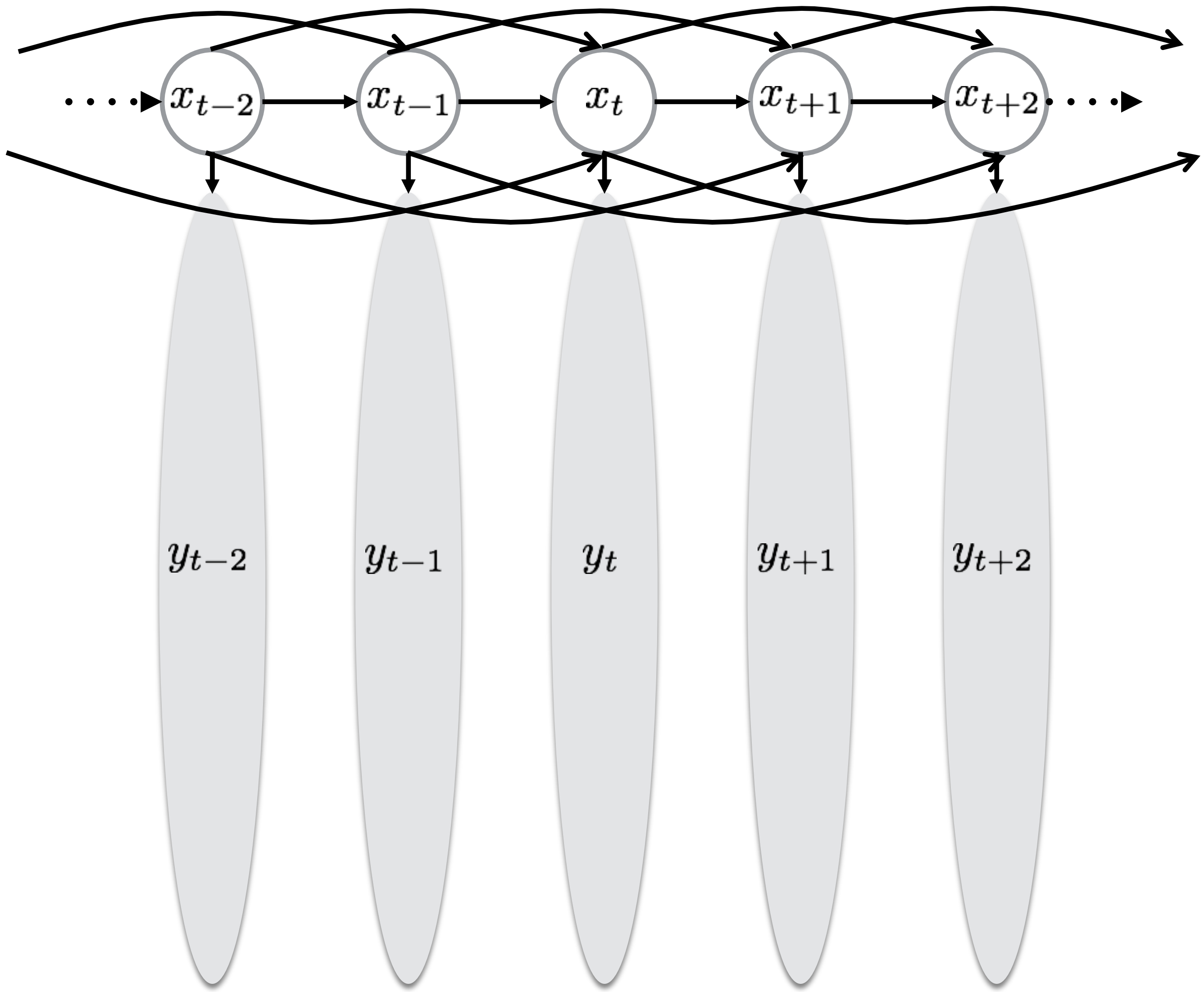
$$\begin{aligned}\Gamma_y(h) &= \mathrm{cov}(y_t, y_{t+h}) \\ &= \Lambda \Gamma_\eta(h) \Lambda' \qquad h > 0\end{aligned}$$

$$\eta_t = \boxed{\Phi \eta_{t-1}} + \underbrace{\nu_t}_{N_k(0, I)}$$

Evolution of latent factors

$$y_t = \boxed{\Lambda} \eta_t + \underbrace{\epsilon_t}_{N_p(0, \Sigma_0)}$$

Adding *Complex* Dynamics



Semiparametric Factor Model

- Can consider a *nonparametric latent factor process*
  - Gaussian processes
  - More on next slides…
- For a regression setting, looks very similar to Teh, Seeger, & Jordan 2004
  - *x* an arbitrary covariate, not necessarily time

Time index

$$\eta_t = \boxed{\psi(x_t)} + \underbrace{\nu_t}_{N_k(0, I)}$$

Nonparametric evolution of latent factors

$$y_t = \boxed{\Lambda}\eta_t + \underbrace{\epsilon_t}_{N_p(0, \Sigma_0)}$$

Gaussian Processes

REVIEW

- Distribution on functions
  - $f \sim GP(m, \kappa)$
    - m: mean function
    - κ: covariance function

$$\Updownarrow$$

  - $p(f(x_1), \ldots, f(x_n)) \sim N_n(\mu, K)$
    - $\mu = [m(x_1), \ldots, m(x_n)]$
    - $K_{ij} = \kappa(x_i, x_j)$

- Idea: If x_i, x_j are similar according to the kernel, then $f(x_i)$ is similar to $f(x_j)$

Gaussian Processes

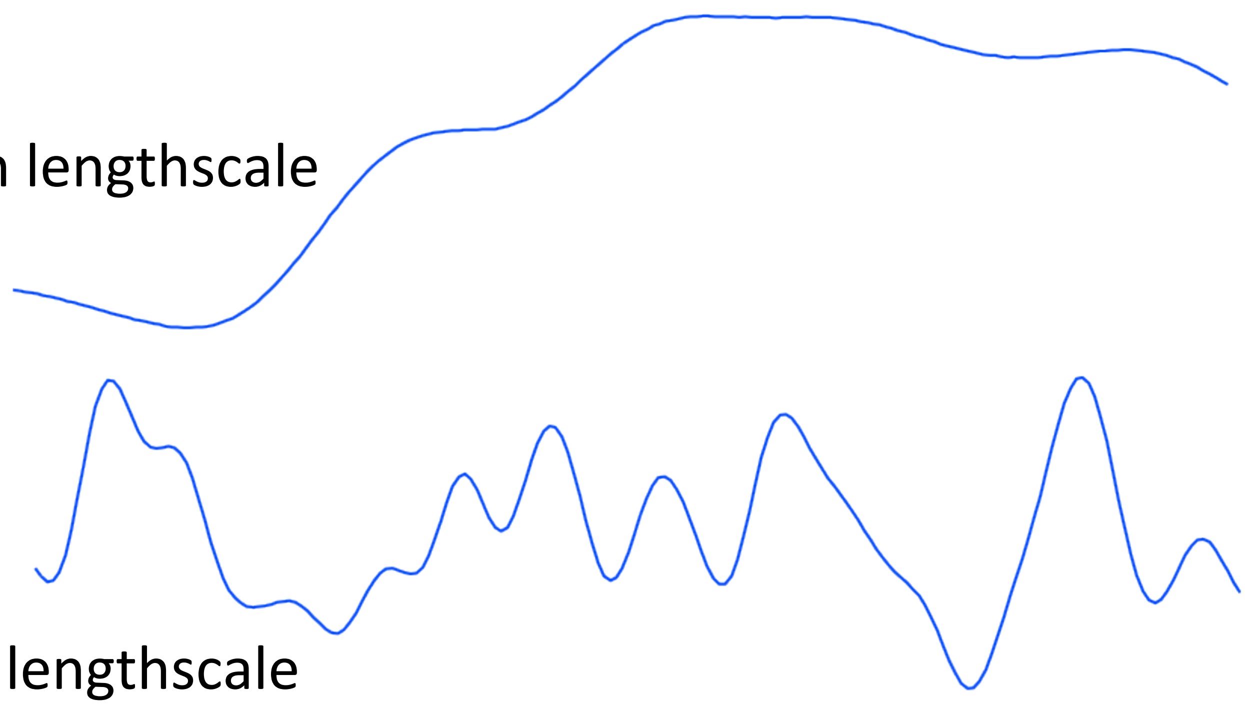
REVIEW

κ: covariance function

f ~ GP(m,κ)

$$\kappa(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}(x - x')^2\right)$$

High lengthscale



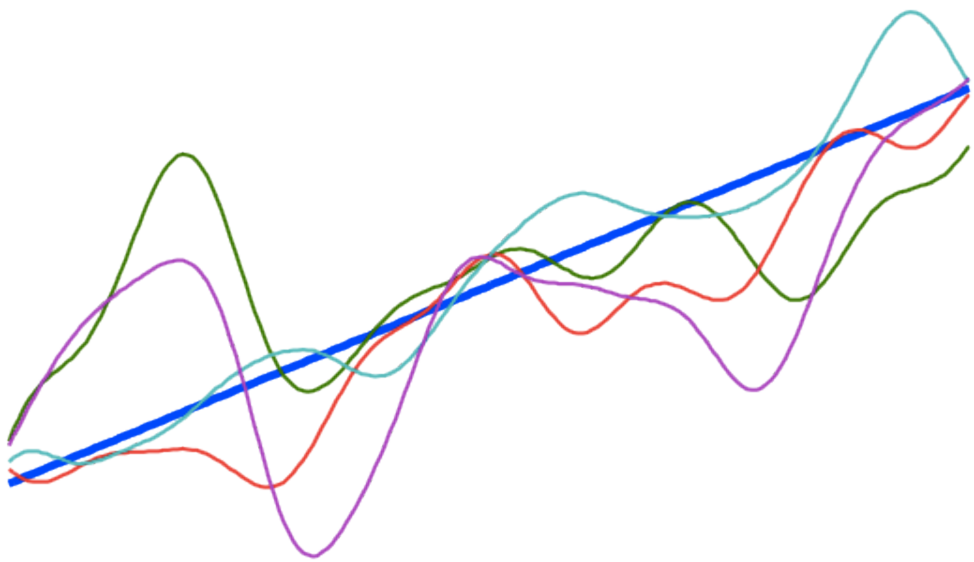
Low lengthscale

Gaussian Processes

REVIEW

m: mean function

$f \sim GP(m, \kappa)$

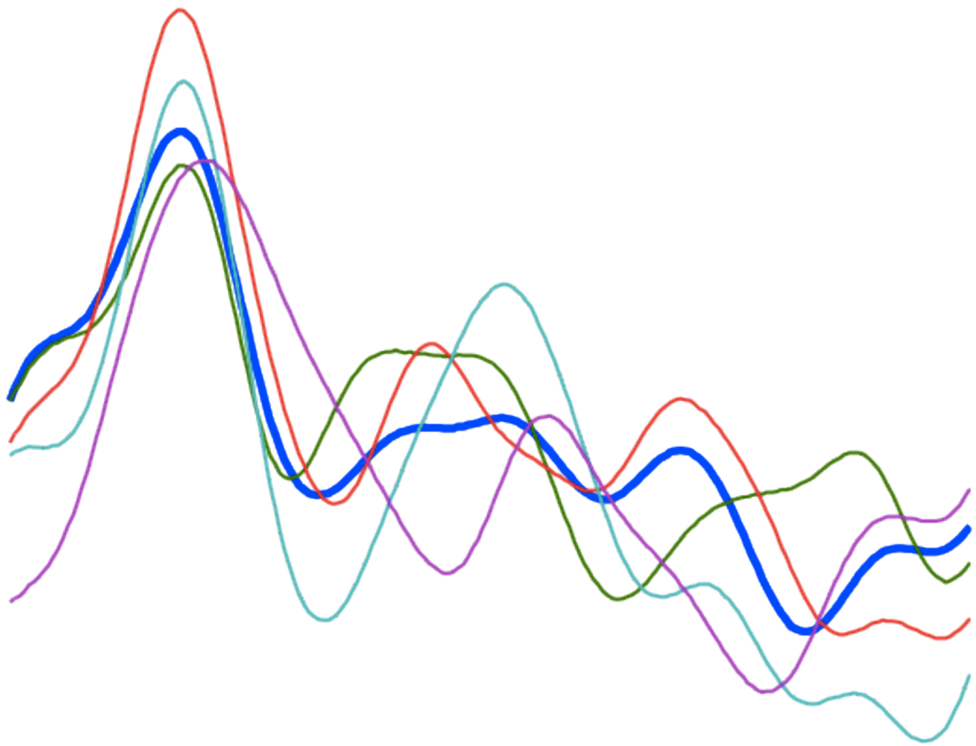


Gaussian Processes

REVIEW

m: mean function

$f \sim GP(m, \kappa)$

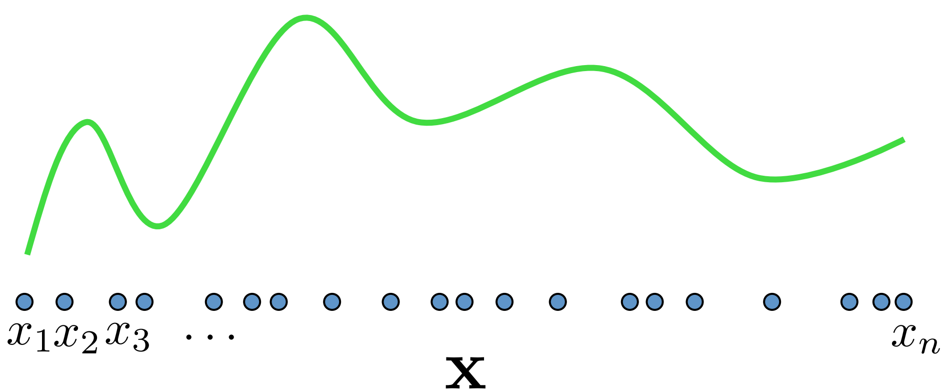


Induced Multivariate Gaussian

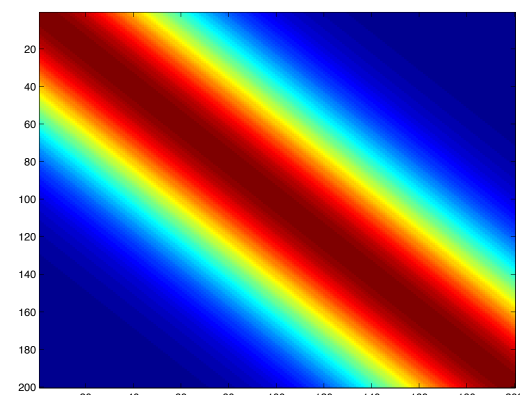
REVIEW

- Evaluating the GP-distributed function at any set of locations (x_1,…,x_n), we have

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \sim N(\mu, K)$$



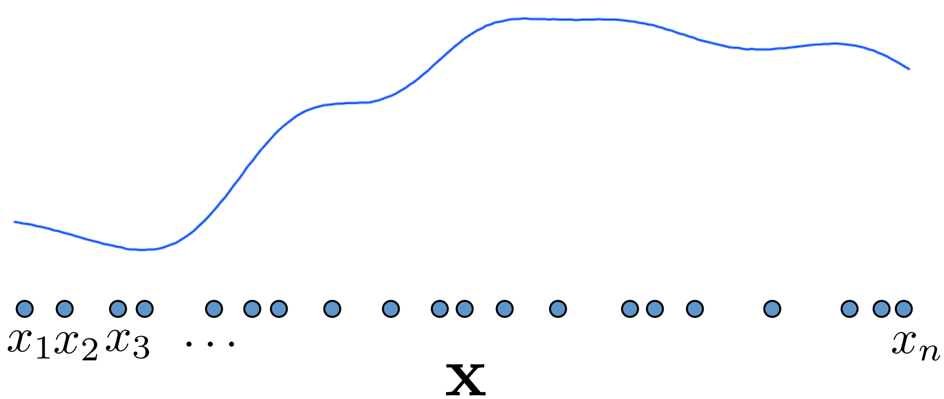
$$K =$$

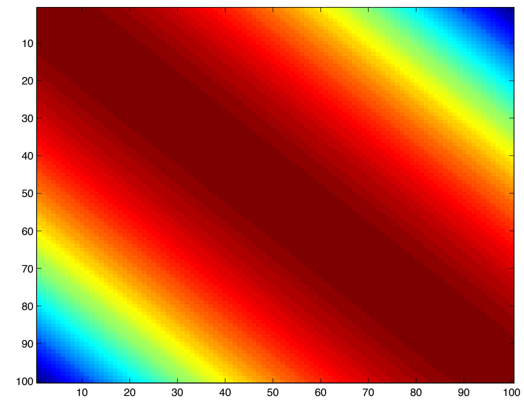


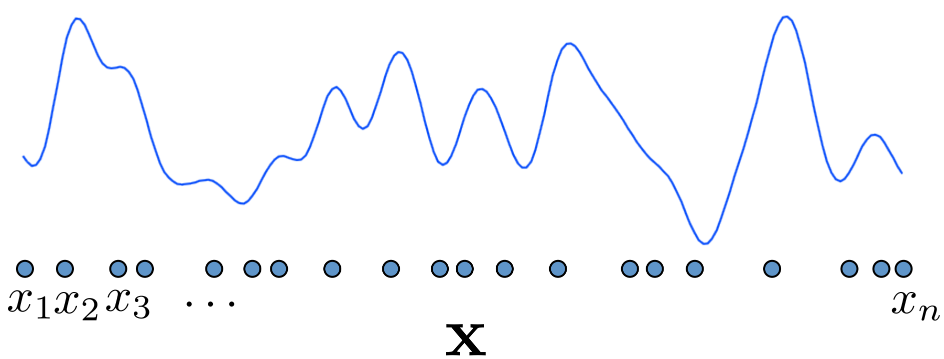
Induced Multivariate Gaussian

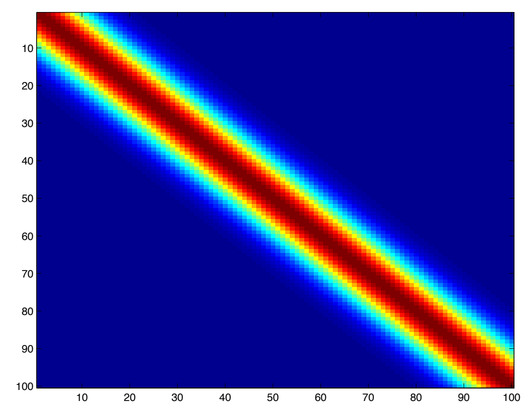
REVIEW

- Comparing length-scales:





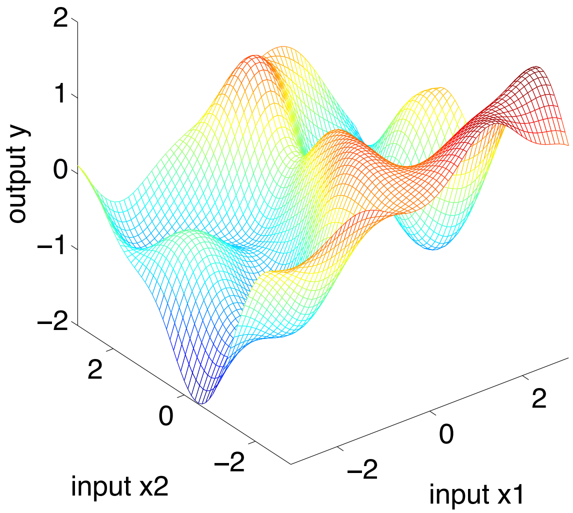


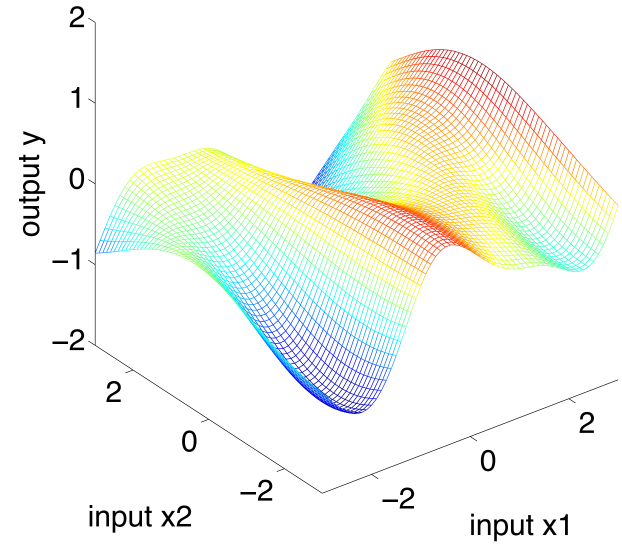


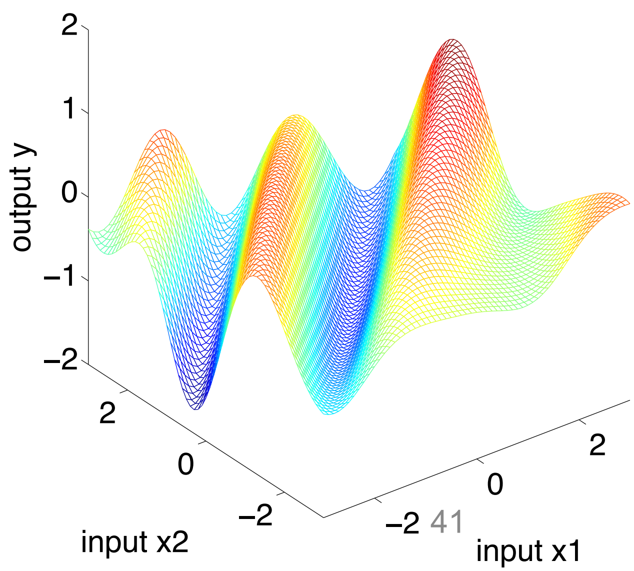
2D Gaussian Processes

REVIEW

$$\kappa(x_p, x'_q) = \ {}^2_f \exp\left(-\frac{1}{2}(x_p - x'_q)^T M (x_p - x'_q)\right)$$

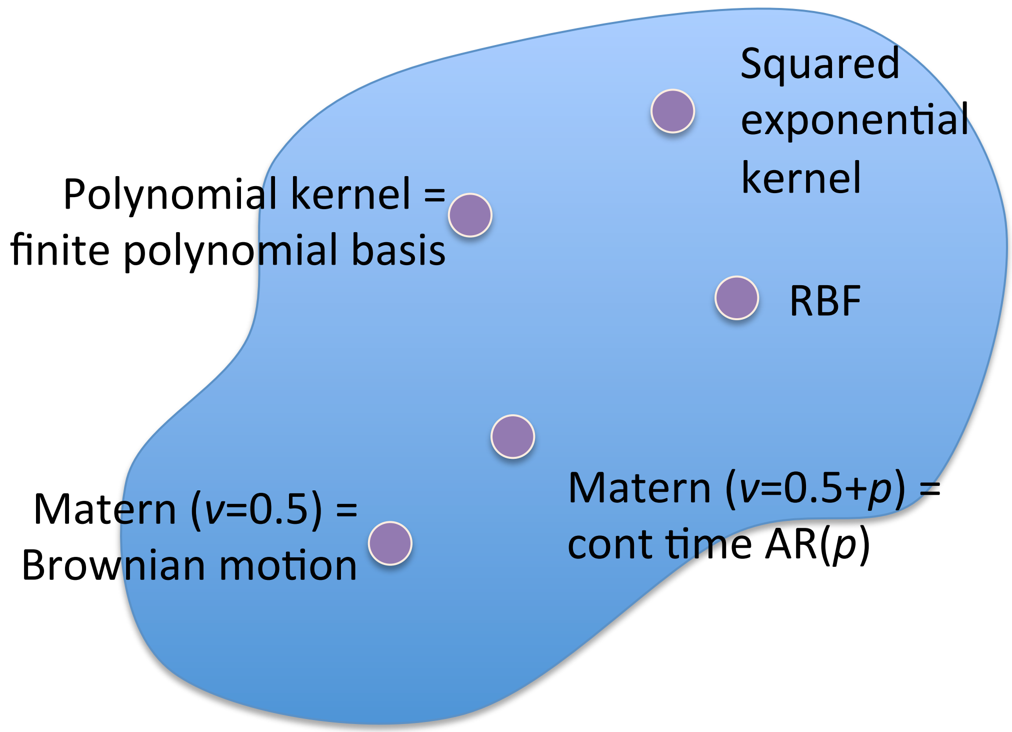






Family of Gaussian Processes

REVIEW



GPs for Regression

REVIEW

- Start with noise-free scenario: directly observe the function

- Training data $\mathcal{D} = \{(x_i, f_i), i = 1, \ldots, n\}$
- Test data locations X^* → predict *f**

- Jointly, we have

$\kappa(X, X^*)$

$$\begin{pmatrix} f \\ f^* \end{pmatrix} \sim N\left(\begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{pmatrix} K & K_* \\ K_*^T & K_{**} \end{pmatrix}\right)$$

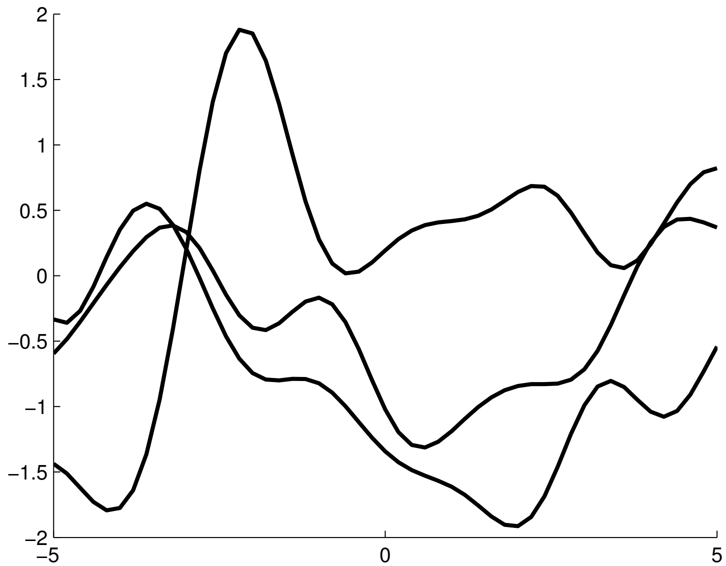
$\kappa(X^*, X^*)$

- Therefore,

$$p(f^* \mid X^*, X, f) = N(f^* \mid \mu_* + K_*' K^{-1}(f - \mu), K_{**} - K_*' K^{-1} K_*)$$

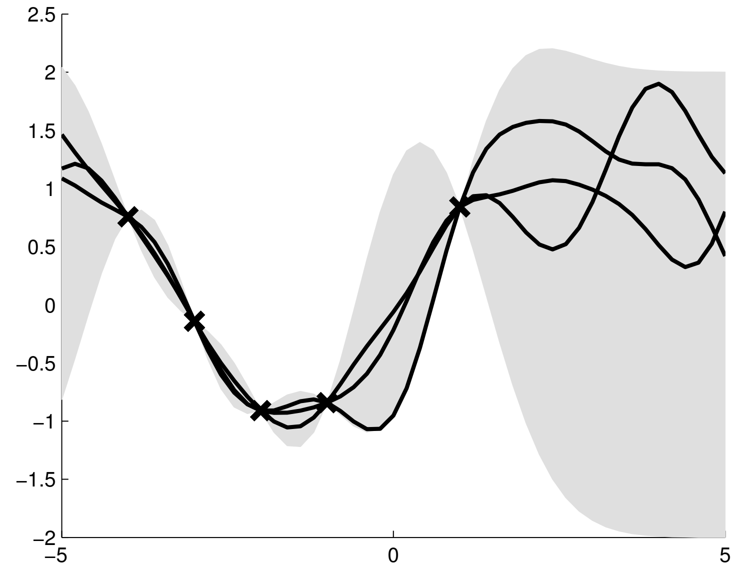
1D Noise-Free Example

REVIEW



Samples from Prior

$$\kappa(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}(x - x')^2\right)$$



Posterior Given 5 Noise-Free Observations

- Interpolator, where uncertainty increases with distance
- Useful as a computationally cheap proxy for a complex simulator
  – Examine effect of simulator params on GP predictions instead of doing expensive runs of the simulator

GPs for Regression

REVIEW

- Noisy scenario: observe a noisy version of underlying function

$$y = f(x) + \epsilon \quad \epsilon \sim N(0, \sigma_y^2)$$

  – Not required to interpolate, just come "close" to observed data

$$\mathrm{cov}(y|X) = \mathrm{cov}(f) + \mathrm{cov}(e) = K + \sigma_y^2 I \triangleq K_y$$

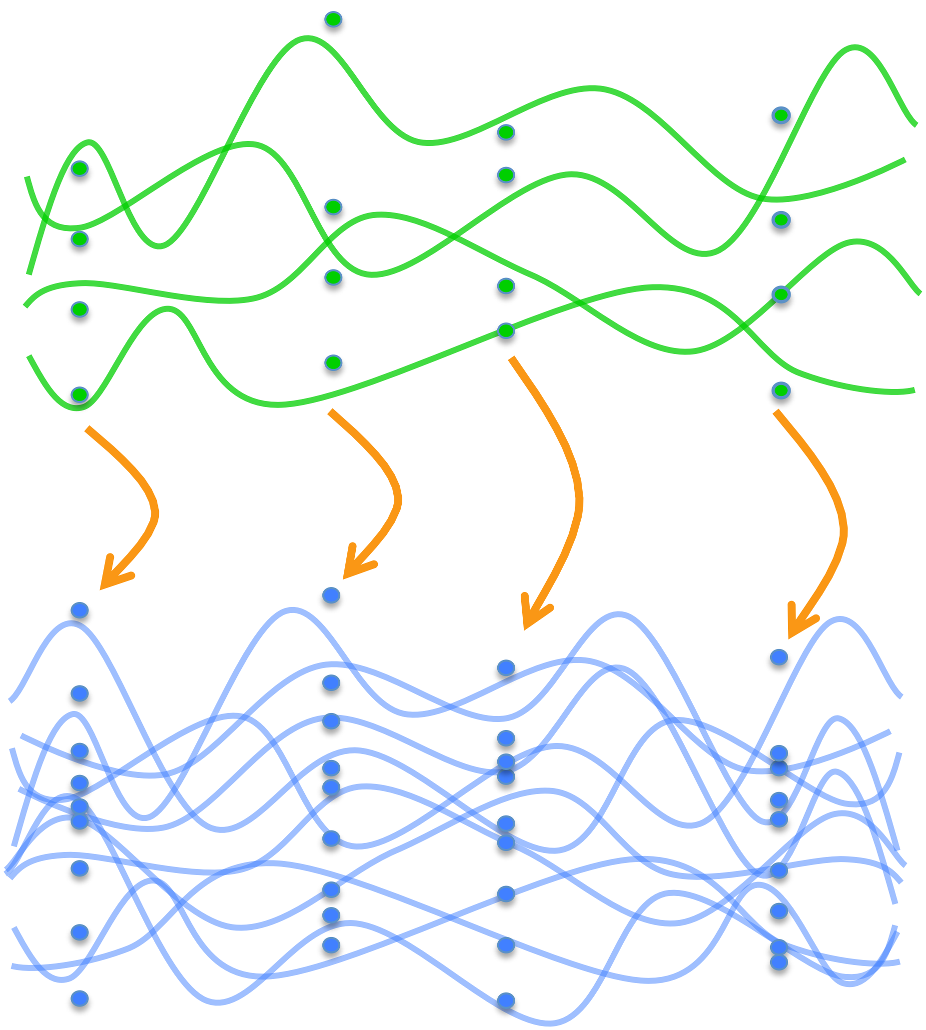
- Training data $\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, n\}$
- Test data locations X^* → predict *f**

- Jointly, we have $\begin{pmatrix} y \\ f^* \end{pmatrix} \sim N\left(0, \begin{pmatrix} K_y & K_* \\ K_*^T & K_{**} \end{pmatrix}\right)$

- Therefore,

$$p(f^* \mid X^*, X, y) = N(f^* \mid K_*' K_y^{-1} y, K_{**} - K_*' K_y^{-1} K_*)$$

Dynamic Latent Factor Model



$N_k(0, I)$

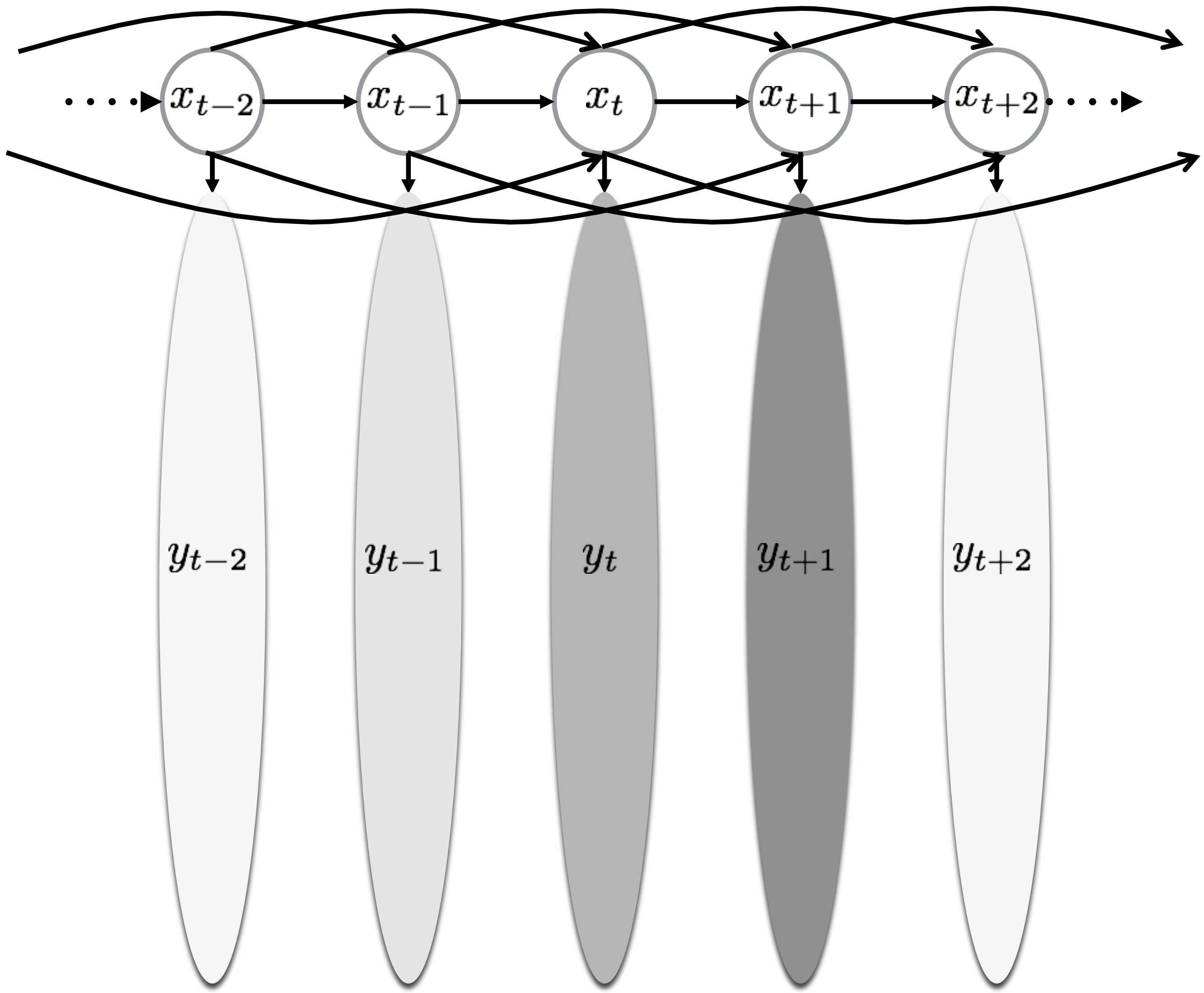
$$\eta_t = \boxed{f(\eta_{1:t-1})} + \nu_t$$

Evolution of latent factors

$$y_t = \boxed{\Lambda}\eta_t + \epsilon_t$$

$N_p(0, \Sigma_0)$

Capturing Changing Correlations

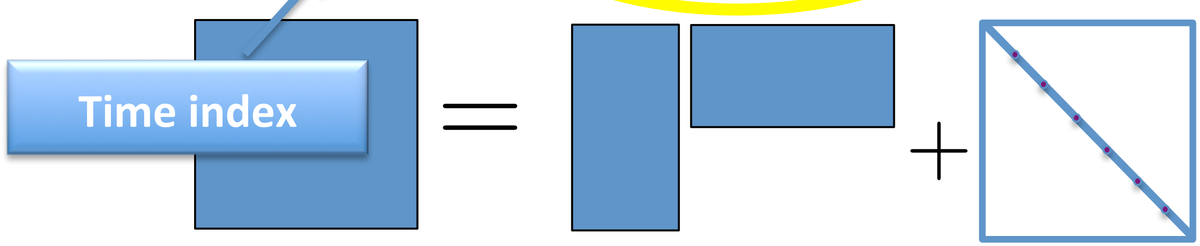


Capturing Changing Correlations

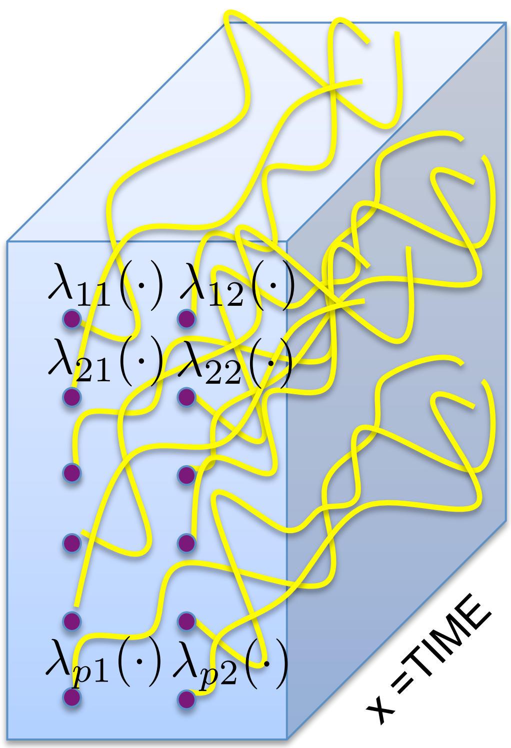
Observation:

1. Sensors are *redundant*
2. Correlation pattern *changes* with time

$$\Sigma(x) = \overset{p \times k}{\Lambda(x)}\Lambda(x)' + \Sigma_0$$



Low-Rank Covariance Evolution



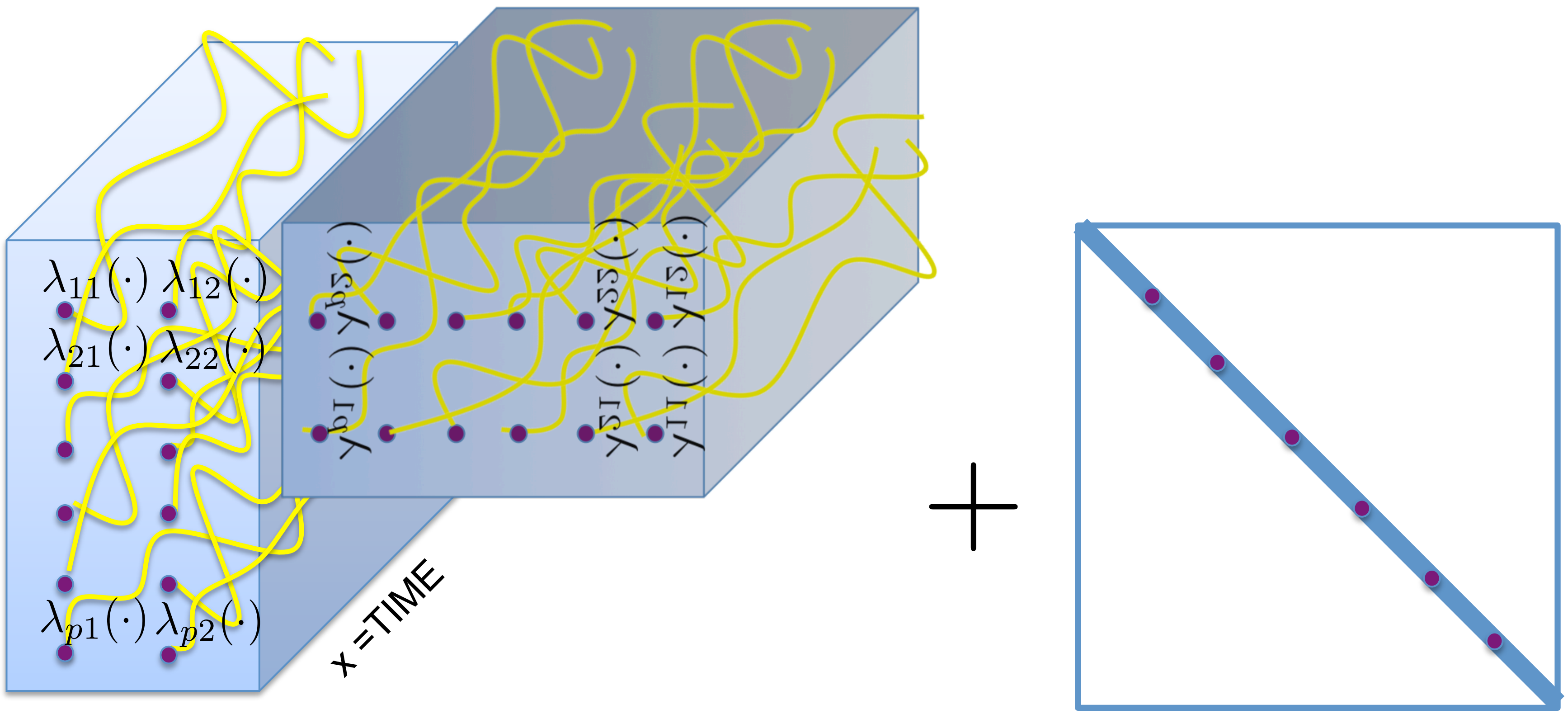
$p \times k$ array of processes over time

$k << p$

$$\Sigma(x) = \underbrace{\boxed{\Lambda(x)}}_{p \times k}\Lambda(x)' + \Sigma_0$$

Fox and Dunson, arXiv 2011.
Related model without low-rank structure: Wilson and Ghahramani, UAI 2011.

Low-Rank Covariance Evolution

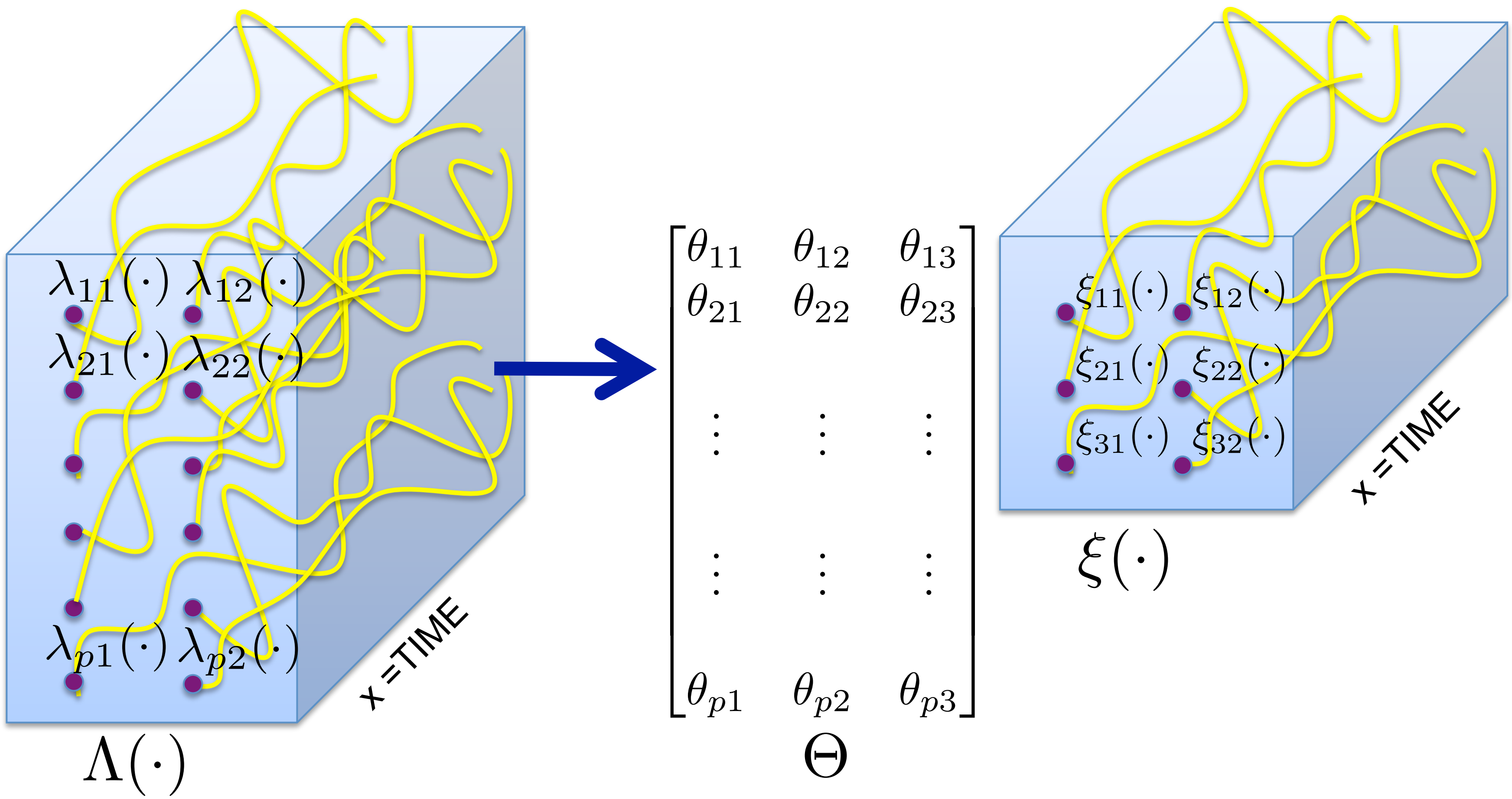


$$\Sigma(x) = \boxed{\Lambda(x)}\boxed{\Lambda(x)'} + \boxed{\Sigma_0}$$

Fox and Dunson, arXiv 2011.

Related model without low-rank structure: Wilson and Ghahramani, UAI 2011.

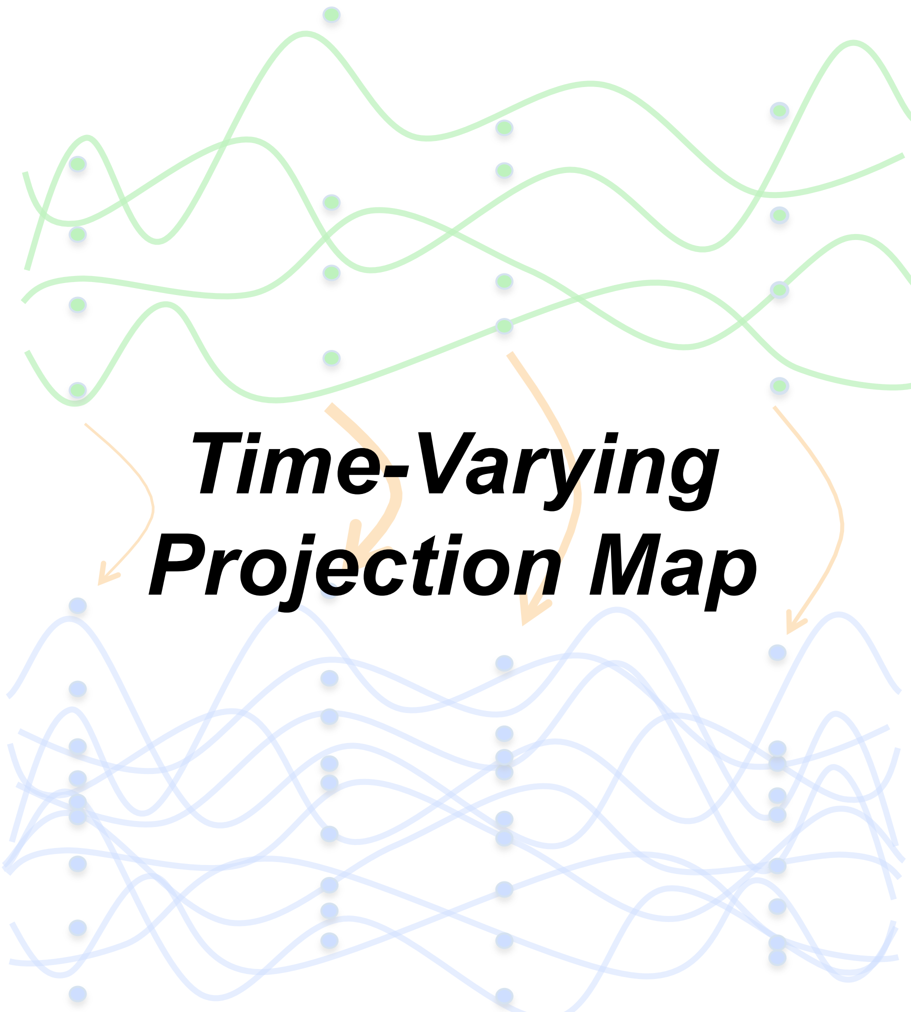
One Step Further...

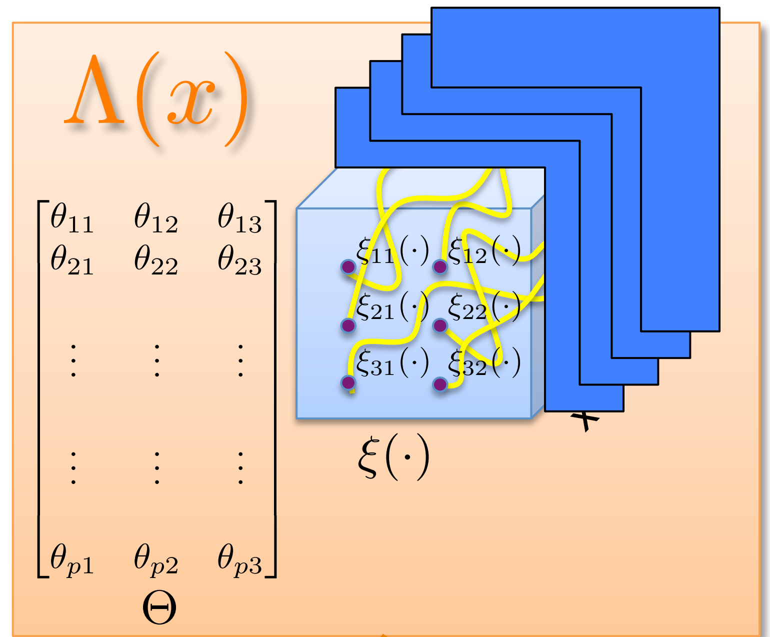


$$\Sigma(x) = \Theta\xi(x)\xi(x)'\Theta' + \Sigma_0$$

Fox and Dunson, arXiv 2011.

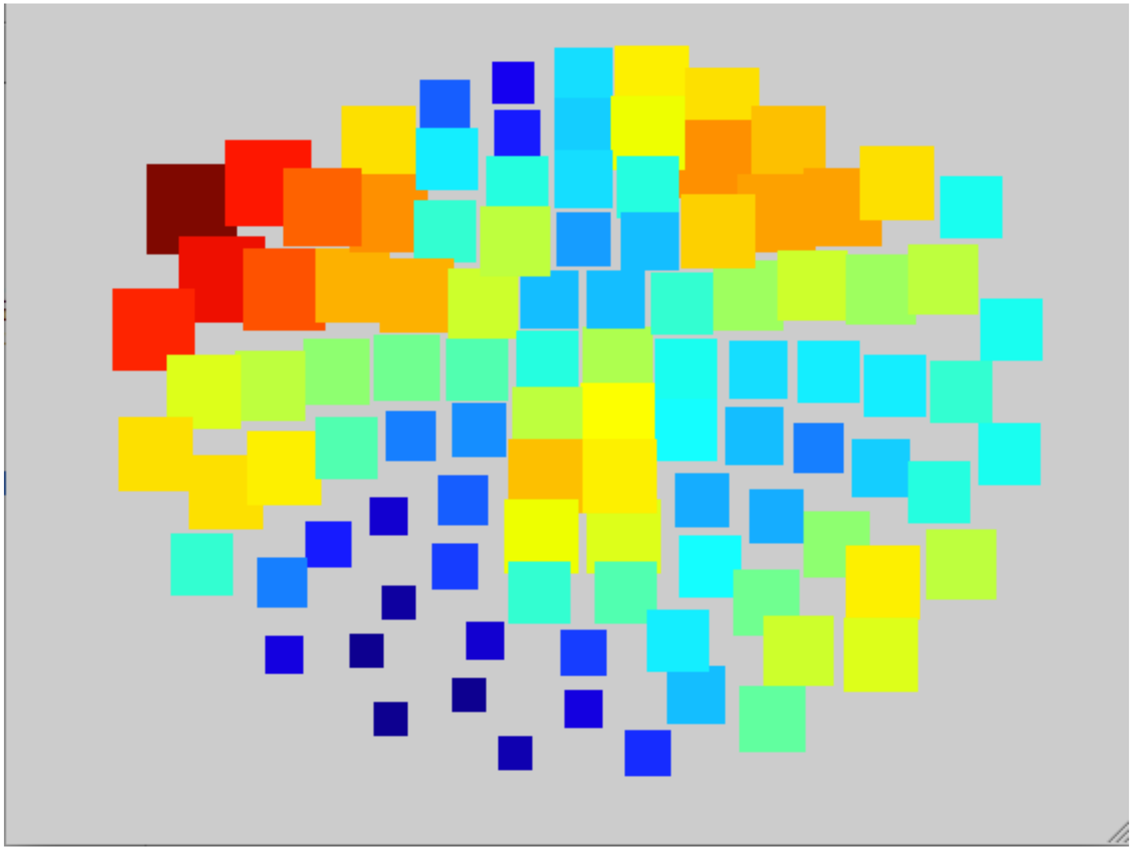
Interpretation as Dynamic LFM



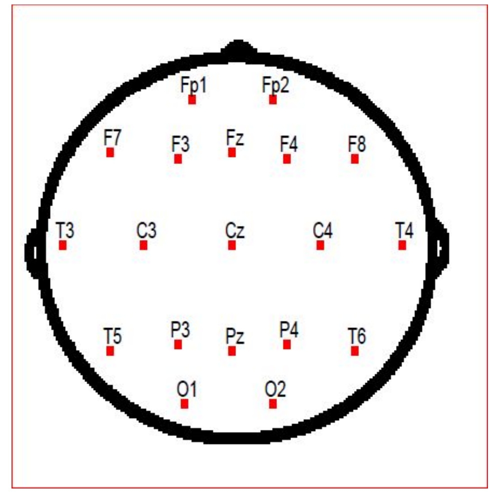


$$y_t = \boxed{\Theta\xi(x_t)}\eta_t + N_p(0, \Sigma_0)$$

Changing Correlations – MEG

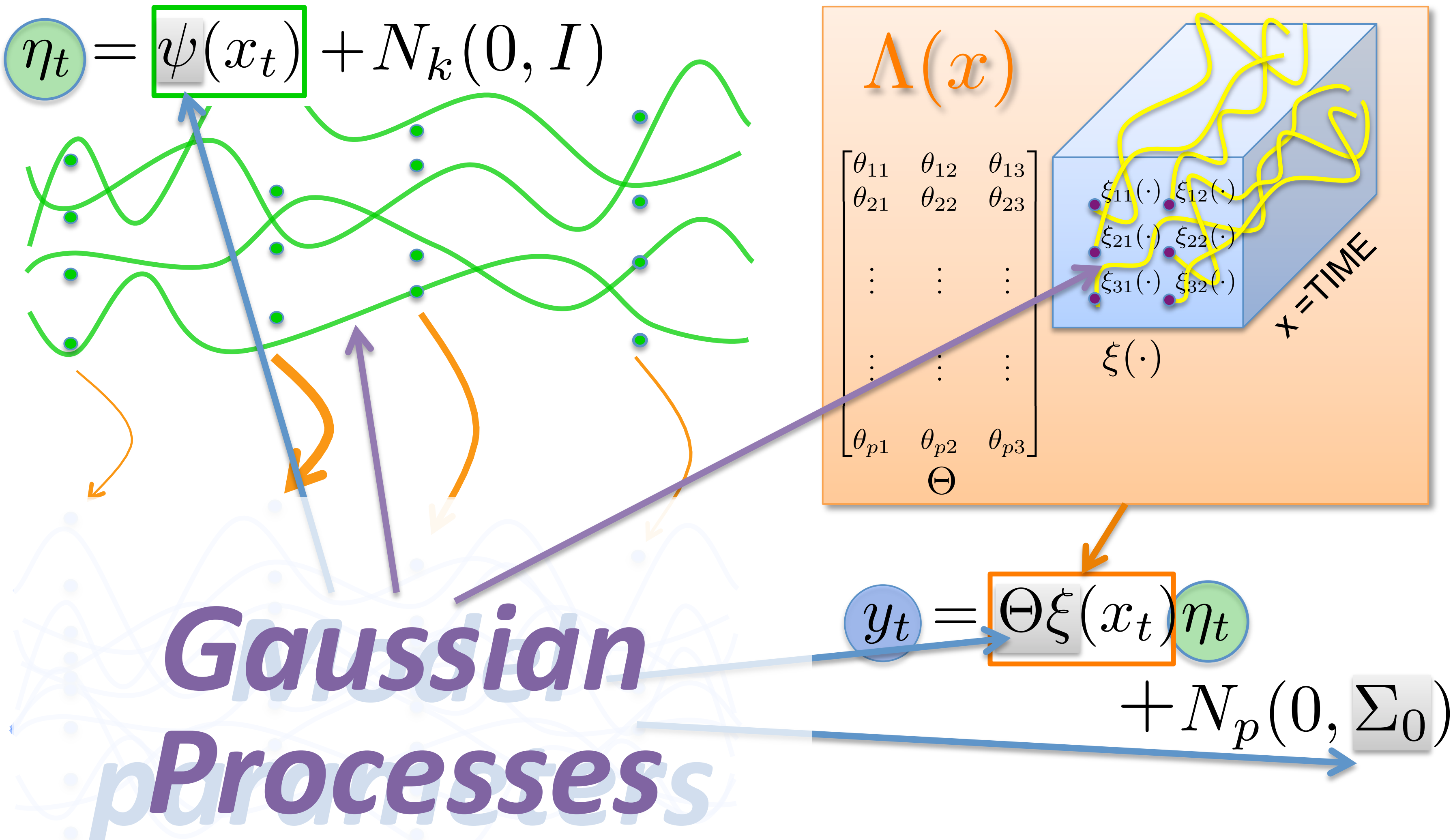


102 sensors:



Correlations between sensors change with processing of word "kick"

Prior Specification



Data Collection

- 4 word categories, 5 words per category

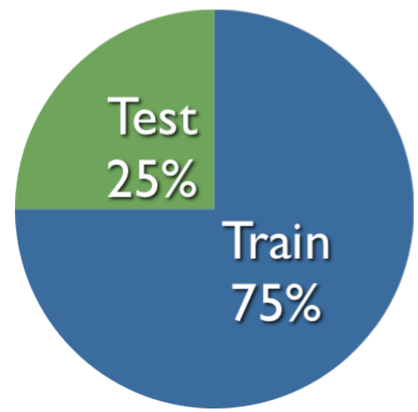
Animals

Food

Tools

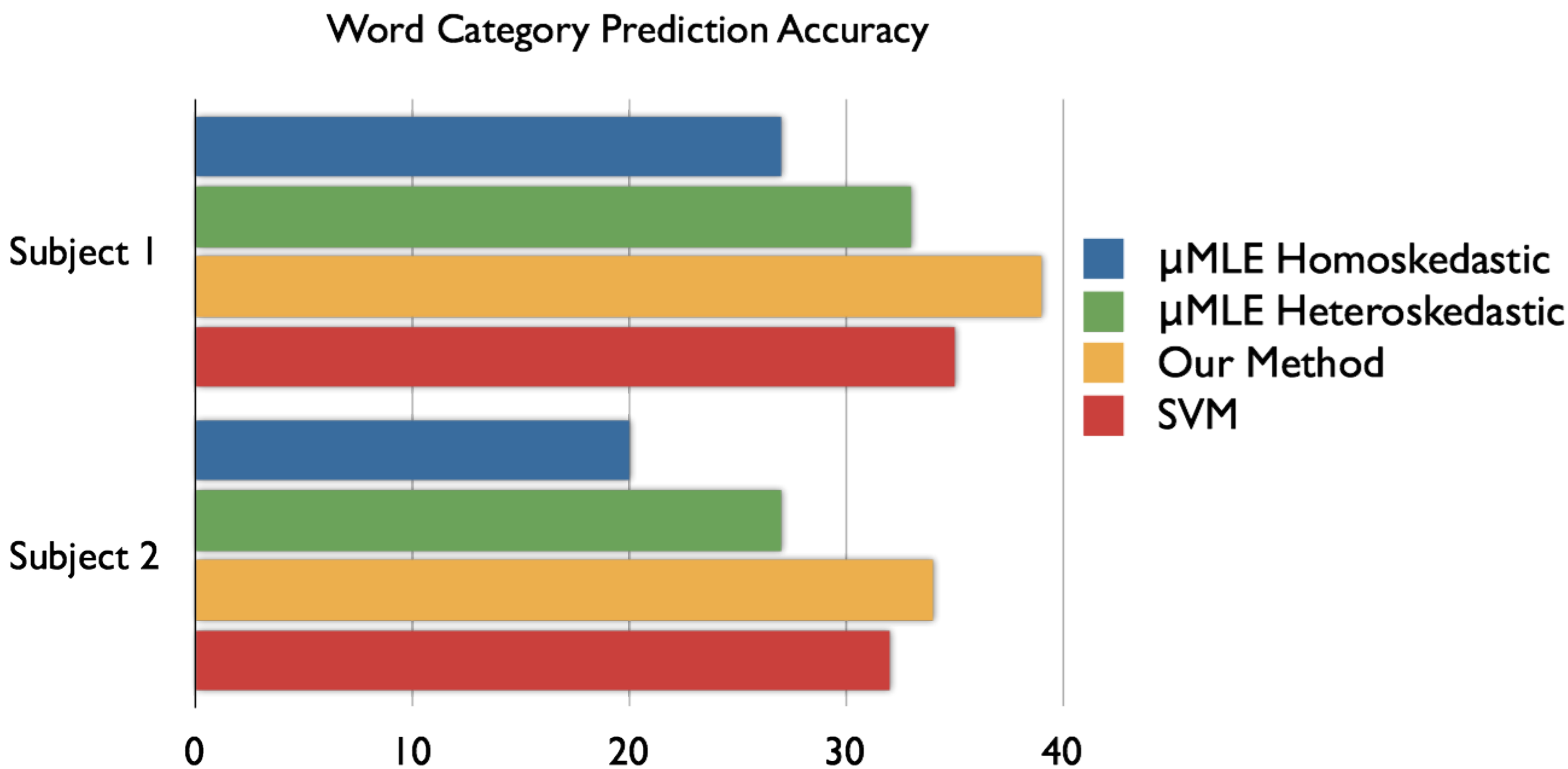
Buildings

- 20 repetitions per word (400 total)
  - 15 train/word (300 total)
  - 5 test/word (100 total)

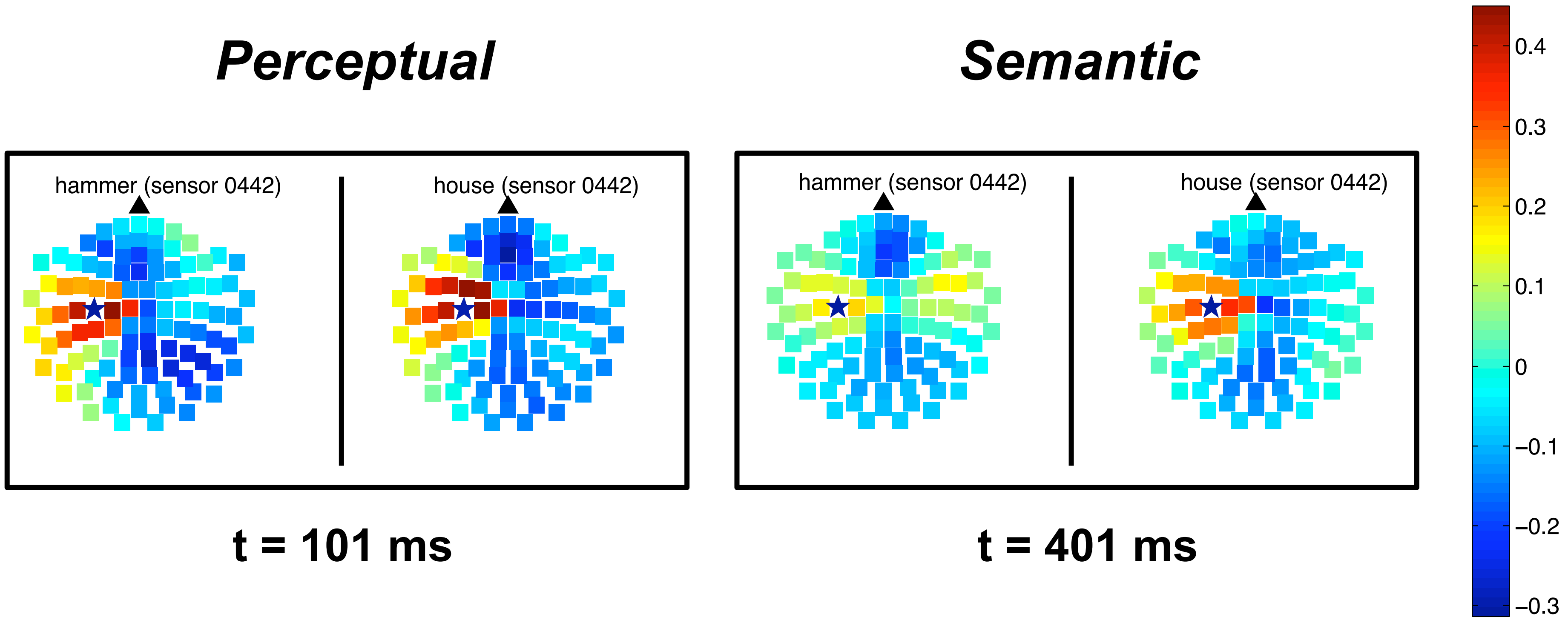


Fyshe, Fox, Dunson, and Mitchell, AISTATS 2012.

Classification Performance



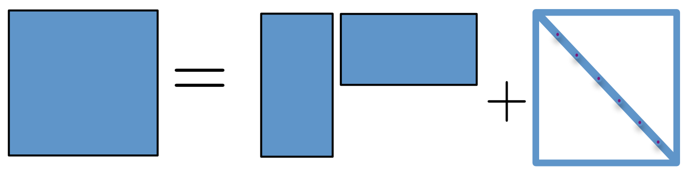
Perceptual vs. Semantic Correlations

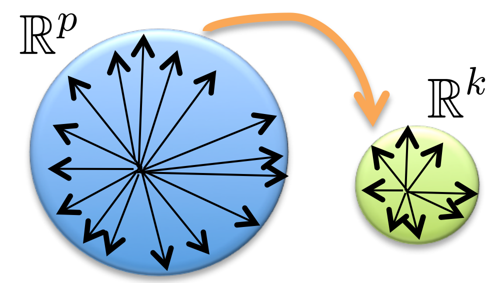


Low-Dim Embedding Summary

Low Rank

$$\Sigma = \Lambda\Lambda' + \Sigma_0$$





k << p

Low-dimensional embedding

- Latent factor models
  - Low-rank covariance approximation to high-dim i.i.d. Gaussian observations
- Dynamic latent factor model
  - Interpretation as state space model with low-dim state
  - Many approaches to modeling latent dynamics, including Gaussian processes
- Capturing changing correlations in high-dim setting
  - Factor structure within dynamic latent factor model
  - Gaussian process “dictionary” functions

Methods for Scaling to High Dimensions

Low Rank $\Sigma = \Lambda\Lambda' + \Sigma_0$

k << p

Low-dimensional embedding

Σ *sparse*

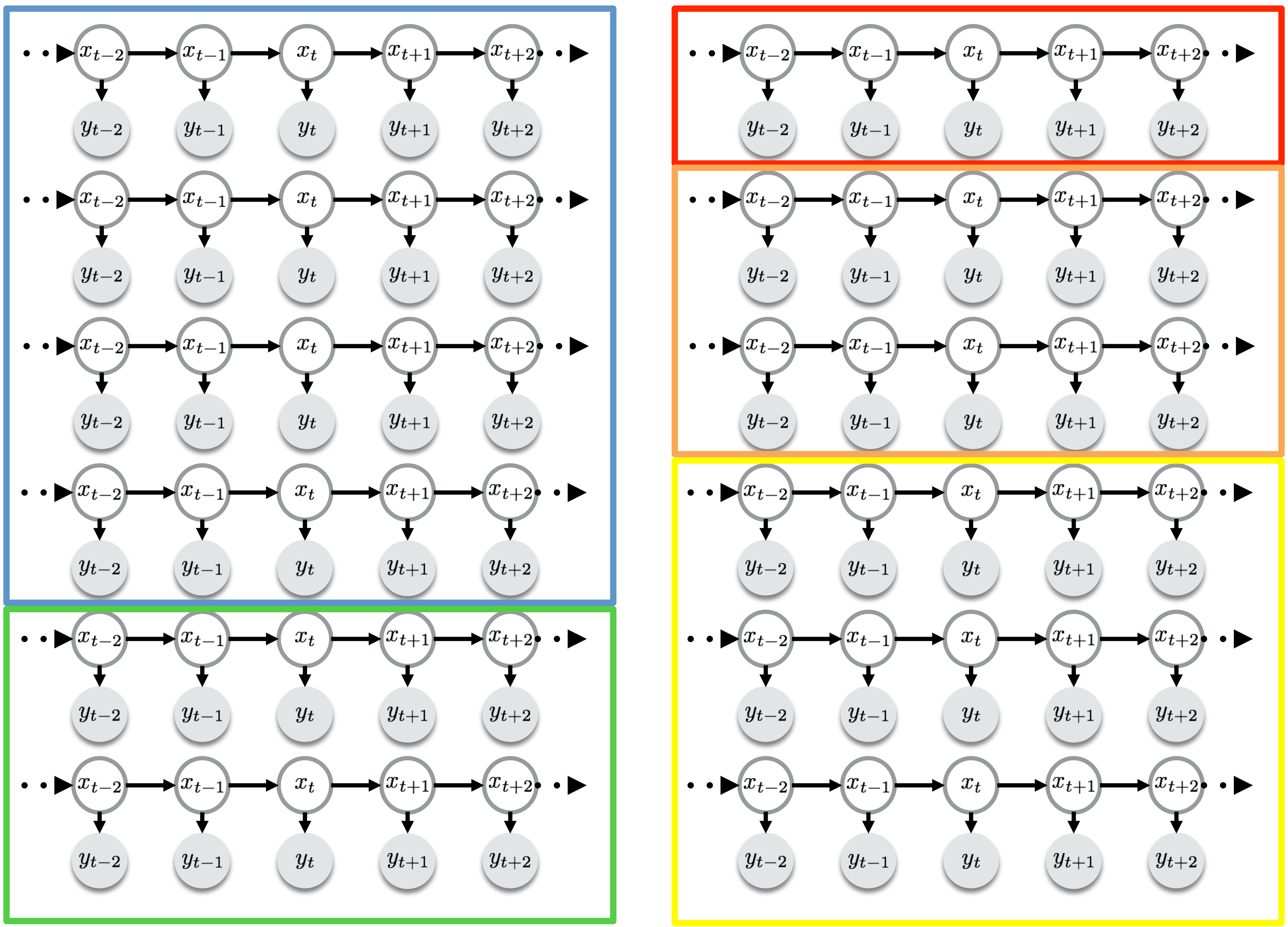
Independent groups of nodes

Σ^{-1} *sparse*

Gaussian Graphical Model

Zeros = no edge in graph, Cond. ind. between nodes

Clustering Time Series



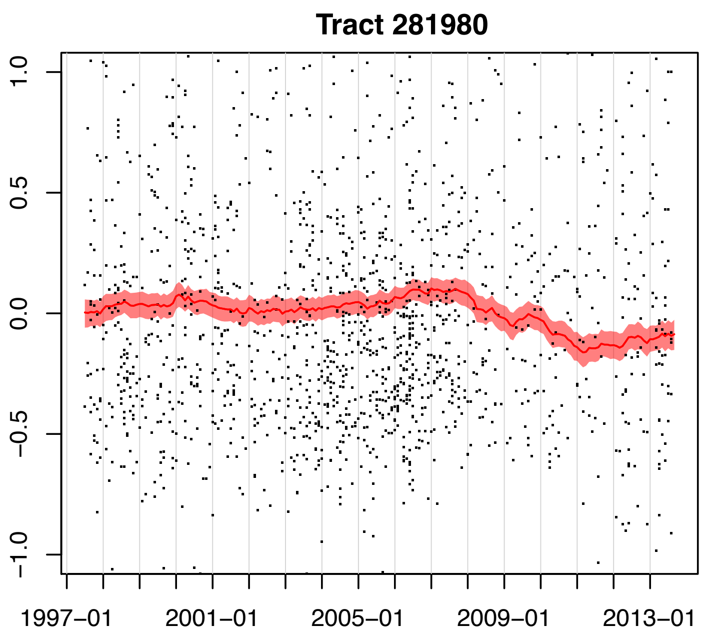
High-Resolution Housing Price Index

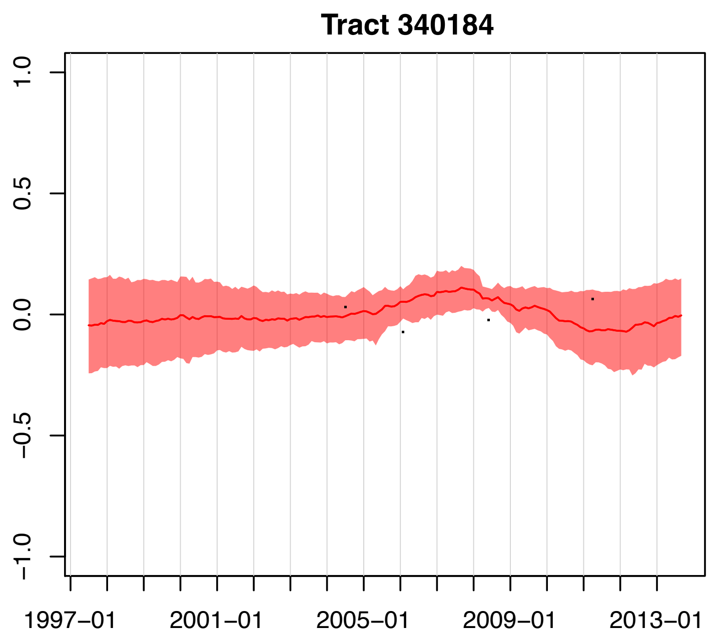
- **Goal:** Model neighborhood housing value over time based on observed house sales (with covariates)

Challenge

Issue: Data are spatiotemporally sparse

Average monthly sales	< 1	< 3	< 5	< 7	< 9
Number of tracts	16	58	114	136	139
Percentage of tracts	11	41	81	97	99



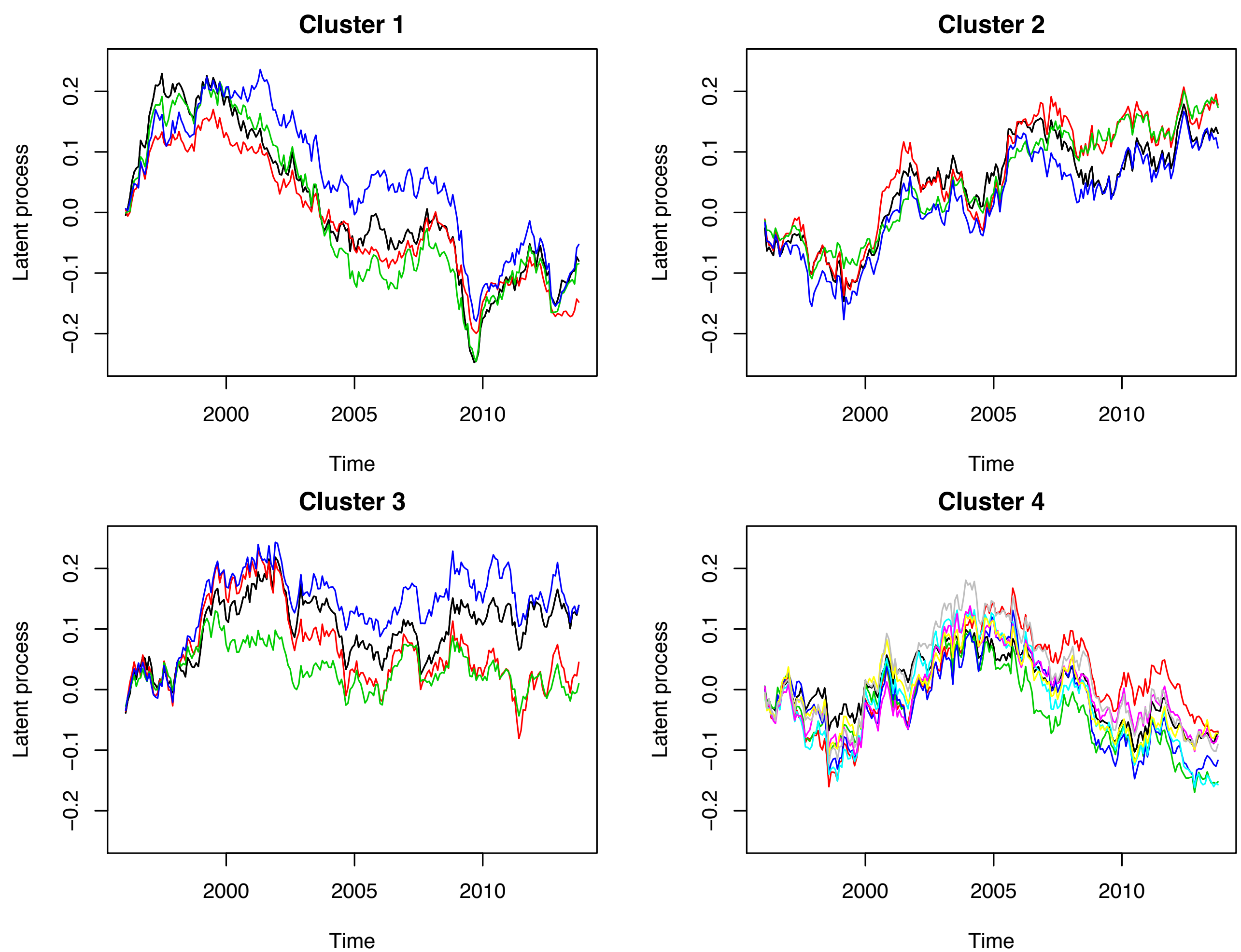


Relate Time Series via Clustering

Solution:

Discover groups of tracts with correlated dynamics

Leverage observations jointly within group



State Space Model

tract i

Hidden: global trend + seasonality

Latent price dynamics $x_{t,i} \;=\; a_i x_{t-1,i} + \epsilon_{t,i} \quad \epsilon_{t,i} \sim \mathcal{N}(0, \sigma_i^2)$

l^{th} sales

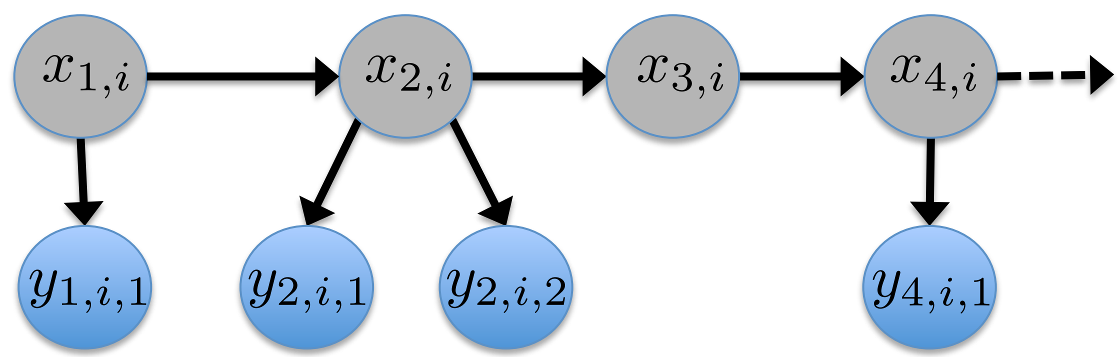
Observed log(price) $y_{t,i,l} \;=\; x_{t,i} + \sum_{h=1}^{H} \beta_{i,h} U_{l,r} + v_{t,i,l} \quad v_{t,i,l} \sim \mathcal{N}(0, R_i)$

covariate effects

Discrete-time linear Gaussian state space model for census tract i

State

Observation

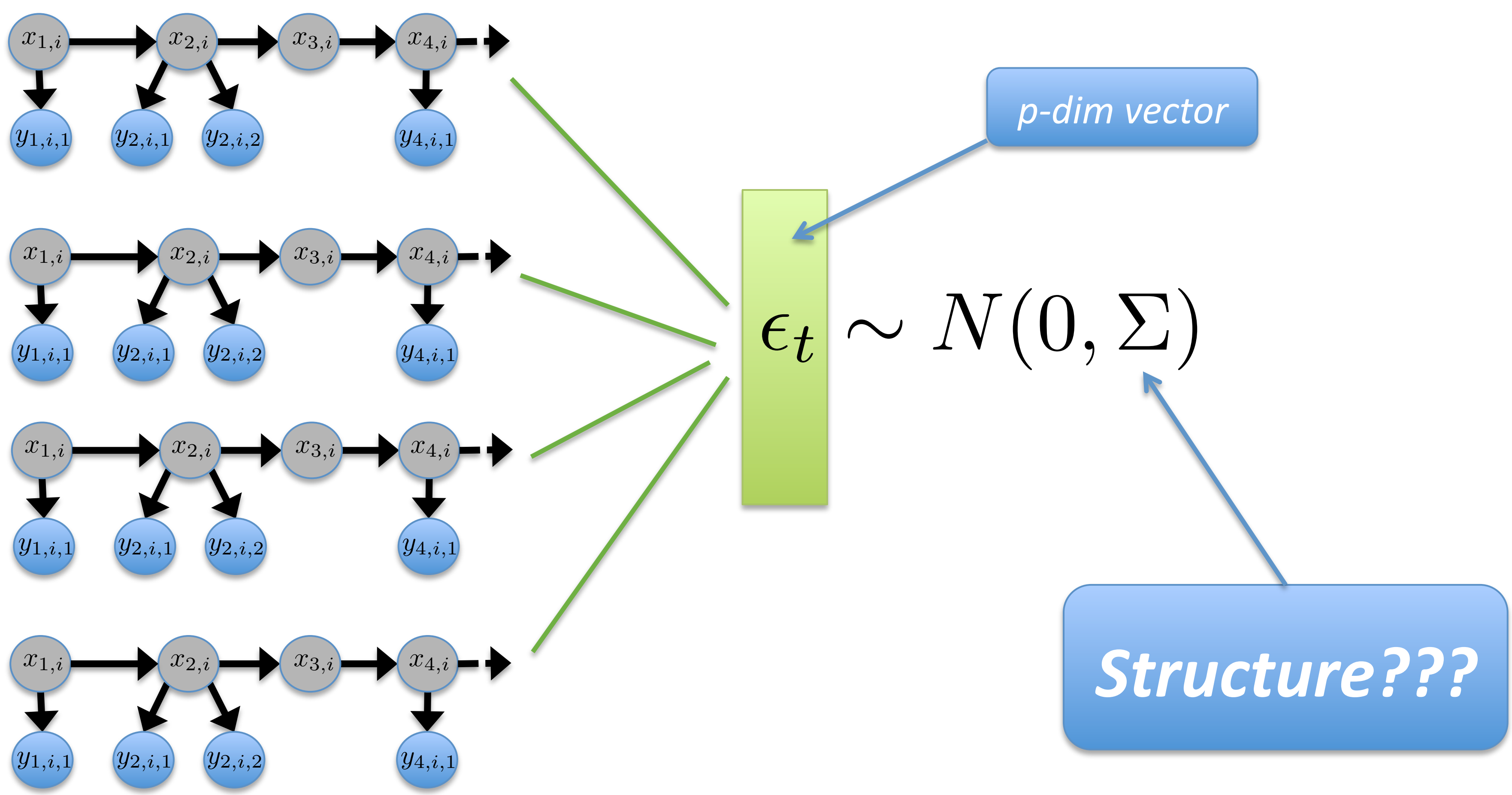


Ren, Fox, Bruce, arXiv 2015.

Multiple Census Tract Model

Latent price dynamics: $x_{t,i} = a_i x_{t-1,i} + \boxed{\epsilon_{t,i}} \quad \epsilon_{t,i} \sim \mathcal{N}(0, \sigma_i^2)$

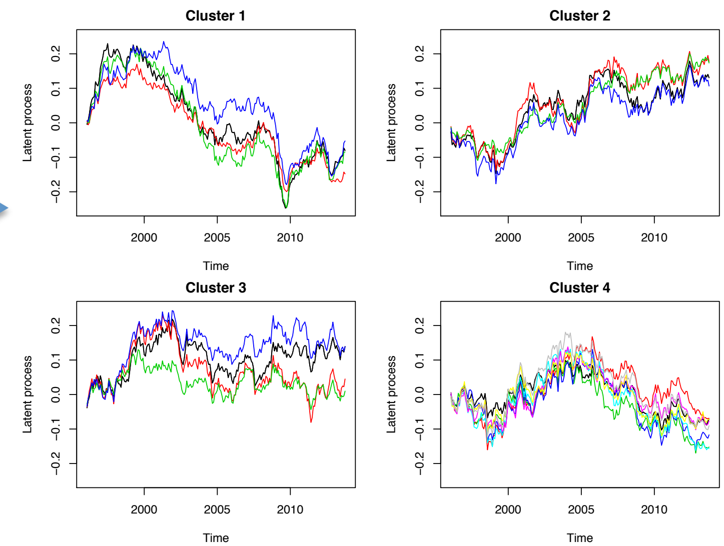
tract *i*



Ren, Fox, Bruce, arXiv 2015.

Cluster and Correlate Multiple Time Series

DYNAMICAL MODEL



$$\begin{pmatrix} \epsilon_{t,1} \\ \epsilon_{t,2} \\ \epsilon_{t,3} \\ \epsilon_{t,4} \\ \vdots \\ \epsilon_{t,p} \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} & & \\ & \Sigma & \\ & & \end{pmatrix} \right]$$

Correlated dynamics in cluster *k*

Cluster and Correlate Multiple Time Series

- Challenge:
  Unknown cluster structure = unknown # of blocks & size of each

- Solution: *Latent factor model* with Bayesian nonparametric prior
  on latent factor processes

$$\begin{pmatrix} \epsilon_{t,1} \\ \epsilon_{t,2} \\ \epsilon_{t,3} \\ \epsilon_{t,4} \\ \vdots \\ \epsilon_{t,p} \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \boxed{\Sigma_1} & & & \mathbf{0} \\ & \boxed{\Sigma_2} & & \\ & & \ddots & \\ \mathbf{0} & & & \boxed{\Sigma_K} \end{pmatrix}\right]$$

Correlated dynamics
in cluster *k*

Latent Factor Model for Innovations

Assume clusterings *known* and *fixed*. If tract i is from cluster k,

$$\epsilon_{t,i} = \lambda_{ik}\eta^*_{t,k} + \tilde{\epsilon}_{t,i} \quad \tilde{\epsilon}_{t,i} \sim N(0,\sigma_0^2) \quad \eta^*_{t,k} \sim N(0,1).$$

factor loadings

latent factor ***process*** for cluster *k*

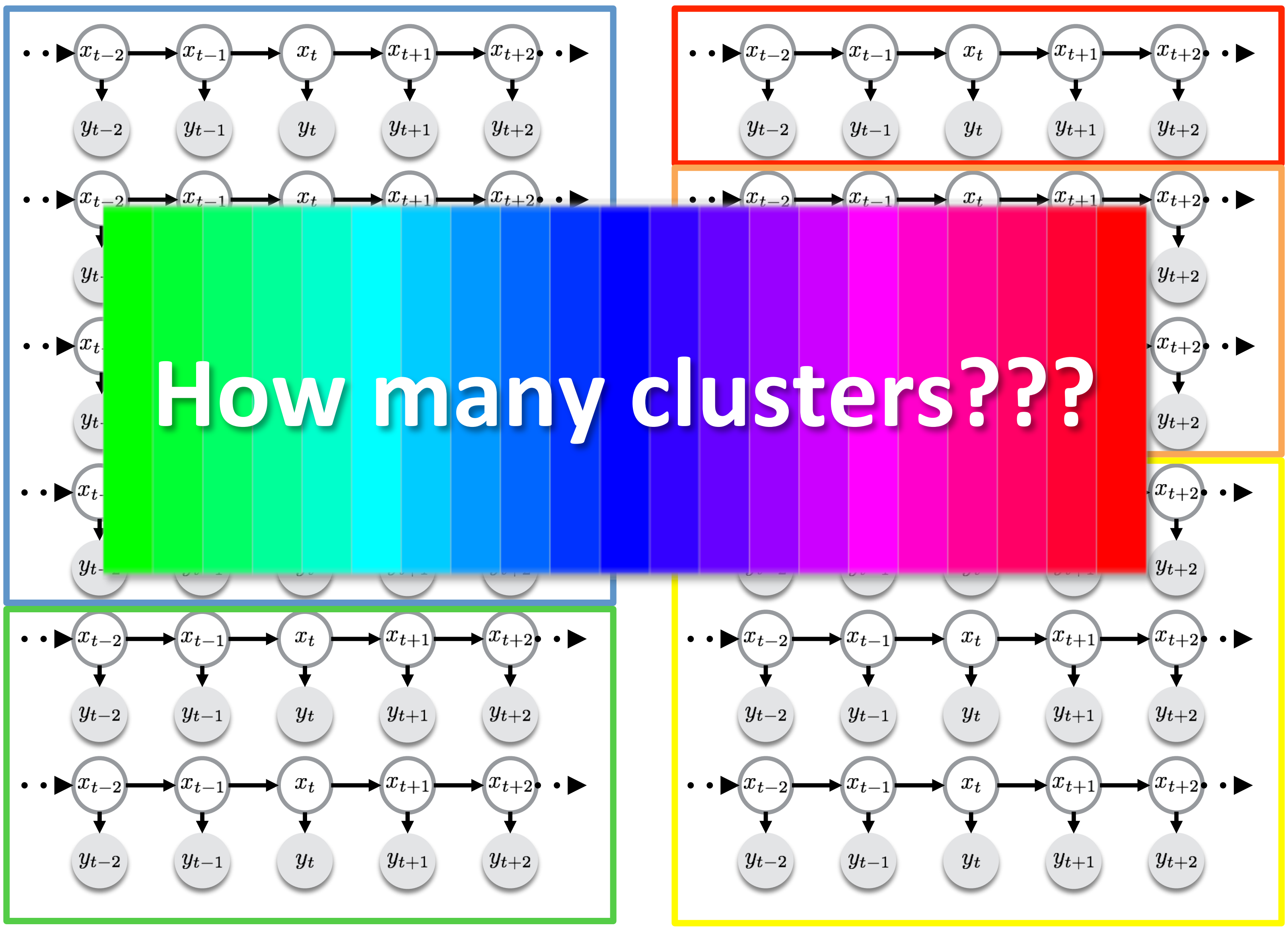
$$\mathrm{cov}(\epsilon_{t,i},\epsilon_{t,i'} \mid \{\lambda\},\{z\}) = \begin{cases} \lambda_{ik}\lambda_{i'k} + \sigma_0^2\delta(i,i') & z_i = z_{i'} = k, \forall k \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{pmatrix} \epsilon_{t,1} \\ \epsilon_{t,2} \\ \epsilon_{t,3} \\ \epsilon_{t,4} \\ \vdots \\ \epsilon_{t,p} \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \boxed{\Sigma_1} & & \mathbf{0} \\ & \boxed{\Sigma_2} & \\ \mathbf{0} & & \ddots \end{pmatrix} \right]$$

(i,j) entry in $\boxed{\Sigma_k} = \lambda_{ik}\lambda_{jk}$

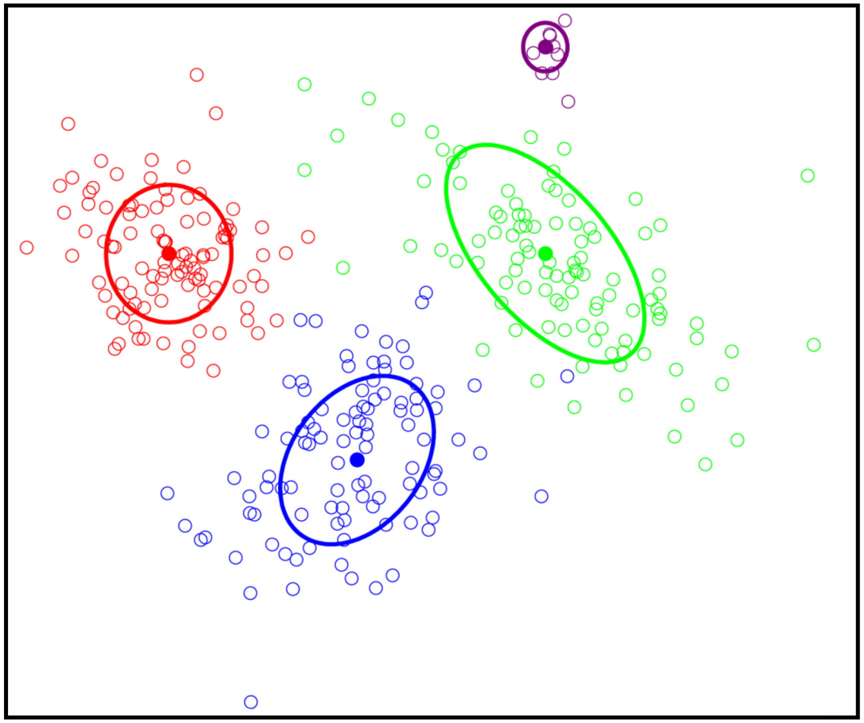
for $i \neq j$

Clustering Time Series



Bayesian Nonparametric Clustering

- Bayesian nonparametric approach:
  - Allows infinite # clusters
  - Uses sparse subset
  - Model ***complexity adapts*** to observations



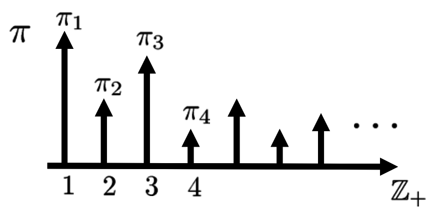
Mixture of Gaussians

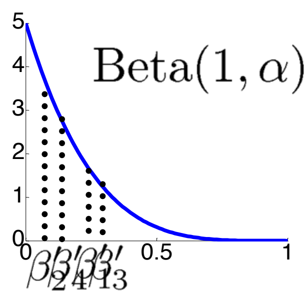
$$\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6 \quad \theta_7 \quad \cdots$$

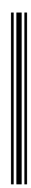
Dirichlet Process Mixture Model

REVIEW

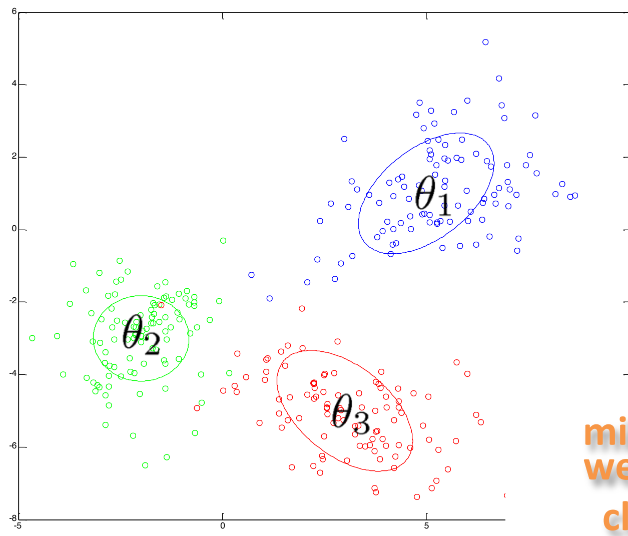
Stick-breaking construction for: DP(αH)



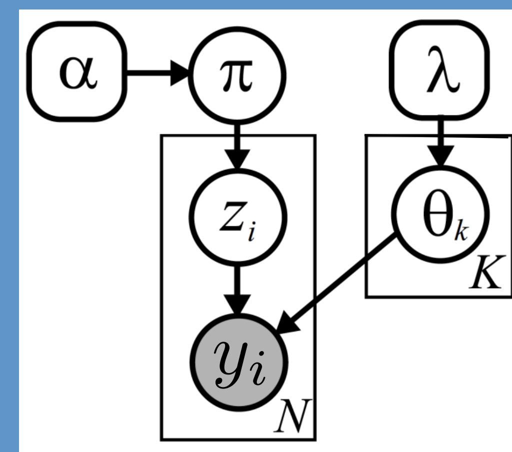




Stick of unit probability mass



DPMM



mixture weights	π	$\sim$	$\text{Stick}(\alpha)$
cluster parameters	θ_k	$\sim$	$H \quad k = 1, 2, \ldots$
cluster labels	z_i	$\sim$	π
observations	y_i	$\sim$	$F(\theta_{z_i})$

Chinese Restaurant Process (CRP)

REVIEW

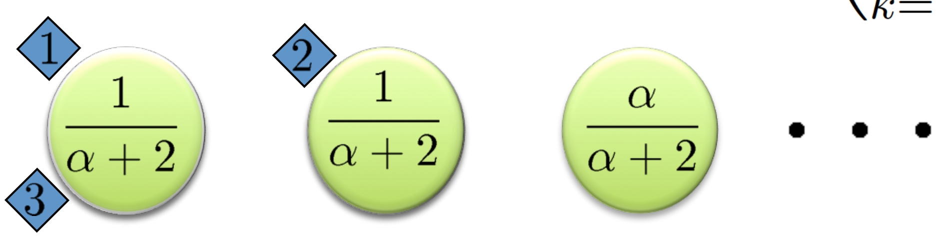
- Distribution on induced partitions described via the CRP
- Visualize clustering as a sequential process of customers sitting at tables in an (infinitely large) restaurant:

customers ⟷ *observed data to be clustered*

tables ⟷ *distinct clusters*

- The first customer sits at a table. Subsequent customers randomly select a table according to:

$$p(z_{N+1} = z \mid z_1, \ldots, z_N, \alpha) = \frac{1}{\alpha + N} \left(\sum_{k=1}^{K} N_k \delta(z, k) + \alpha \delta(z, \bar{k}) \right)$$



Number of current assignments to parameter k

Cluster by Latent Factor Process

tract *i*

Latent price dynamics $x_{t,i} \quad = \quad a_i x_{t-1,i} + \epsilon_{t,i} \quad \epsilon_{t,i} \sim \mathcal{N}(0, \sigma_i^2)$

l^th sales

Observed log(price) $y_{t,i,l} \quad = \quad x_{t,i} + \sum_{h=1}^{H} \beta_{i,h} U_{l,r} + v_{t,i,l} \quad v_{t,i,l} \sim \mathcal{N}(0, R_i)$

covariate effects

Recall: Desired structure attained by assuming that if tract i is from cluster k ,

$$\epsilon_{t,i} = \lambda_{ik}\eta^*_{t,k} + \tilde{\epsilon}_{t,i} \quad \tilde{\epsilon}_{t,i} \sim N(0, \sigma_0^2) \quad \eta^*_{t,k} \sim N(0,1).$$

factor loadings

latent factor ***process*** for cluster *k*

Motivates: Dirichlet process mixture model with

$$\theta_k = \eta^*_{1:T,k} \quad k = 1, 2, \ldots$$

Gaussian i.i.d. version: [Palla et al., NIPS 2012]

Ren, Fox, Bruce, arXiv 2015.

Alternative Clustering

tract i

Latent price dynamics $\quad x_{t,i} = a_i x_{t-1,i} + \epsilon_{t,i} \quad \epsilon_{t,i} \sim \mathcal{N}(0, \sigma_i^2)$

l^{th} sales

Observed log(price) $\quad y_{t,i,l} = x_{t,i} + \sum_{h=1}^{H} \beta_{i,h} U_{l,r} + v_{t,i,l} \quad v_{t,i,l} \sim \mathcal{N}(0, R_i)$

covariate effects

Alternative: Dirichlet process mixture model with

$$\theta_k = \{x^*_{1:T,k}, \beta^*_{k,h}\} \quad k = 1, 2, \ldots$$

Cluster-specific latent trend **Cluster-specific covariate model**

[Nieto-Barajas and Contreras-Cristán, 2014]

Assumes all census tracts in cluster have *same* latent value rather than just *correlated* latent value

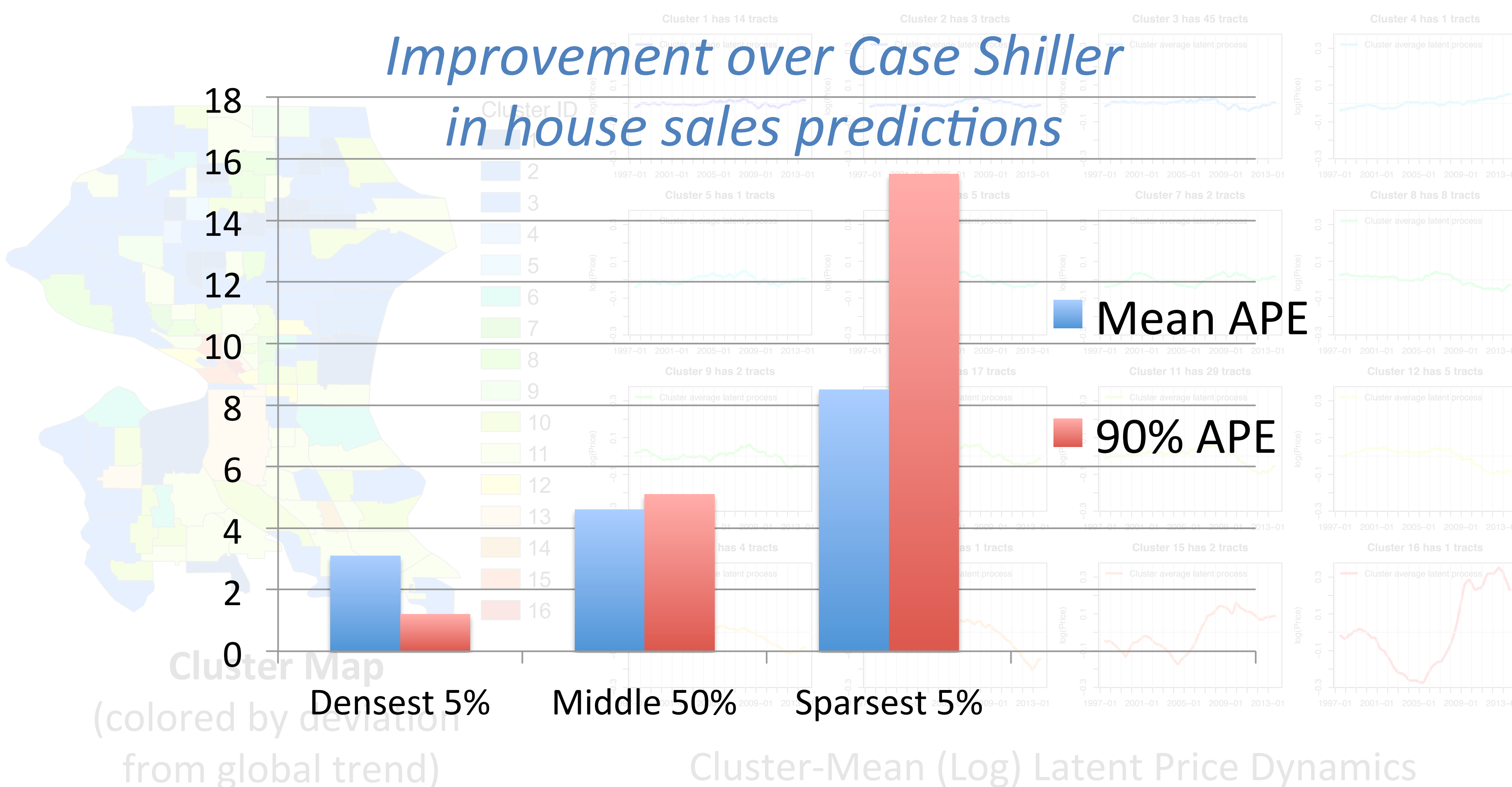
(also cluster parameter x_{T+1} depends on x_T, whereas ε_{T+1} ind. of ε_T)

Housing Data Analysis

- Seattle City
  - 140 census tracts
  - 125k transactions during 17 years

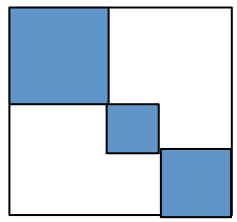
- Computational details:
  - Parallel (collapsed) Dirichlet process MCMC sampler [Williamson et al., ICML 2013]
  - 10x speedup with 10 processors

Seattle City Analysis (17 years)



Clusters of Time Series Summary

Σ *sparse*



Independent groups of nodes

- **Goal:** Cluster time series to share information
  - Individually not informative enough
  - Full joint model statistically and computationally infeasible
- Cluster structure
  - Cluster on latent state process → clusters of *identical latent trends*
  - Assume latent factor model for AR innovations + cluster latent factor process → clusters of *correlated time series*
- Bayesian nonparametric clustering
  - Dirichlet process prior allows unknown number of clusters

Methods for Scaling to High Dimensions

Low Rank $\Sigma = \Lambda\Lambda' + \Sigma_0$

k << p

Low-dimensional embedding

Σ *sparse*

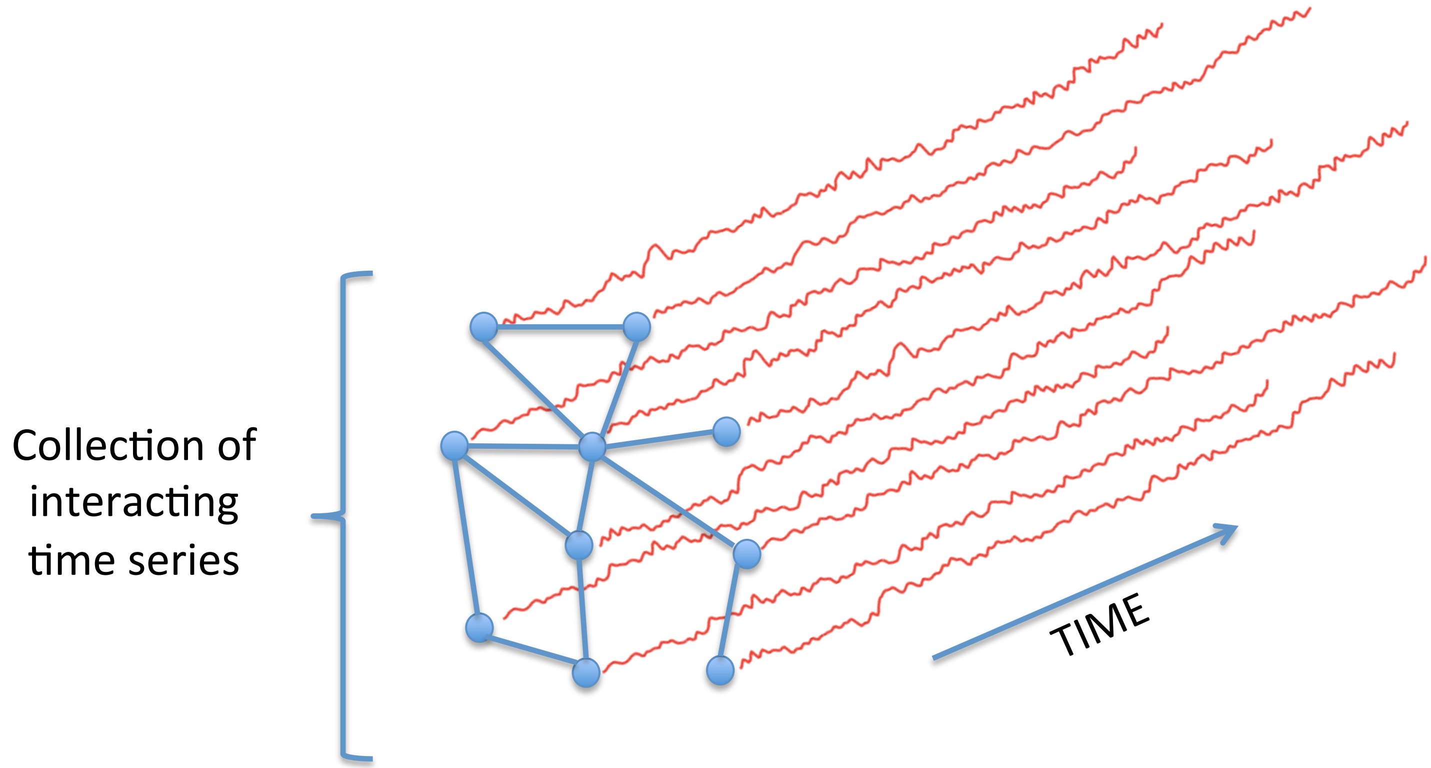
Independent groups of nodes

Σ^{-1} *sparse*

Gaussian Graphical Model

Zeros = no edge in graph,
Cond. ind. between nodes

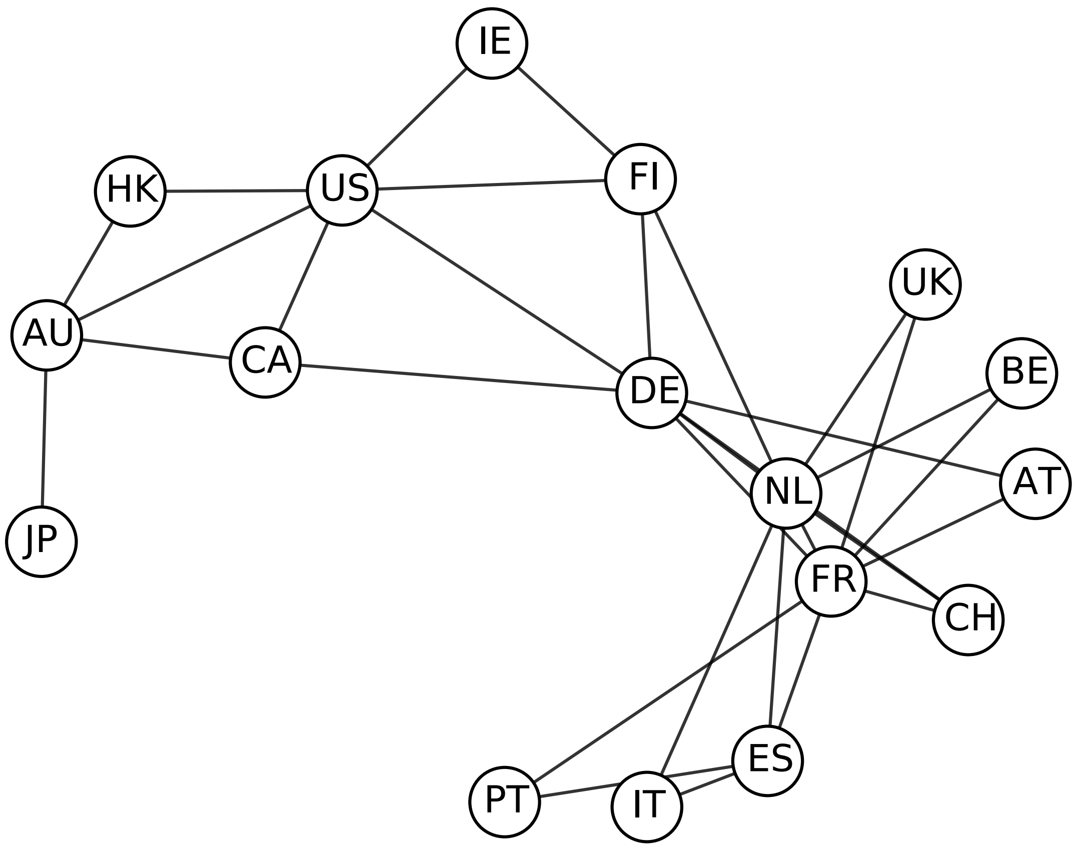
Graphs of Time Series



Conditional Independencies

Collection of global stock indices

Data: time series of daily returns

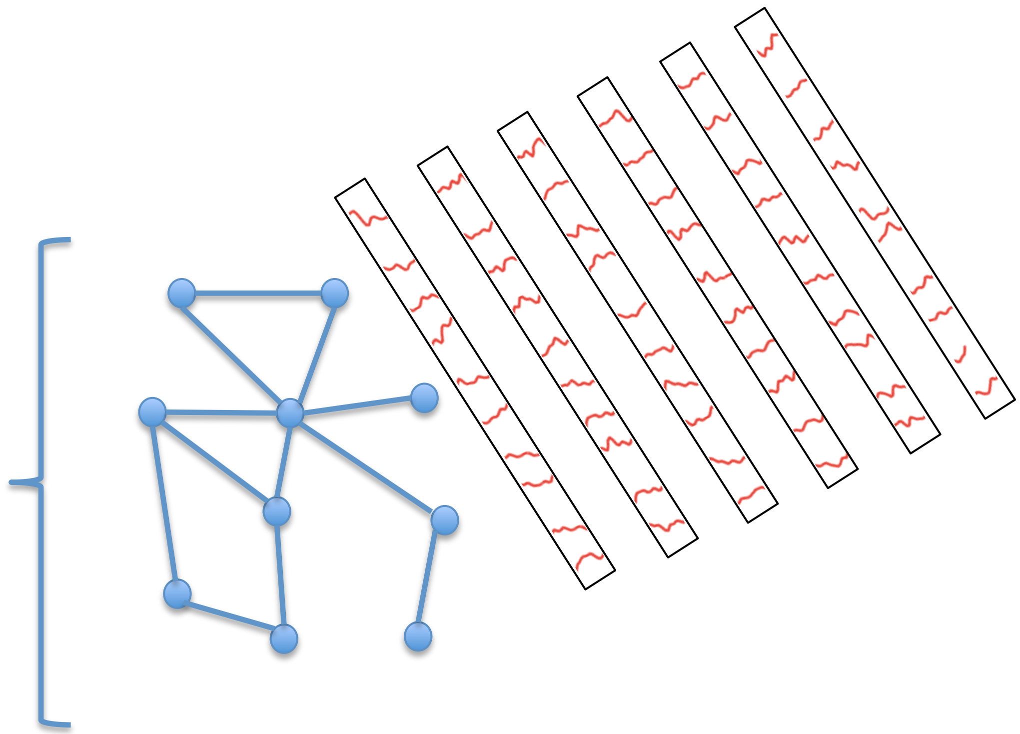


Countries:

Australia (AU)
Austria (AT)
Belgium (BE)
Canada (CA)
Finland (FI)
France (FR)
Germany (DE)
Hong Kong (HK)
Ireland (IE)
Italy (IT)
Japan (JP)
Netherlands (NL)
Portugal (PT)
Spain (ES)
Switzerland (CH)
United Kingdom (UK)
United States (US)

Graphs of i.i.d. Data

Collection of interacting random variables

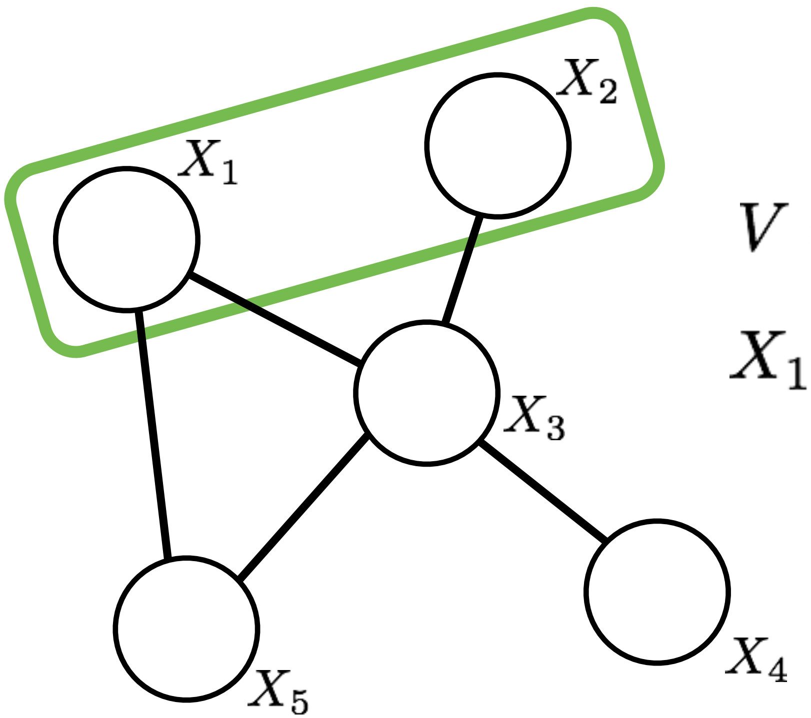


Graphical Models for Random Variables

- Graph G=(V,E) encodes conditional independence statements

nodes edges

no edge (i,j) $\Rightarrow X_i$, X_j cond. ind. given rest

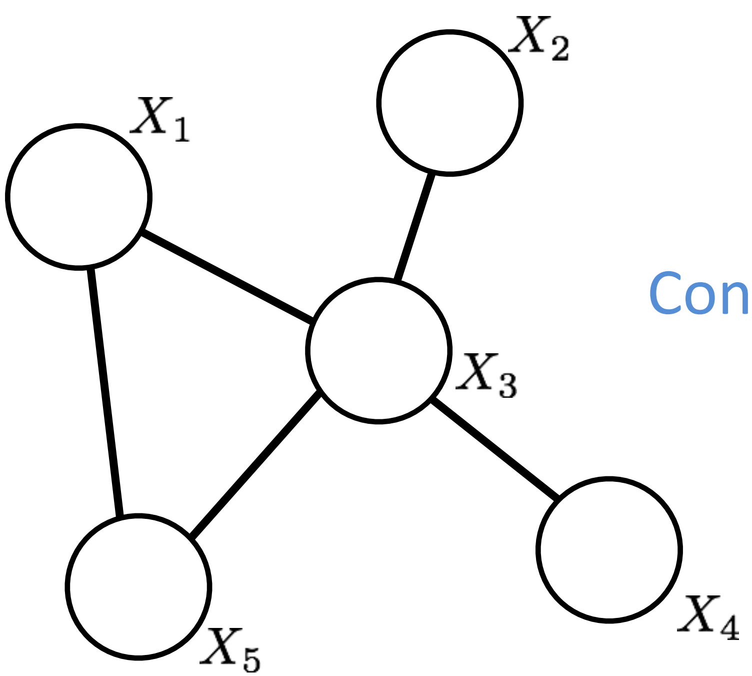


$$V = \{1, 2, 3, 4, 5\}$$

$$X_1 \perp\!\!\!\perp X_2 | (X_3, X_5, X_4)$$

Gaussian Graphical Models

- Assume Gaussian random vector $X \sim \mathcal{N}(0, \Sigma)$



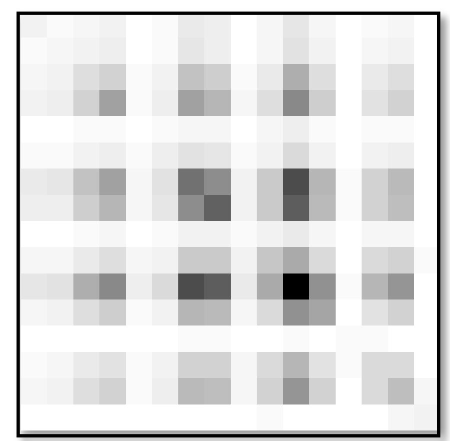
no edge (*i,j*) $\Rightarrow$ X_i , X_j cond. ind. given rest

$$\Updownarrow$$

$$(\Sigma^{-1})_{ij} = 0$$

Cond. ind. encoded in *precision* matrix

$$\Sigma^{-1} =$$



Information Form Gaussian

- Motivations for considering "information form" of multivariate normal
  - Easier to read off conditional densities
  - Has log-linear form in terms of "information parameters"

$x \sim N(\mu, \Sigma)$

$$\frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

$$\Updownarrow \quad \boxed{\begin{aligned}\Omega &= \Sigma^{-1} \\ \eta &= \Sigma^{-1}\mu\end{aligned}}$$

$x \sim N^{-1}(\eta, \Omega)$

$$\propto e^{\eta^T x - \frac{1}{2} x^T \Omega x}$$

Info. Gaussian Conditional Densities

- Assume a model with

$$x \sim N^{-1}(\eta, \Omega)$$

and divide the dimensions into two sets $A, \bar{A}$

- Then,

$$\begin{bmatrix} x_A \\ x_{\bar{A}} \end{bmatrix} \sim N^{-1}\left(\begin{bmatrix} \eta_A \\ \eta_{\bar{A}} \end{bmatrix}, \begin{bmatrix} \Omega_{AA} & \Omega_{A\bar{A}} \\ \Omega_{\bar{A}A} & \Omega_{\bar{A}\bar{A}} \end{bmatrix}\right)$$

$$p(x_A \mid x_{\bar{A}}) = N^{-1}(\eta_A - \Omega_{A\bar{A}} x_{\bar{A}}, \Omega_{AA})$$

Info. Gaussian Conditional Densities

- Let $A = \{s, t\}$ and $\bar{A}$ everything else

$$\begin{bmatrix} \Omega_{ss} & \Omega_{st} \\ \Omega_{ts} & \Omega_{tt} \end{bmatrix}$$

inverse cov. of $p(x_s, x_t \mid x_{\setminus st})$

$$p(x_A \mid x_{\bar{A}}) = N^{-1}(\eta_A - \Omega_{A\bar{A}} x_{\bar{A}}, \Omega_{AA})$$

- What if $\boxed{\Omega_{st} = 0}$?

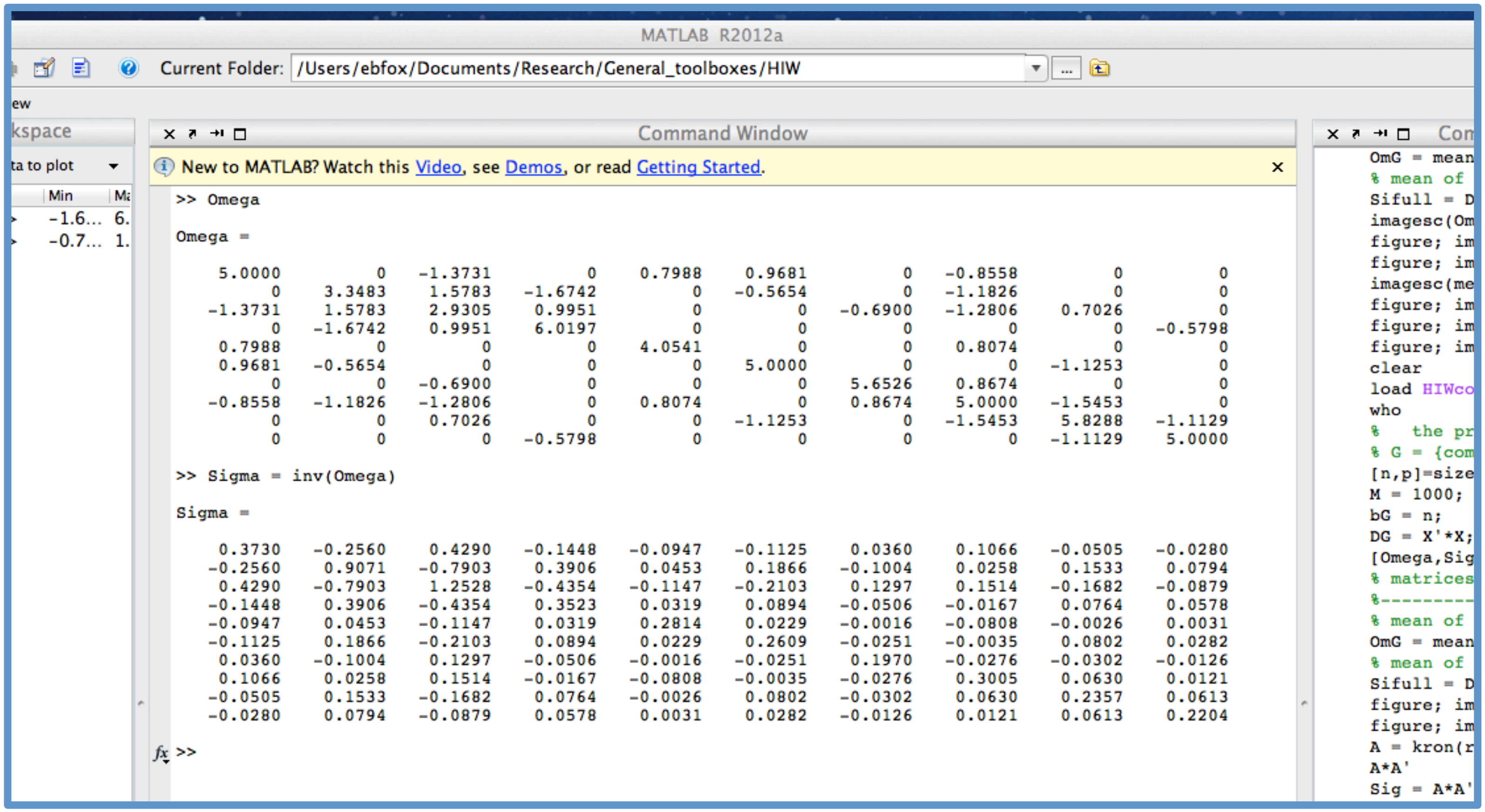
$$\text{cov}(x_s, x_t \mid x_{\setminus st}) = \Omega_{AA}^{-1} = \begin{bmatrix} \Omega_{ss}^{-1} & 0 \\ 0 & \Omega_{tt}^{-1} \end{bmatrix}$$

$$\boxed{\Leftrightarrow x_s \perp\!\!\!\perp x_t \mid x_{\setminus st}}$$

- *Precision matrix encodes conditional independencies*

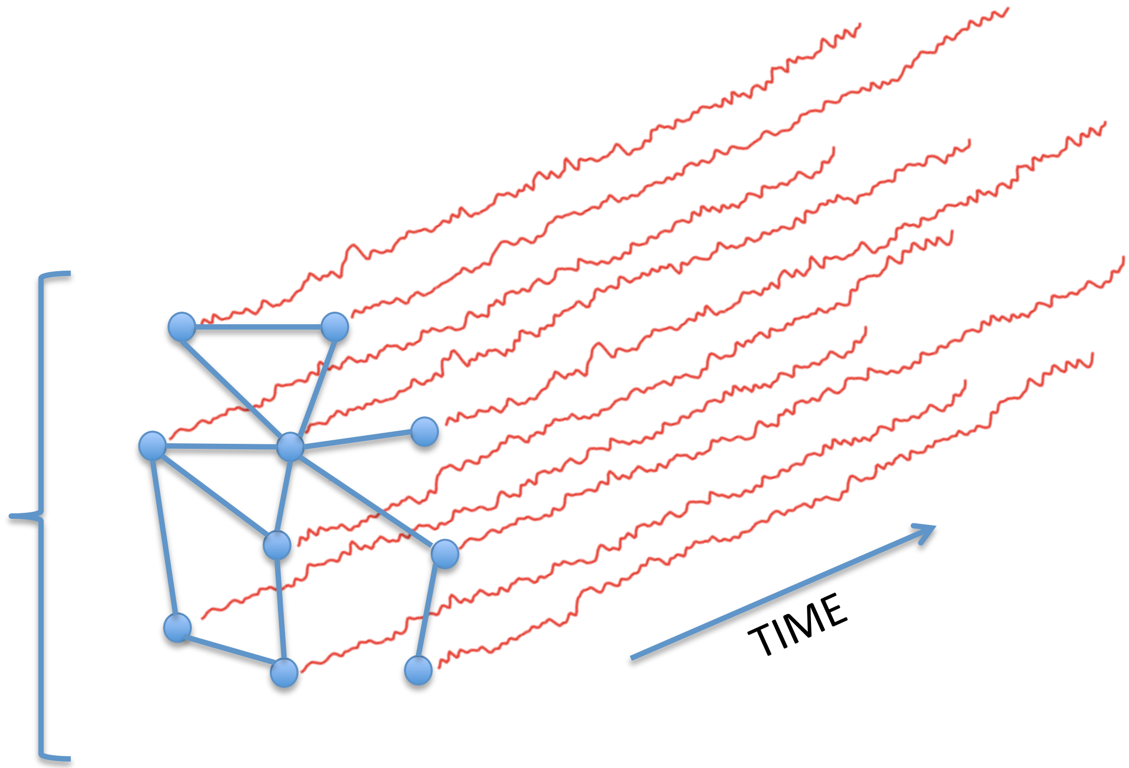
Sparse Precision vs. Covariance

- For a sparse precision matrix, the covariance need not be

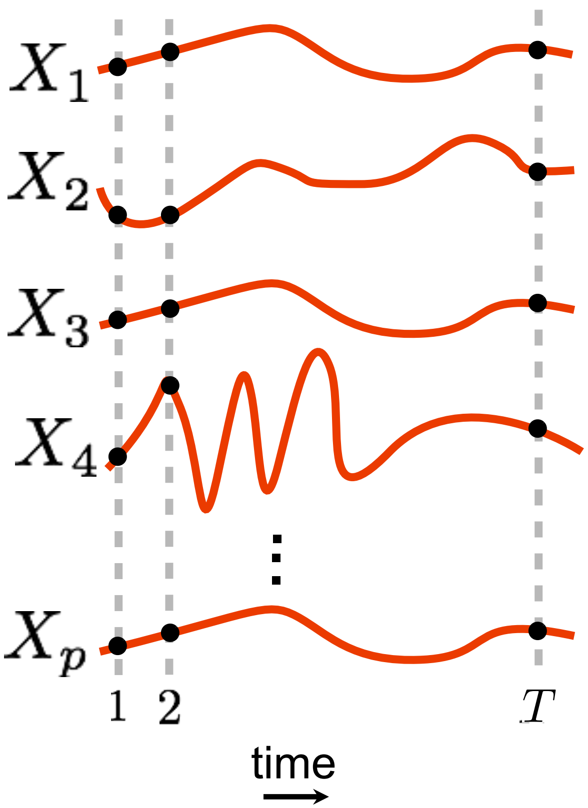


Defining Graphs of Time Series

Collection of interacting time series



Random Variables → Stochastic Processes



Goal: Represent and infer conditional independence relations between *time series*

Assume *stationarity*:

$$E(X(t)) = \mu$$

$$\text{Cov}(X(t), X(t+h)) = \Gamma(h)$$

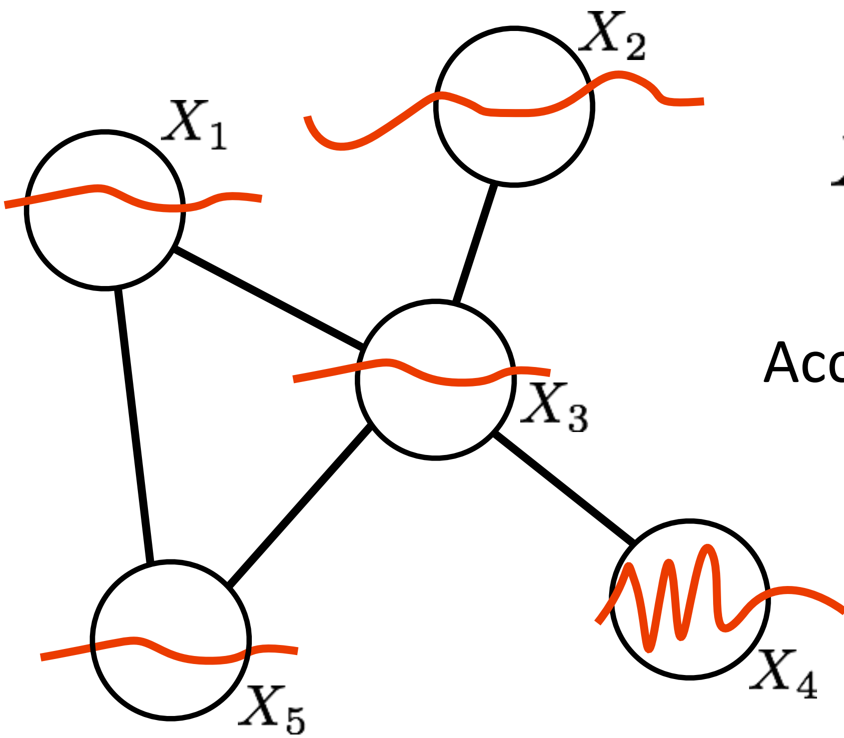
For simplicity, zero mean

Graphical Models for Time Series

no edge $(i,j) \Longrightarrow$ *time series* X_i , X_j cond. ind.
given *entire histories* of other series

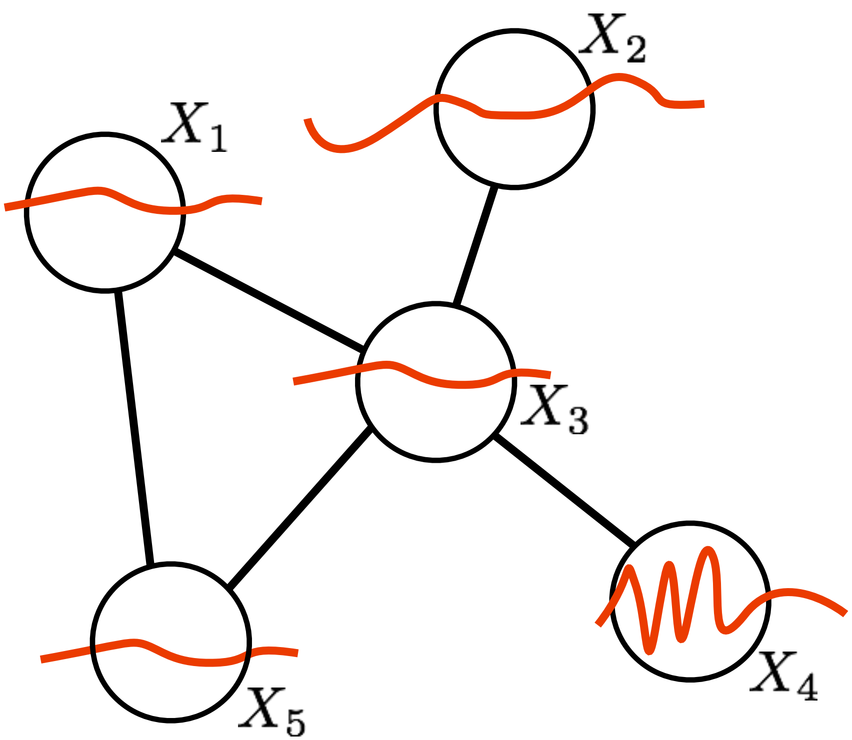
$$\Updownarrow$$

$$X_1(t) \perp\!\!\!\perp X_2(t+h) | X_{V \setminus 1,2}$$



Accounts for interactions at *any lag*

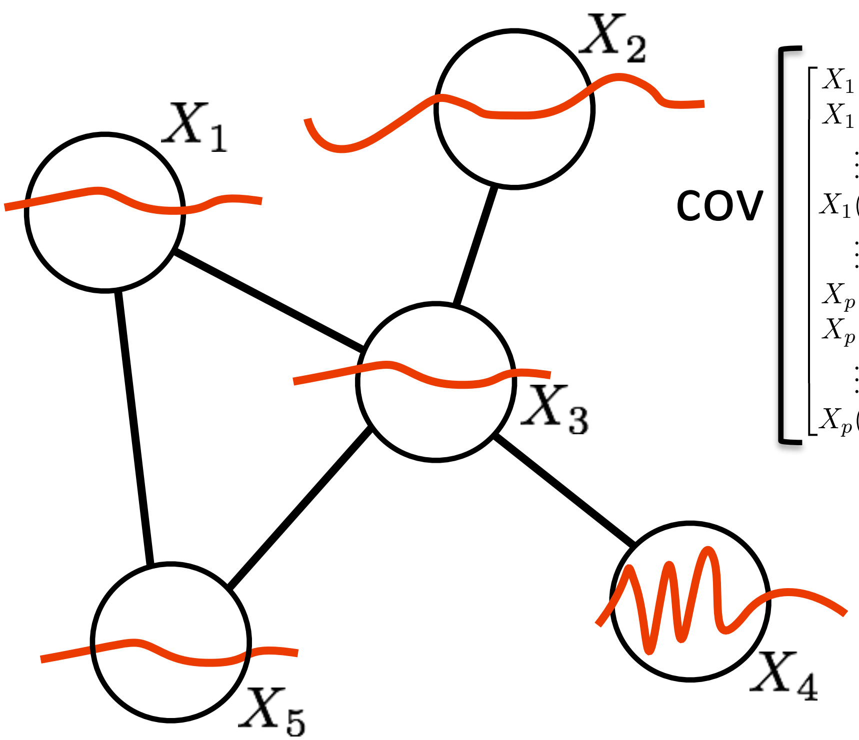
Naïve Approach to Structure Learning



$$\tilde{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = \begin{bmatrix} X_1(1) \\ X_1(2) \\ \vdots \\ X_1(T) \\ \vdots \\ X_p(1) \\ X_p(2) \\ \vdots \\ X_p(T) \end{bmatrix}$$

Naïve Approach to Structure Learning

$$\boxed{\tilde{X}} \sim N(0, \Sigma)$$



$$\text{cov}\begin{bmatrix} X_1(1) \\ X_1(2) \\ \vdots \\ X_1(T) \\ \vdots \\ X_p(1) \\ X_p(2) \\ \vdots \\ X_p(T) \end{bmatrix} = \begin{bmatrix} \Gamma(0) & \Gamma(1) & \Gamma(2) & \cdots & \Gamma(T-1) \\ \Gamma(1)' & \Gamma(0) & \Gamma(1) & \cdots & \Gamma(T-2) \\ \vdots & & \ddots & & \vdots \\ \Gamma(T-1)' & \Gamma(T-2)' & \cdots & \Gamma(1)' & \Gamma(0) \end{bmatrix}$$

Tp x Tp matrix

Lagged covariance matrix

$$\Gamma(h) = \text{Cov}(X(t), X(t+h))$$

Previous Approaches

- Songsiri et al. 2011
  - Assume parametric time series model
  - Determine conditions on parameters leading to conditional independencies
  - Optimize penalized likelihood

VAR(q) process: $X(t) = \sum_{i=1}^{q} A_i X(t-i) + \epsilon(t) \quad \epsilon(t) \sim N(0, \Sigma_\epsilon)$

Define: $B_k = \Sigma_\epsilon^{-\frac{1}{2}} A_k \qquad Y^k = \sum_{\ell=0}^{p-k} B_\ell^T B_{k+\ell}$

Main Result:

$$X_a \perp X_b \mid X_{V \setminus \{a,b\}} \iff Y_{ab}^k = Y_{ba}^k = 0 \quad \forall k$$

Objective: Penalized likelihood with a group penalty to enforce group sparsity

Tool: Optimize convex relaxation

Previous Approaches

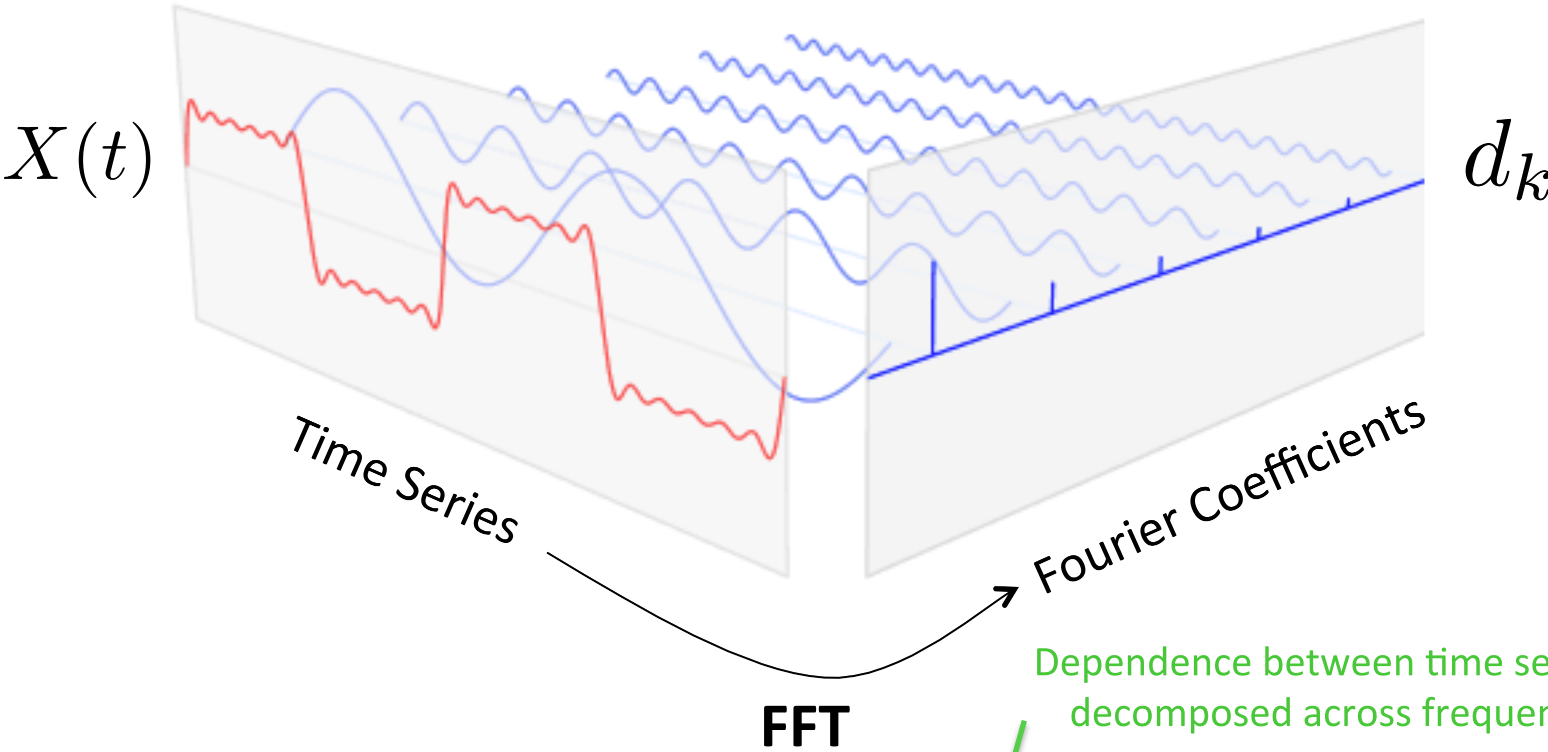
- Songsiri et al. 2011
  - Assume parametric time series model
  - Determine conditions on parameters leading to conditional independencies
  - Optimize penalized likelihood

- Dahlhaus 2000, Matsuda 2006, Wolstenholme and Walden 2015, Bach and Jordan 2004, Jung et al. 2014
  - Transform to frequency domain
  - Determine conditions on spectral parameters leading to conditional independencies
  - Hypothesis test to see if conditions are satisfied

    OR

  - Optimize a Whittle-approximated (penalized) likelihood

Model in the Frequency Domain



Dependence between time series, decomposed across frequencies

Spectral density matrix

Lagged covariance matrix

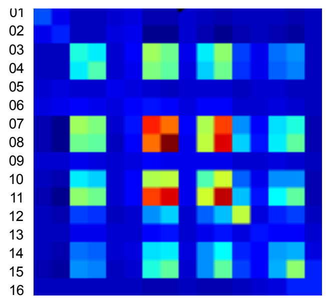
$$\Gamma(h) = \mathrm{Cov}(X(t), X(t+h)) \longrightarrow S(\lambda) = \sum_{h=-\infty}^{\infty} \Gamma(h) e^{-i\lambda h}$$

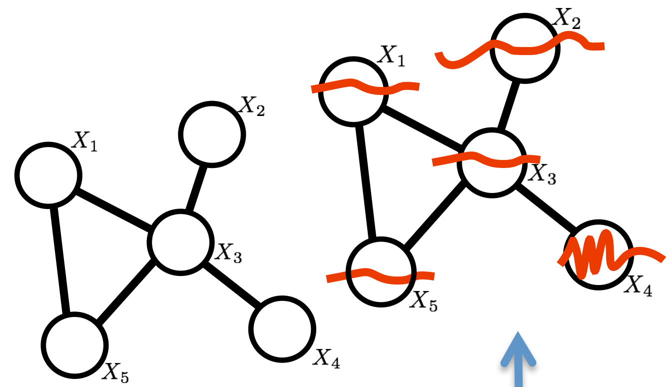
What conditions on S leads to conditional independence?

Encoding Time Series Structure

For **Gaussian** i.i.d. random variables,

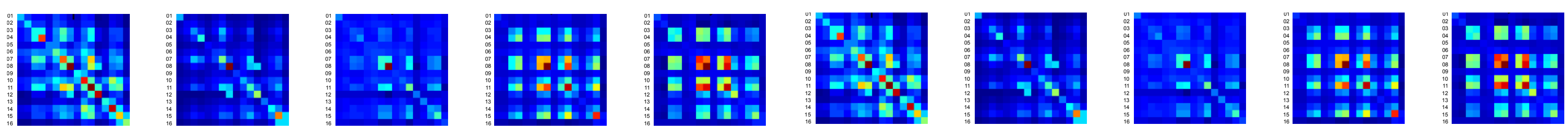
$$\Sigma^{-1} =$$





(*Dahlhaus, 2000*) For **Gaussian** stationary time series,

$S^{-1}(\lambda)$:



$$\underbrace{\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_k \quad \cdots \quad \lambda_T}$$

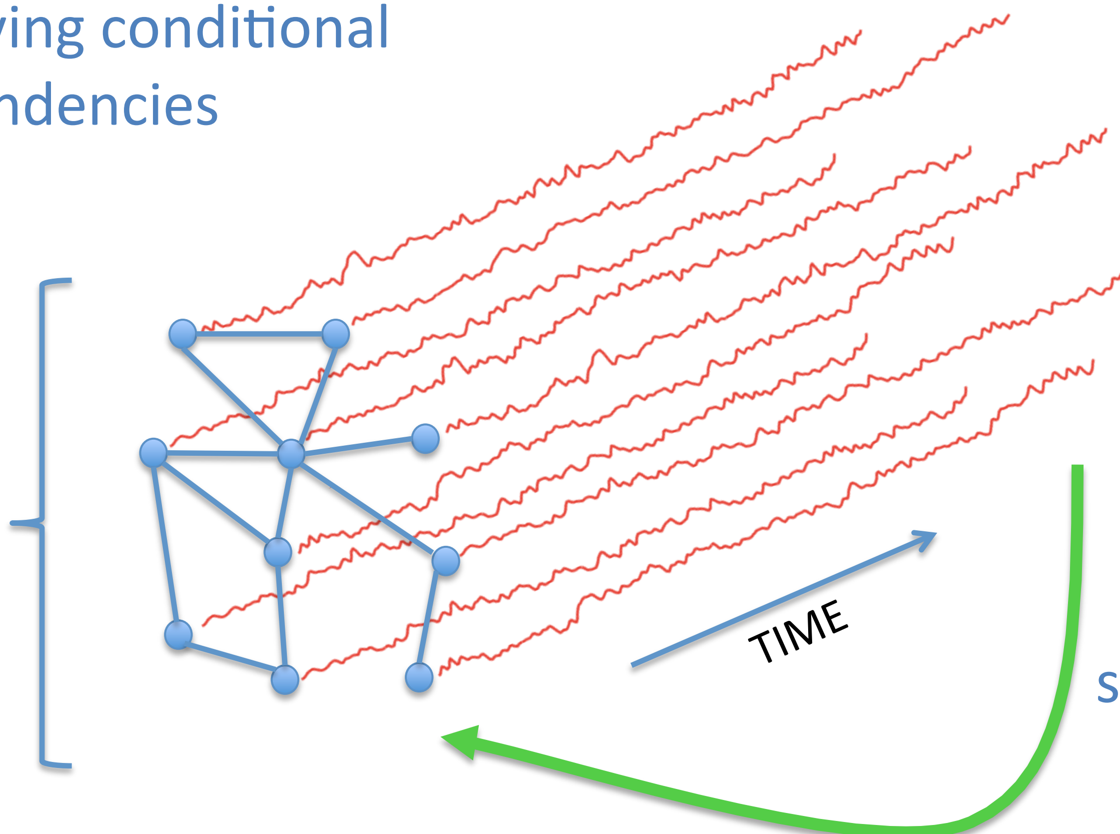
complex inverse **spectral density** matrices

$$X_i \perp X_j \mid X_{V\backslash i,j} \iff S(\lambda)^{-1}_{ij} = 0 \quad \forall \lambda \in [0, 2\pi]$$

Learning Graphs of Time Series

Dalhaus gives conditions for identifying conditional independencies

Collection of interacting time series



How to learn structure from data???

Likelihood approach:

need to relate data to spectral density matrix

Whittle Approximation (no graph)

$$d_k = \frac{1}{T}\sum_{t=0}^{T-1} X(t)e^{-i\lambda_k t} \qquad \lambda_k = \frac{2\pi k}{T}$$

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = \tilde{X} \sim N(0, \Sigma)$$

- ***Fourier coeff. asymptotically independent***

$$d_k \sim \mathcal{N}_c(0, S_k) \quad k = 0, \ldots, T-1$$

- Instead of likelihood depending on *Tp* x *Tp* covariance, decomposes over *p* x *p* spectral density matrices:

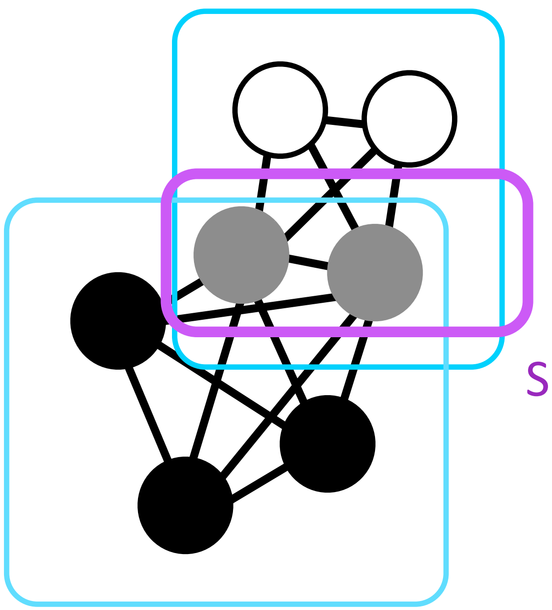
$$p(X_{1:p}|S_{0:T-1}) \approx \prod_{k=0}^{T-1} \frac{1}{\pi^p |S_k|} e^{-d_k^* S_k^{-1} d_k}$$

$*$ indicates the conjugate transpose

Close to Gaussian likelihood with iid data…

Decomposable Graphs

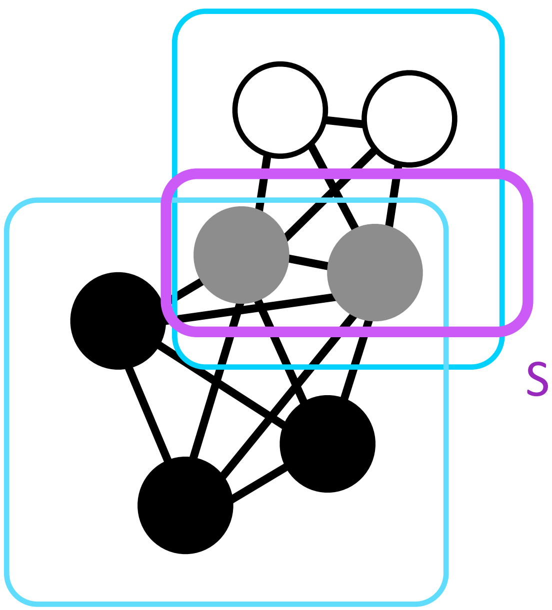
- If graph is ***decomposable***, joint distribution decomposes over cliques C and separators S



$$p(x) = \frac{\prod_{C \in \mathcal{C}} p(x_C)}{\prod_{S \in \mathcal{S}} p(x_S)}$$

Whittle on Decomposable Graphs

$$p(X_{1:p}|S_{0:T-1}) \approx \prod_{k=0}^{T-1} \frac{1}{\pi^p |S_k|} e^{-d_k^* S_k^{-1} d_k}$$



$$p(X_{1:p}|\boxed{G}, S_{0:(T-1)}) \approx$$

$$\prod_{k=0}^{T-1} \frac{\prod_{C\in\mathcal{C}} \frac{1}{\pi^{p|C|}|S_{kC}|^p} e^{-\mathrm{tr}\boxed{P_{kC}}S_{kC}^{-1}}}{\prod_{S\in\mathcal{S}} \frac{1}{\pi^{p|S|}|S_{kS}|^p} e^{-\mathrm{tr}\boxed{P_{kS}}S_{kS}^{-1}}}$$

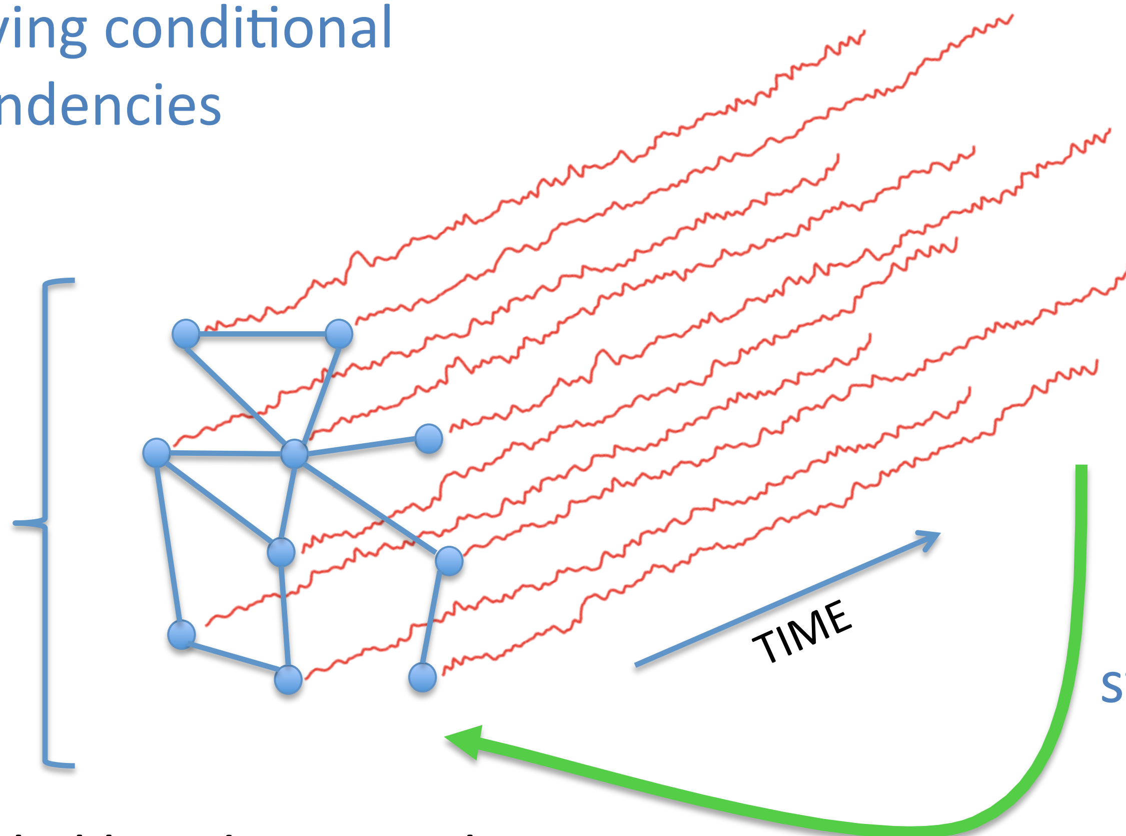
Periodogram at frequency λ_k

$$d_k d_k^*$$

Learning Graphs of Time Series

Dalhaus gives conditions for identifying conditional independencies

Collection of interacting time series



How to learn structure from data???

Likelihood approach:

need to relate data to spectral density matrix

Bayesian approach: specify priors

Bayesian Approach to Structure Learning

Gaussian Graphical Model

Graph prior $\mathcal{G} \sim p(\mathcal{G})$

Prior on *graph-constrained* covariance (inverses with specified 0's) $\Sigma \sim p(\Sigma \mid \mathcal{G})$

Normal obs. $X \sim \mathcal{N}(0, \Sigma)$

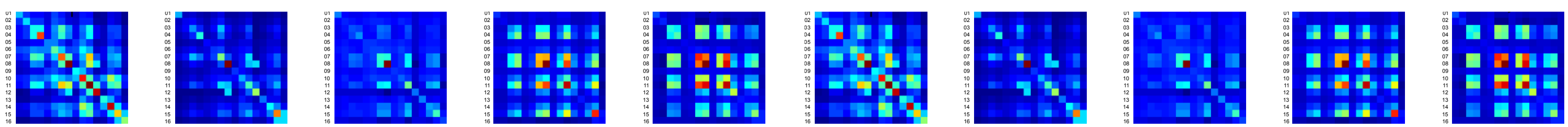
Time Series Graphs

$\mathcal{G} \sim p(\mathcal{G})$ Graph prior

$S_k \sim p(S_k \mid \mathcal{G})$ Prior on *graph-constrained* spectral density matrices (inverses with specified 0's)

$d_k \sim \mathcal{N}_c(0, S_k)$ Complex normal Fourier coeff.

Fourier coefficients **Whittle approx.**



$\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_k \quad \cdots \quad \lambda_T$

Graph-constrained, *complex* spectral density matrices

Tank, Foti, Fox, *UAI 2015.*

Defining a Conjugate Prior

Gaussian Graphical Model

Graph prior $\mathcal{G} \sim p(\mathcal{G})$

Prior on *graph-constrained* covariance (inverses with specified 0's) $\Sigma \sim p(\Sigma \mid \mathcal{G})$ ***HIW distribution***

Normal obs. $X \sim \mathcal{N}(0, \Sigma)$

Time Series Graphs

$\mathcal{G} \sim p(\mathcal{G})$ Graph prior

$S_\omega \sim p(S_\omega \mid \mathcal{G})$ ***complex HIW distribution*** Prior on *graph-constrained* spectral density matrices (inverses with specified 0's)

$d_k \sim \mathcal{N}_c(0, S_k)$ Complex normal Fourier coeff.

Conjugate prior → marginal likelihood $\underbrace{p(\overset{\text{data}}{X} \mid \overset{\text{graph structure}}{\mathcal{G}})}_{\text{graph score}}$

Use existing graph search algorithms

Tank, Foti, Fox, *UAI 2015.*

Marginal Likelihood

REVIEW

- **Hyper complex inverse Wishart prior** on Σ

$$p(\Sigma|\delta, W, G) = \boxed{h(W,\delta,G)} \mathbf{1}_{\Sigma\in M^+(G)} |\Sigma|^{-(\delta+2p)} e^{-\mathrm{tr} W\Sigma^{-1}}$$

- Complex normal observation *d*

$$p(d|G,\Sigma) = \boxed{\frac{1}{\pi^p}} |\Sigma|^{-1} e^{-\mathrm{tr} P\Sigma^{-1}}$$

- Marginal likelihood

$$\frac{1}{\boxed{h(W+P,\delta+1,G)}} p(\Sigma \mid \delta+1, W+P, G)$$

$$p(d|G) = \int_\Sigma p(d \mid \Sigma) p(\Sigma) d\Sigma$$

$$= \int_\Sigma \boxed{\frac{1}{\pi^p}} \boxed{h(W,\delta,G)} \boxed{\mathbf{1}_{\Sigma\in M^+(G)} |\Sigma|^{-(\delta+1+2p)} e^{-\mathrm{tr}(W+P)\Sigma^{-1}}} d\Sigma$$

$$= \boxed{\frac{1}{\pi^p}} \frac{\boxed{h(W,\delta,G)}}{\boxed{h(W+P,\delta+1,G)}}$$

Marginal Likelihood

REVIEW

- Generically:

$$\Sigma \mid G \sim HIW_c(\delta, W, G)$$
$$d \mid \Sigma \sim N_c(0, \Sigma)$$

$$\longrightarrow \quad p(d \mid G) = \boxed{\frac{1}{\pi^p}} \frac{\boxed{h(W, \delta, G)}}{\boxed{h(W + P, \delta + 1, G)}}$$

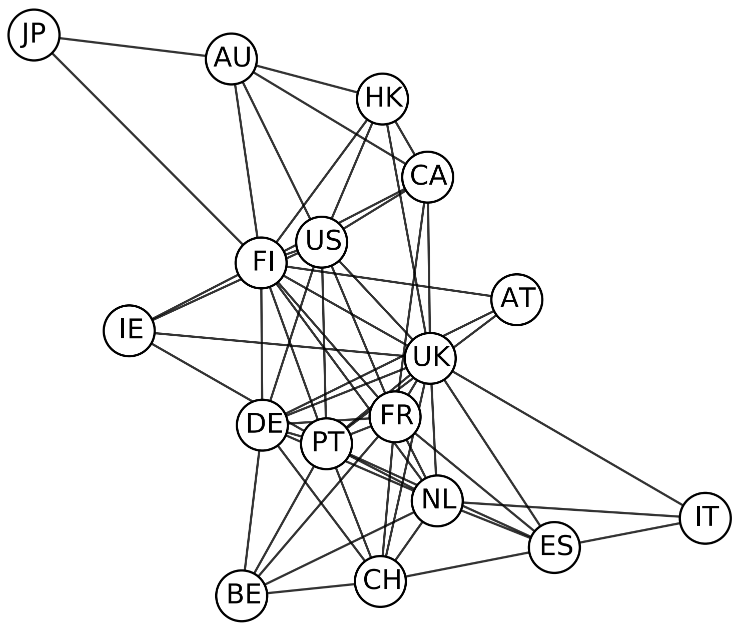
- For time series graph:

$$S_k \mid G \sim HIW_c(\delta_k, W_k, G)$$
$$d_k \mid S_k \sim N_c(0, S_k)$$

$$\longrightarrow \quad p(X_{1:p} \mid G) \approx \prod_{k=1}^{T} \boxed{\frac{1}{\pi^p}} \frac{\boxed{h(W_k, \delta_k, G)}}{\boxed{h(W_k + P_k, \delta_k + 1, G)}}$$

- For decomposable graphs
  - Prior decomposes over cliques *C* and separators *S*
  - $\boxed{h(W, \delta, G)}$ decomposes over cliques *C* and separators *S*
  - **Marginal likelihood** decomposes over cliques *C* and separators *S*

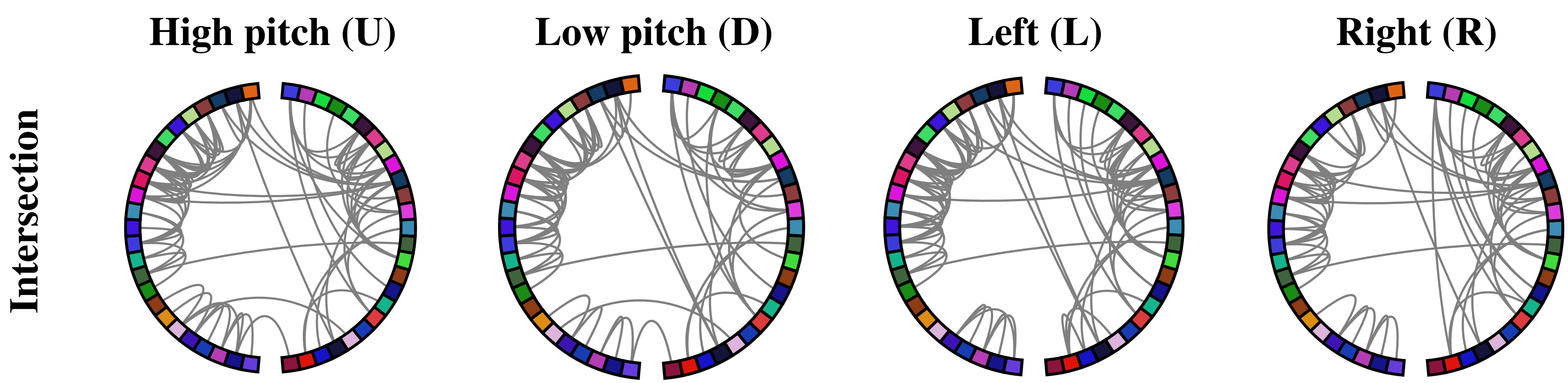
Global Stock Indices



i.i.d. graph

MEG Auditory Attention Task

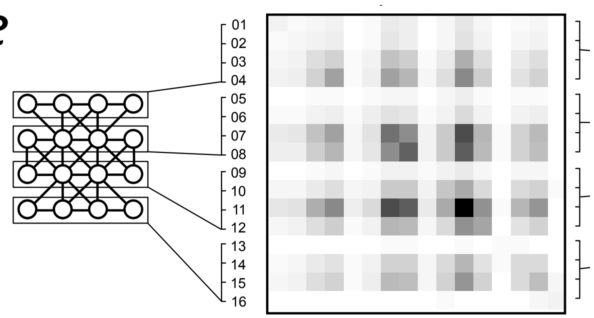
- Two tasks: (1) Focus attention, (2) Switch attention
- Four setups: high pitch, low pitch, left sound, right sound



Graphs of Time Series Summary

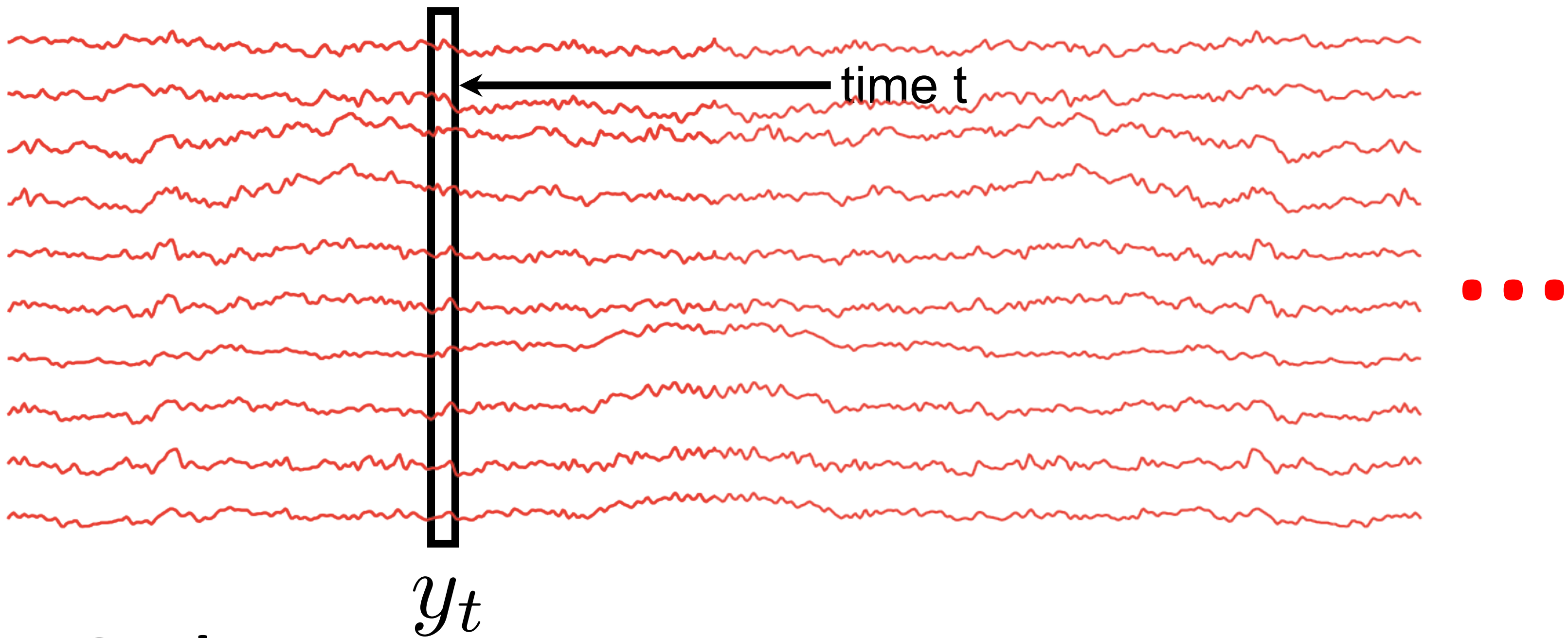
Σ^{-1} *sparse*

Gaussian
Graphical
Model



Zeros = no edge in graph,
Cond. ind. between nodes

- **Goal:** Infer conditional independencies between time series
- Efficient representation via spectral density matrix
  - Conditional independencies encoded by zeros in *inverse spectral density matrices*
- Whittle likelihood approximation defines tractable likelihood of data (Fourier coefficients) given spectral density matrices
- Defined hyper complex inverse Wishart prior
  - Conjugate prior on graph-constrained spectral density matrices
  - Enables closed-form marginal likelihood of data given graph



Goals:

- Evolution – Dynamics across time
- Relational structure – Dependencies between series

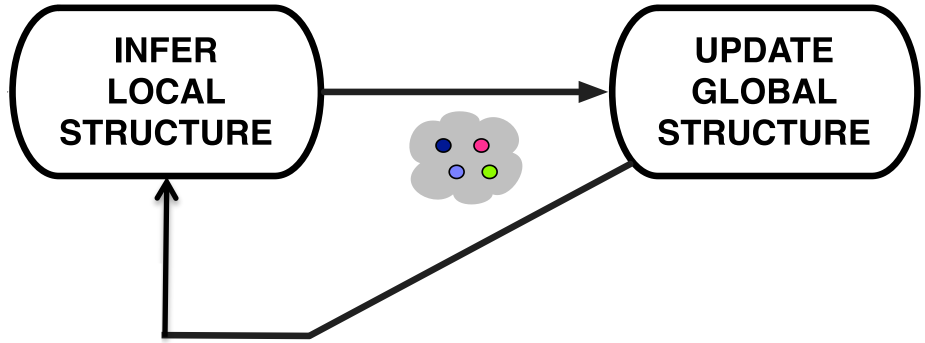
Modeling challenges:

- Large p – Many dimensions/series
- Irregular grid of observations
- Missing values
- Heterogeneous data sources
- …

Computational challenges:

- Large n – Long time series
- Streaming data – Continuum of observations

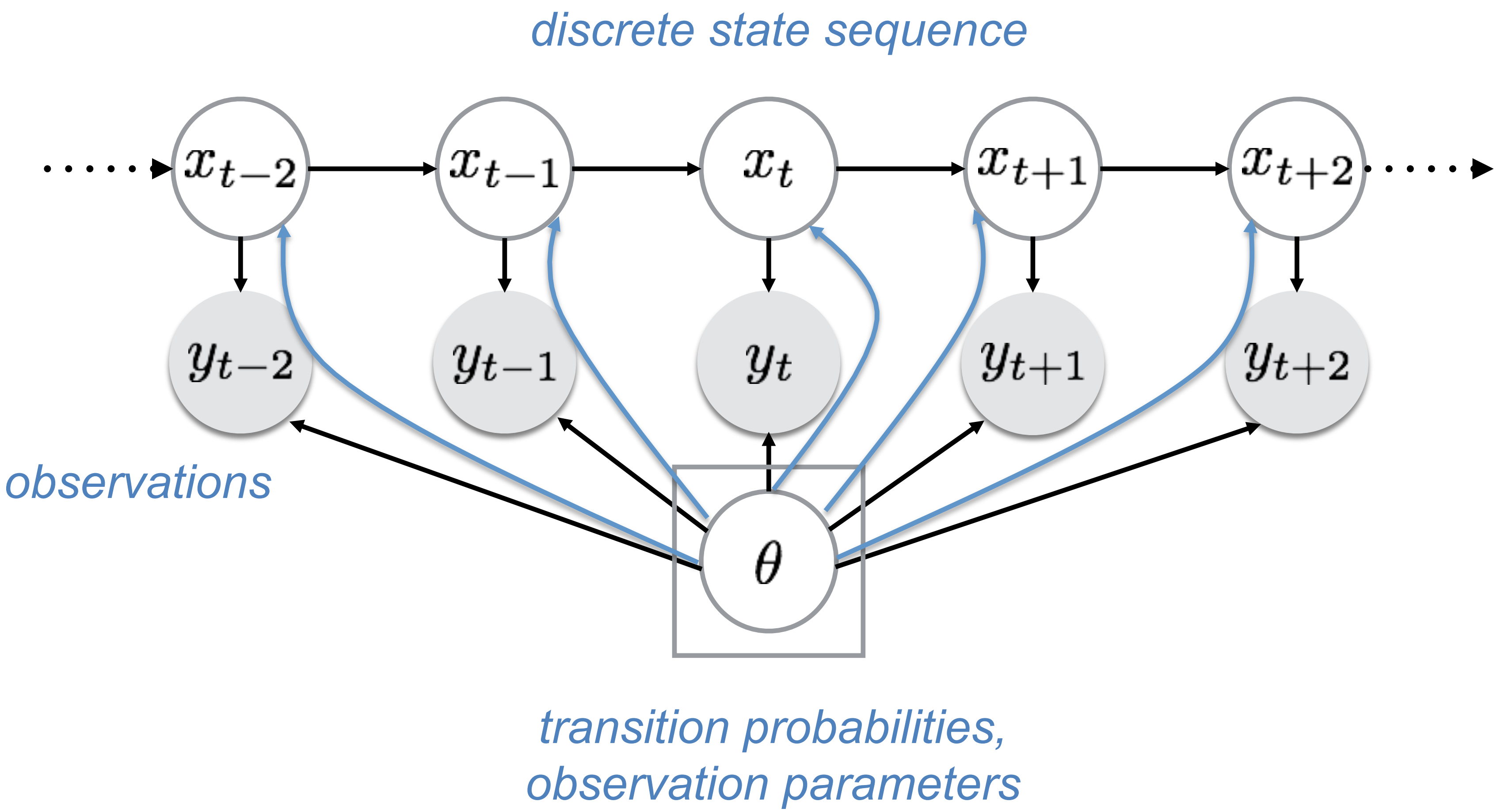
Minibatch-Based Algorithms



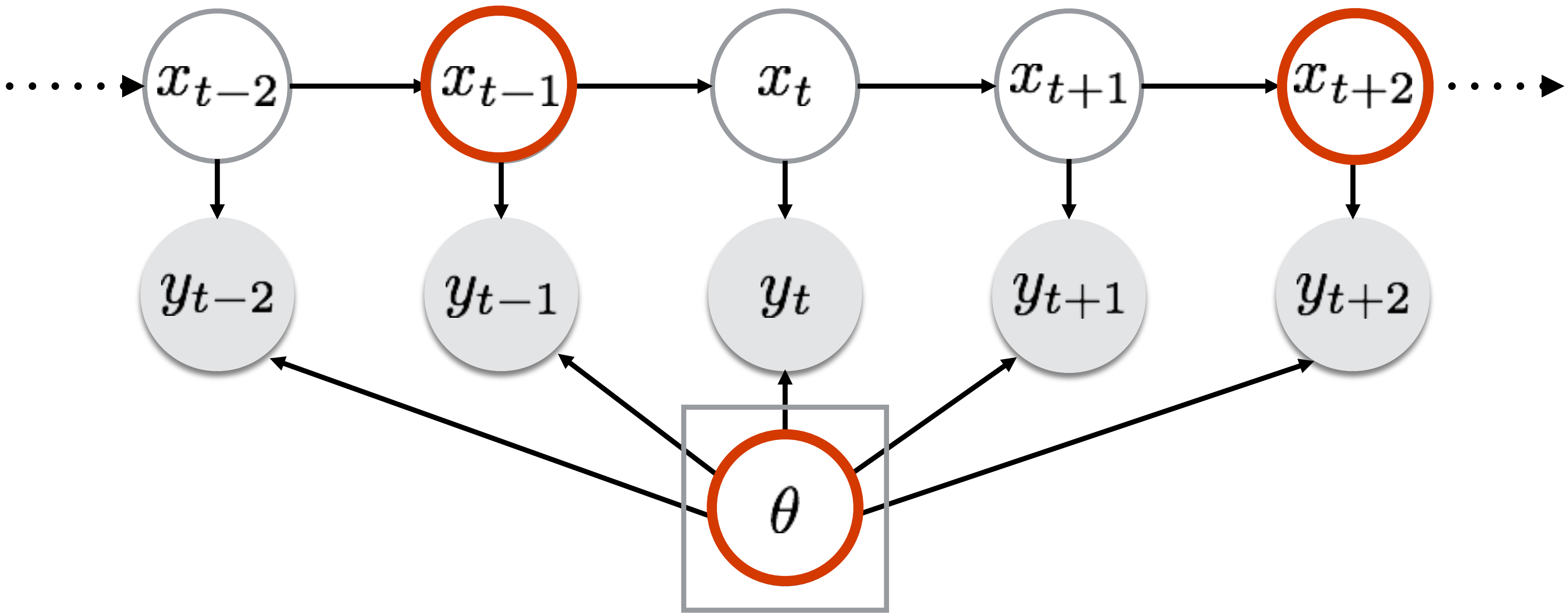
- Many ML/stat algorithms (e.g., gradient descent, Gibbs sampling,...) iterate between
  - operations involving all data
  - updating parameters
- Costly for large data / infeasible for streaming data
- Common approach for scalability:
  - *subsample data* → noisy operation
  - noisy update of parameters

Not appropriate for dependent data

Hidden Markov Models (HMMs)



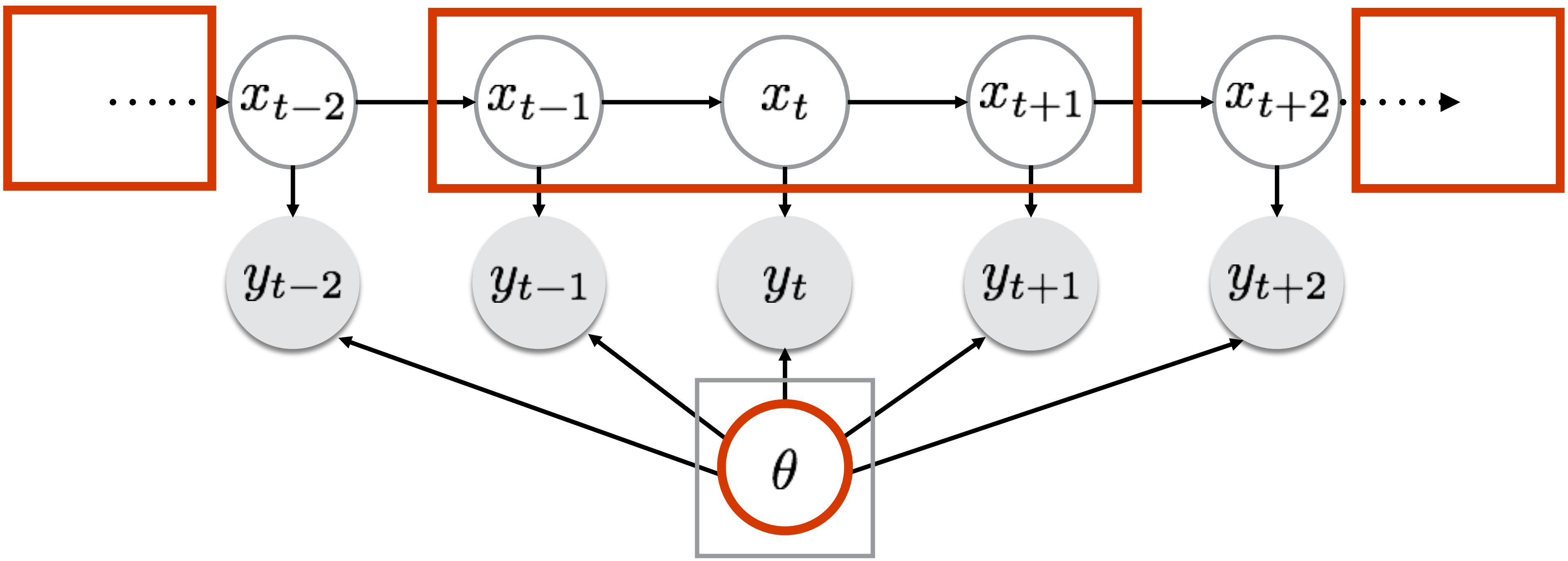
Minibatches for HMMs



- Why not just subsample observations independently?
- Cannot learn transition structure

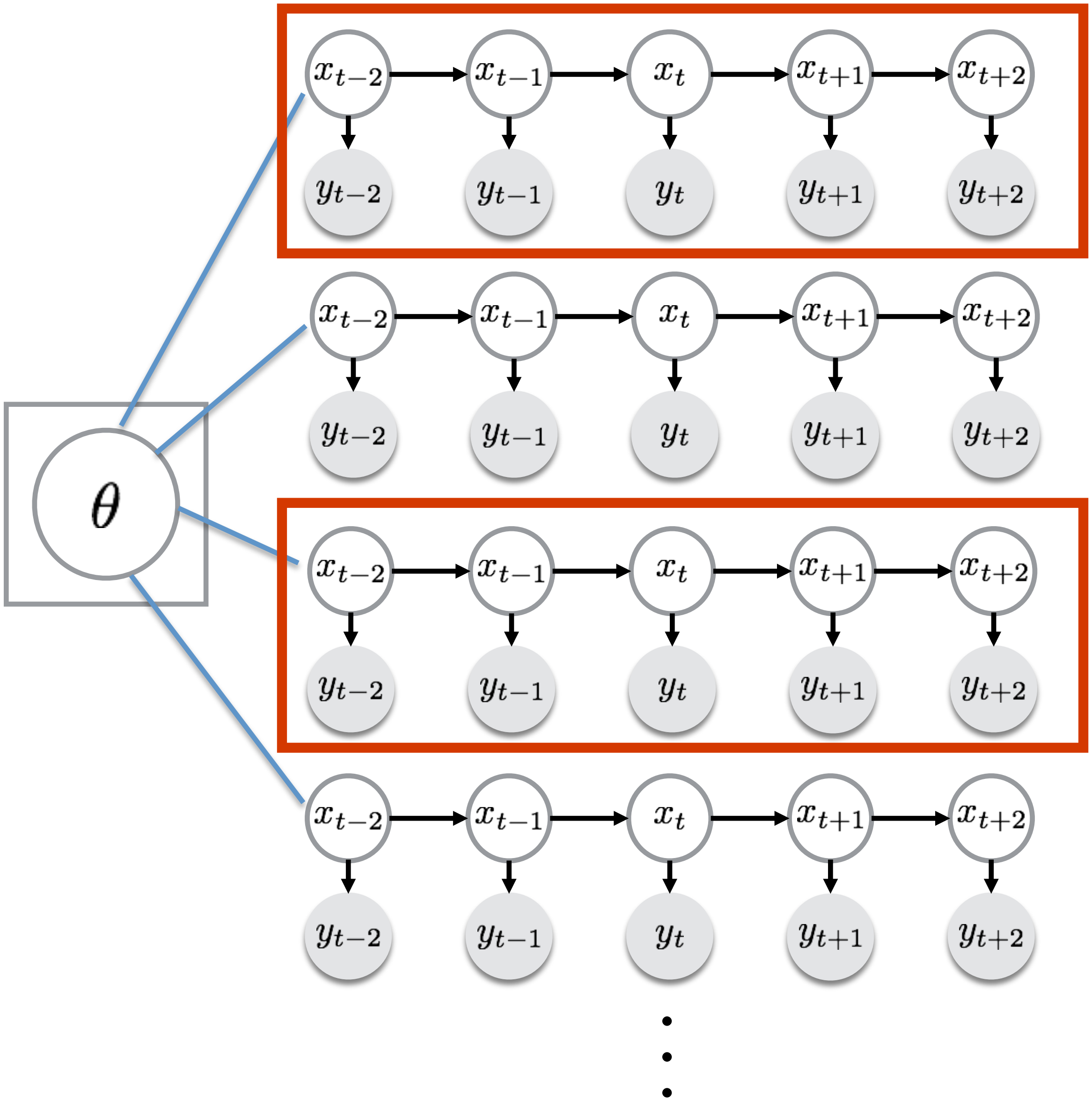
$$p(\mathbf{y}, \mathbf{x}, \theta) = p(\theta)\pi(x_1) \prod_{t=2}^{T} p(x_t \mid x_{t-1}, \theta_A) p(y_t \mid x_t, \theta_\phi)$$

Minibatches for HMMs



- How about sampling *subchain*? $x^S = (x_{t-L}, \ldots, x_t, \ldots, x_{t+L})$
- Do we just sever dependencies between subchains and analyze separately?

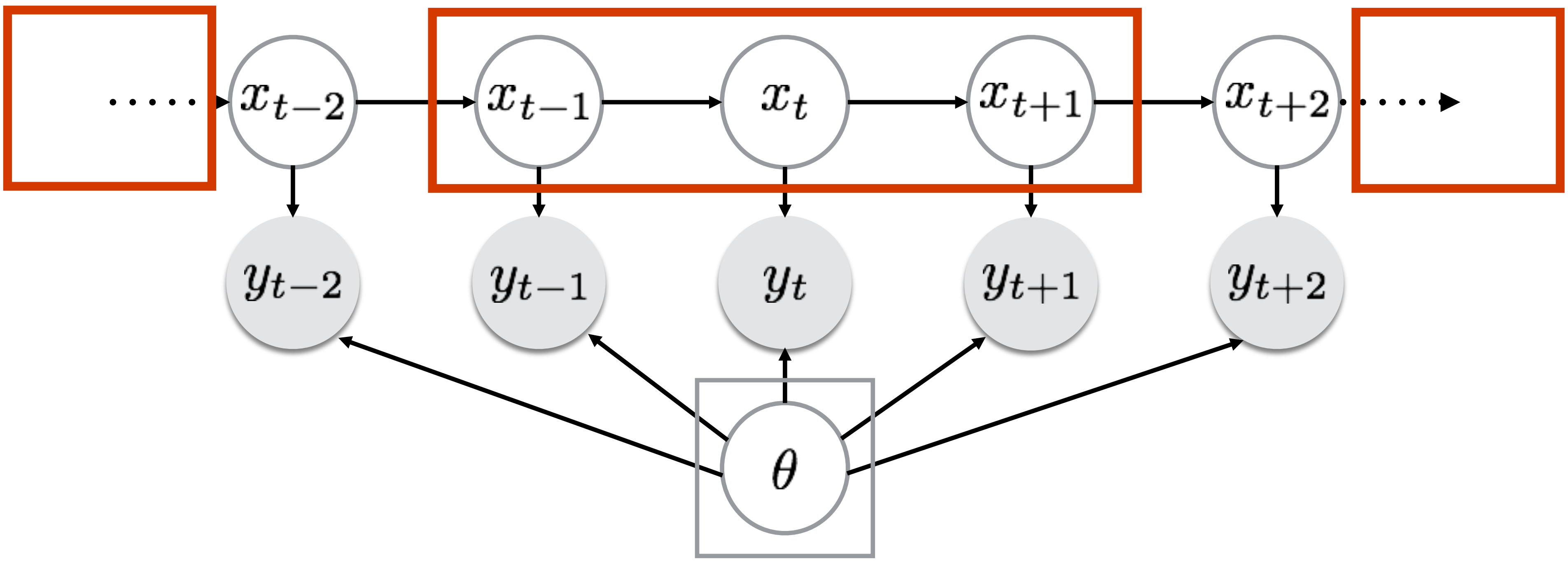
Large Collections of Short Chains



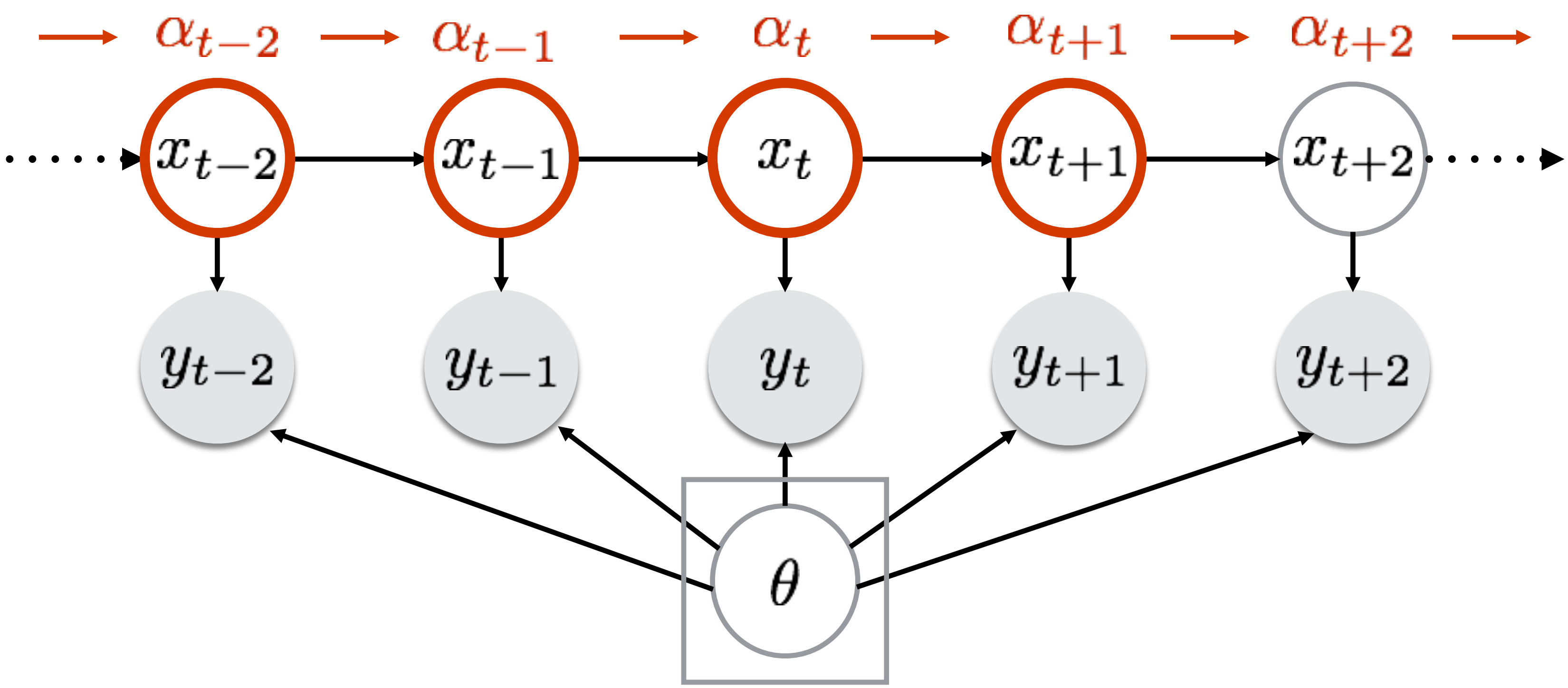
Johnson and Willsky, ICML 2014

Hughes et al., preprint

One Long Chain



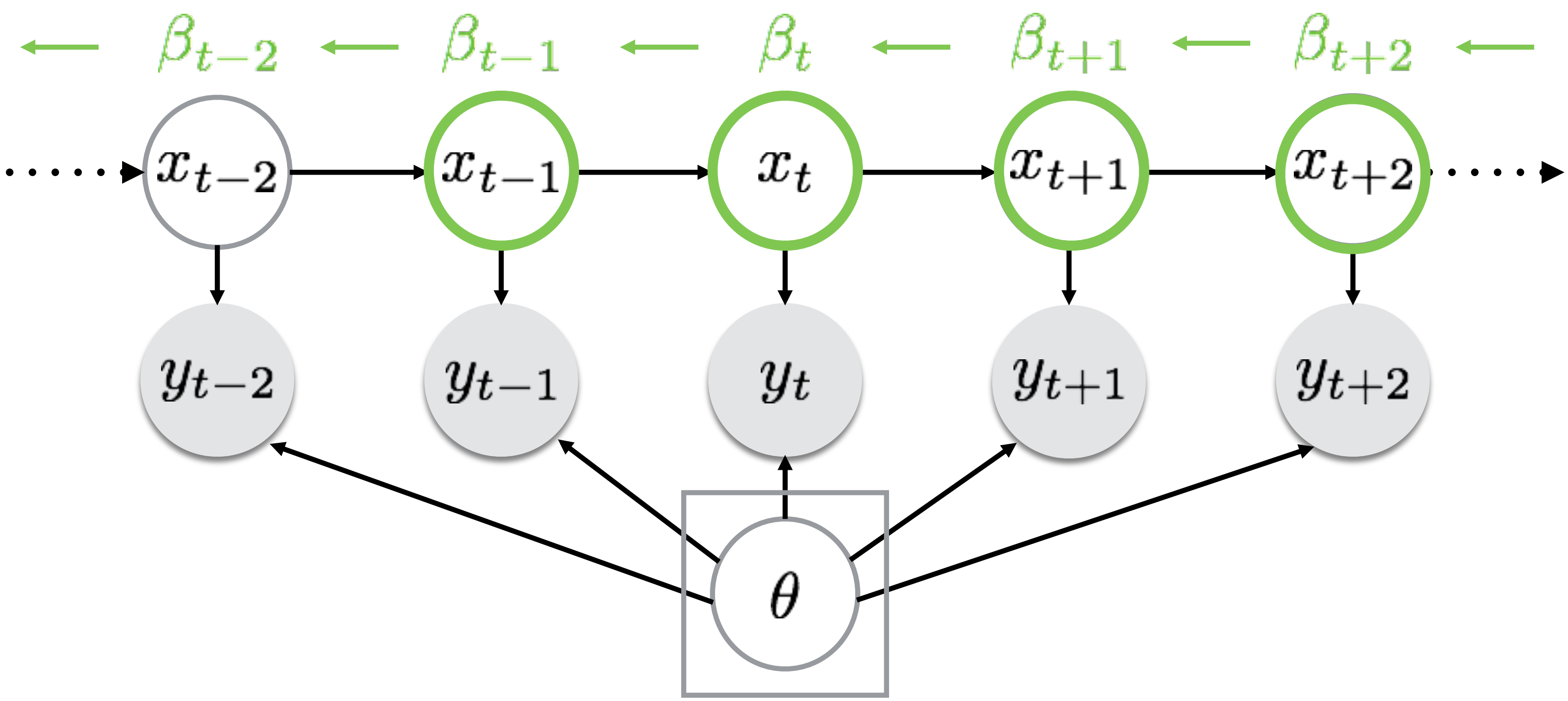
Batch Learning for HMMs



- Use current θ to form local state beliefs:
  - Propagate info forwards to form $\alpha_t = p(y_1, \ldots, y_t, x_t)$

$$\alpha_{t+1,k} = p(y_{t+1} \mid x_{t+1} = k) \sum_{j=1}^{K} \alpha_{t,j} p(x_{t+1} = k \mid x_t = j)$$

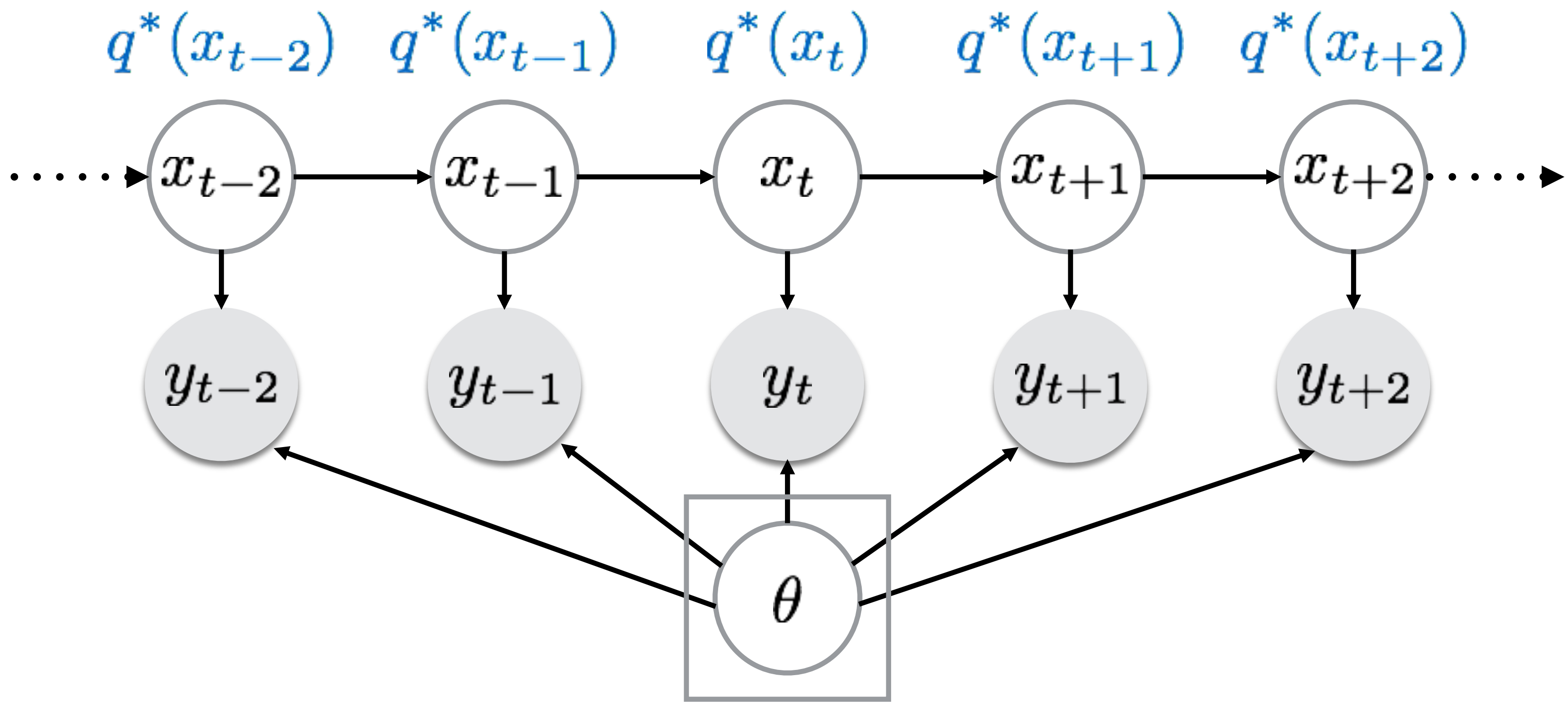
Batch Learning for HMMs



- Use current θ to form local state beliefs:
  - Propagate info backwards $\quad \beta_t = p(y_{t+1}, \ldots, y_T \mid x_t)$

$$\beta_{t,k} = \sum_{j=1}^{K} p(y_{t+1} \mid x_{t+1} = j) p(x_{t+1} = j \mid x_t = k) \beta_{t+1,k}$$

Batch Learning for HMMs

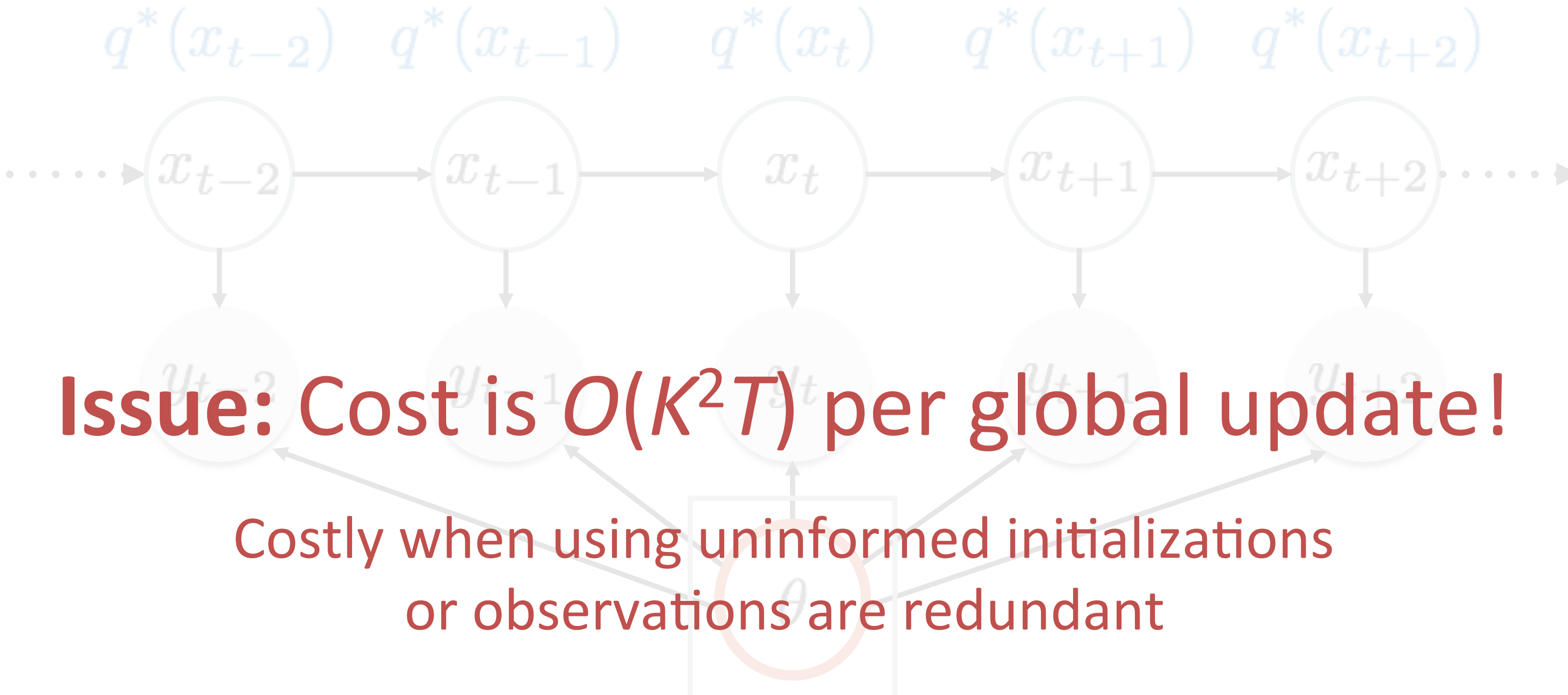


- Combine to form *smoothed* local state belief:

$$q^*(x_t) \propto \alpha_t \beta_t$$

$p(x_t \mid y_1, \ldots, y_T)$

Batch Learning for HMMs



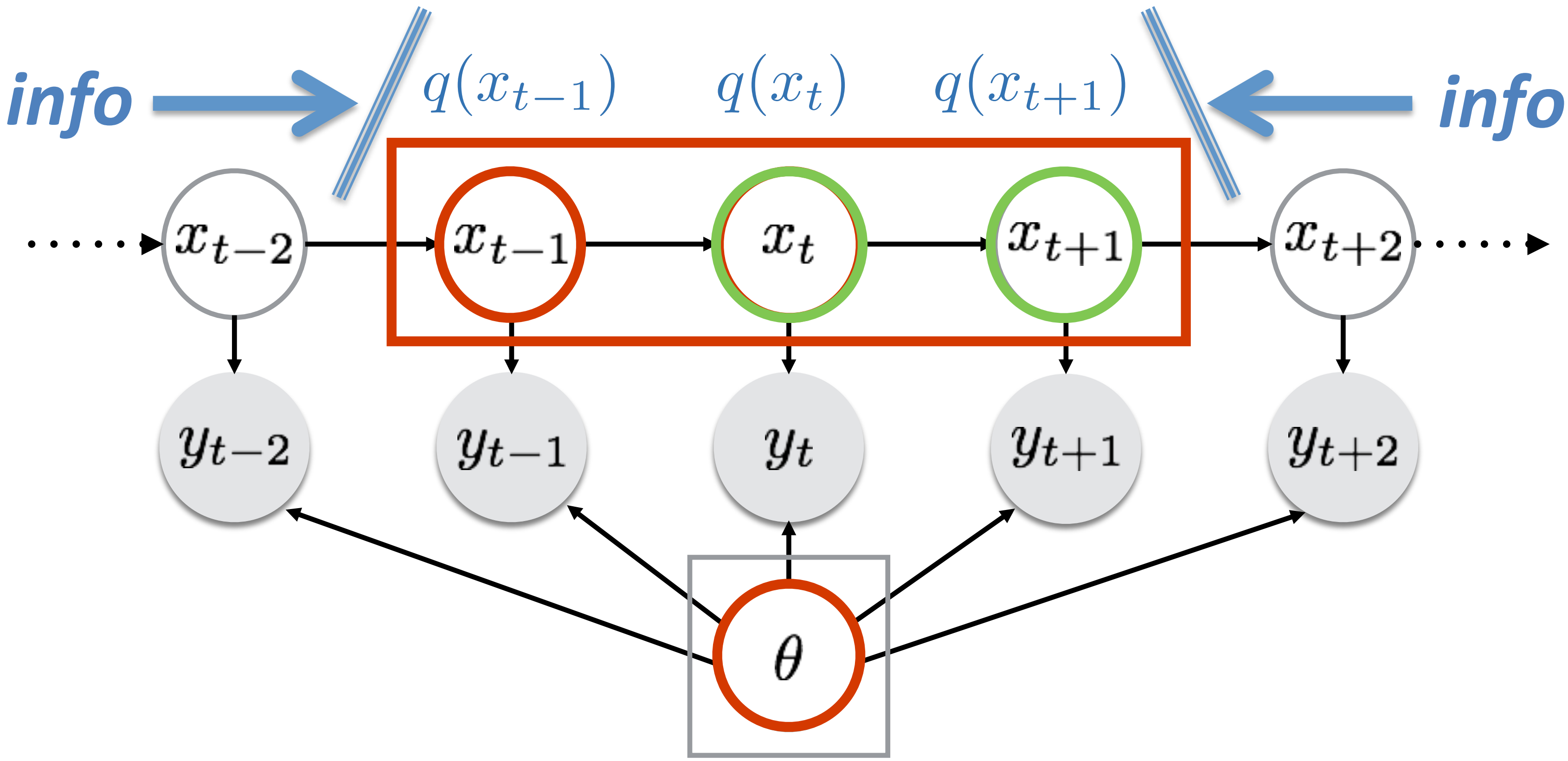
Issue: Cost is $O(K^2T)$ per global update!

Costly when using uninformed initializations
or observations are redundant

T = 250 million

- Given local beliefs, update global parameter

Minibatch Inference for HMMs

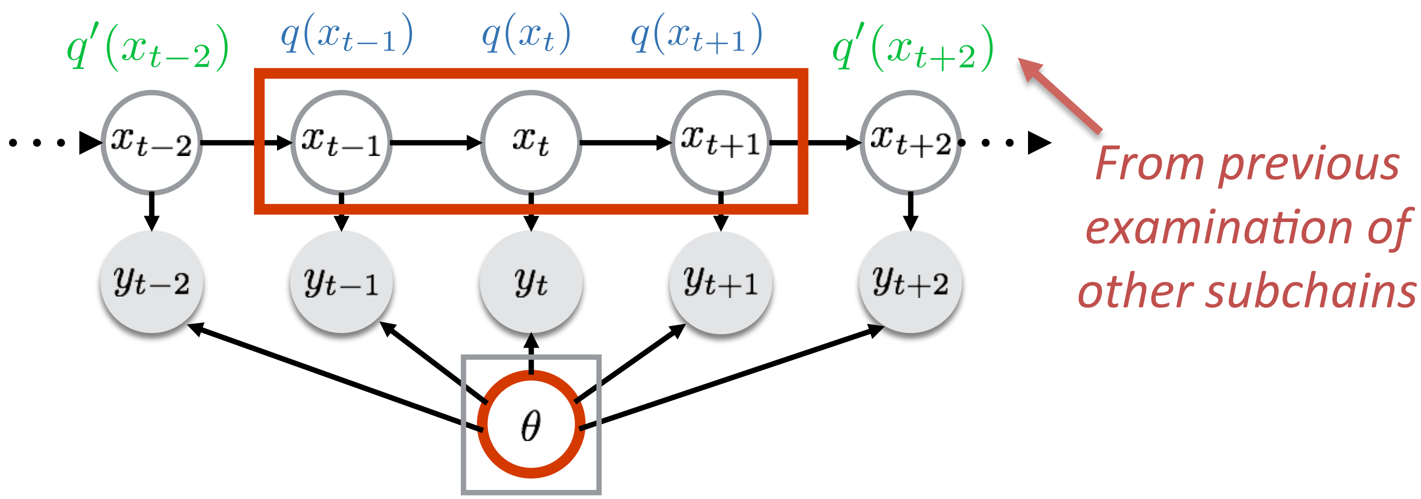


- Form **local** beliefs $q(x_t) \propto \tilde{\alpha}_t \tilde{\beta}_t$ → perform **global** update

Local forward message Local backward message

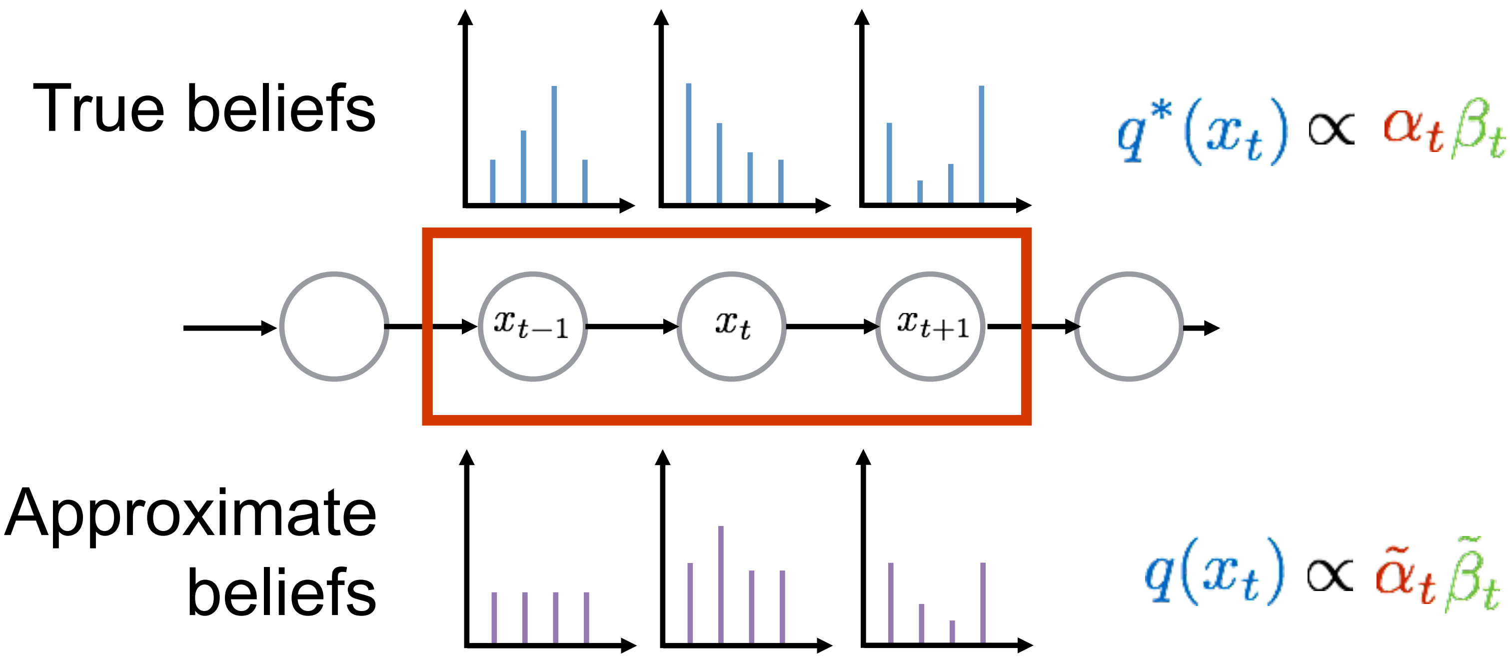
Storage Limitations

- Can local message passing harness previous beliefs on nodes outside the subchain? **NO!**



- T=250 M obs x K=25 latent states → 25 GB storage
- Need constant space algorithm
  → *can't remember past beliefs*

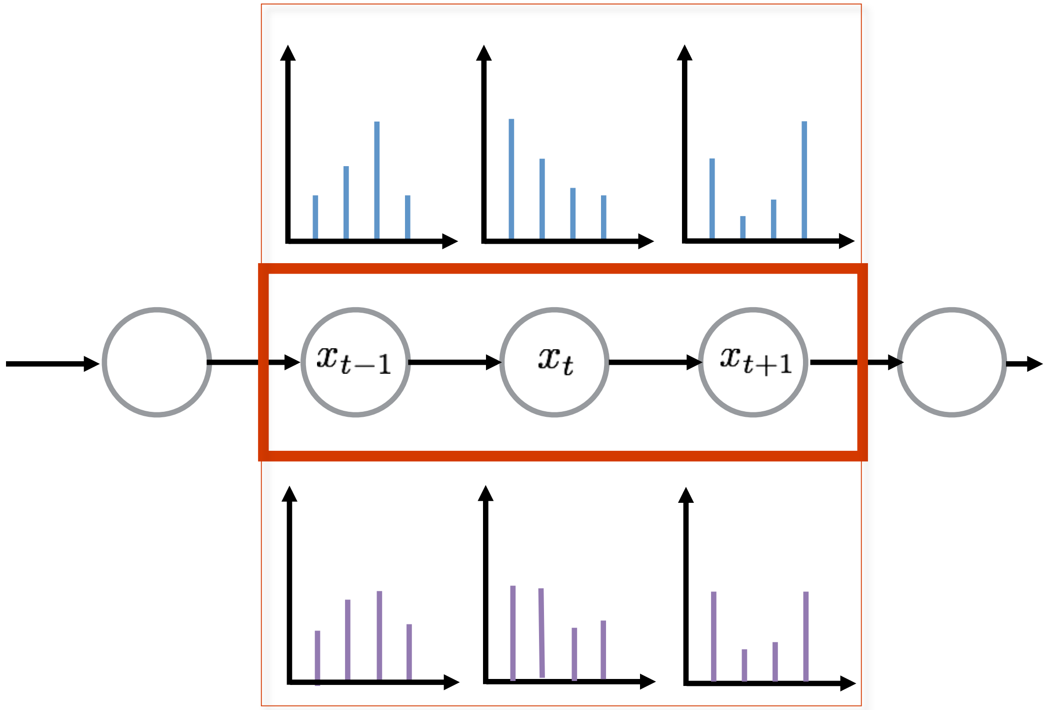
Harnessing Memory Decay



Do we expect x_t to influence $x_{t+1,000,000}$?

Leverage memory decay

Buffering Subchains



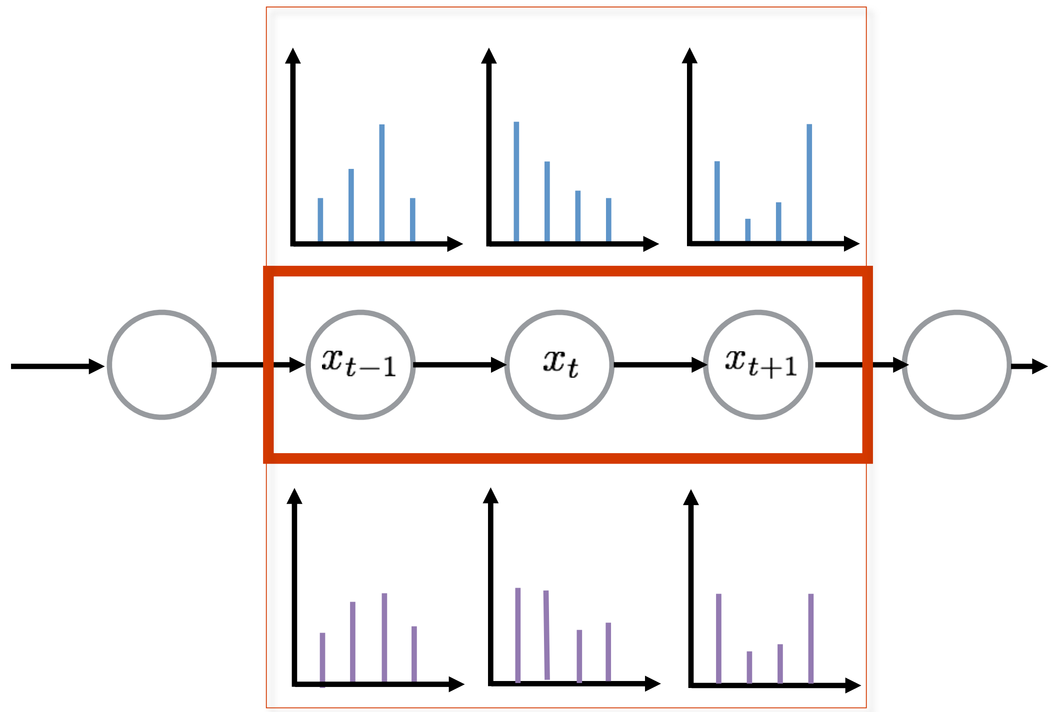
$q^*(x_t) \propto \alpha_t \beta_t$

$q(x_t) \propto \tilde{\alpha}_t \tilde{\beta}_t$

Check that subchain marginals are approximated well:

$$\max_{i \in S} ||q(x_i) - q^*(x_i)|| < \epsilon$$

Buffering Subchains



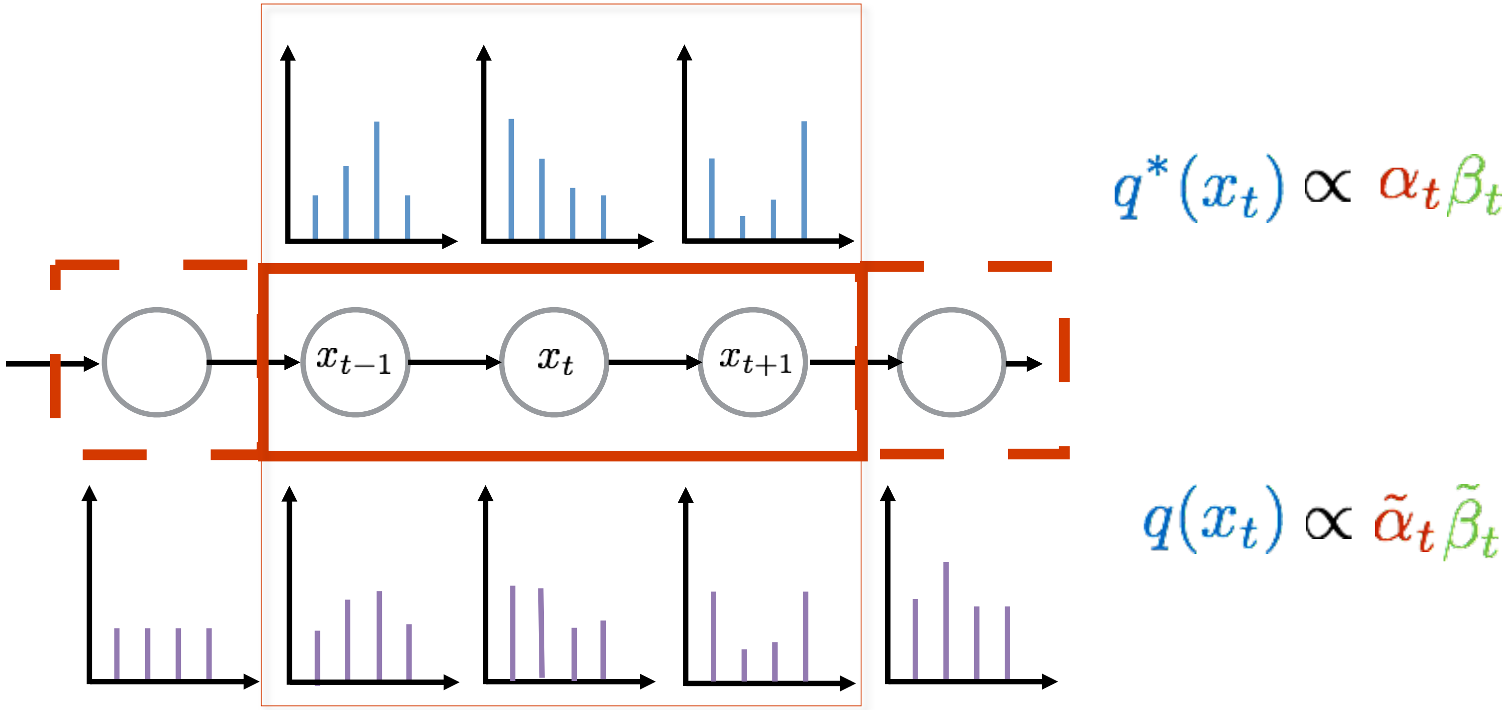
$$q^*(x_t) \propto \alpha_t \beta_t$$

$$q(x_t) \propto \tilde{\alpha}_t \tilde{\beta}_t$$

?

$$\max_{i \in S} ||q(x_i) - q^*(x_i)|| < \epsilon$$

Buffering Subchains



$$\max_{i \in S} ||q(x_i) - q^*(x_i)|| \overset{?}{<} \epsilon$$

Buffering Subchains

- **Only need limited buffer**
- **Complexity is now $O(K^2 L_{buffer})$ per iteration**

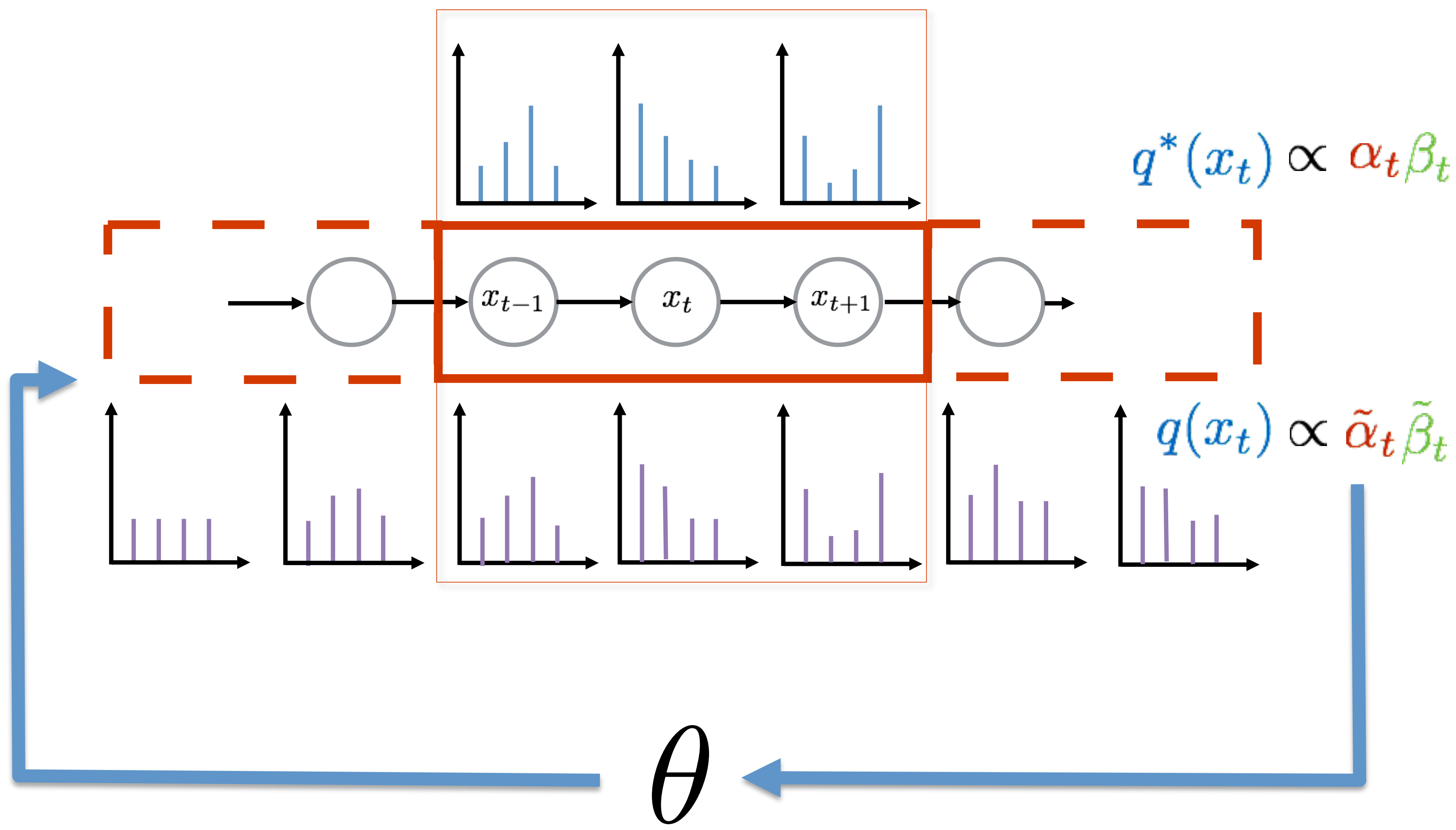
  Large savings for *L*+buffer << *T*

- **Similar idea as Splash BP (parallelizing BP)**

  [Gonzalez, et. al. 2009]

But, uncertain parameter setting here

Buffering for Learning

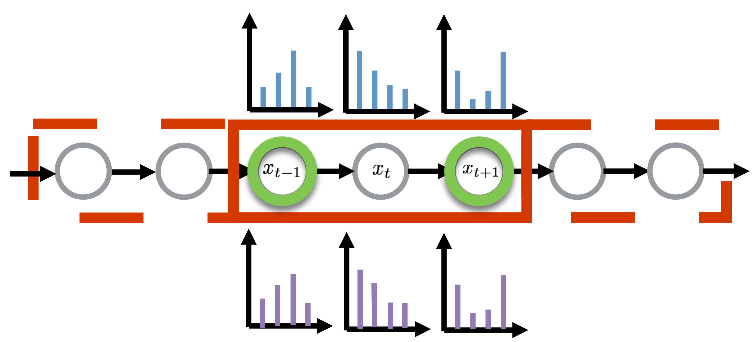


Buffering in Practice

- We do not actually know the true marginals
- Monitor changes in approximate subchain beliefs:

$$\max_{i \in S} \left|\left| q(x_i)^{\text{new}} - q(x_i)^{\text{old}} \right|\right| < \epsilon$$

- Chain structuring implies that only endpoints must be checked



- During buffer expansions, forward-backward passes can reuse computations of previous buffer

Variational Bayes (VB)

- Approximate posterior with variational distribution

$$p(\underset{\uparrow \atop \text{latent variables}}{x}, \overset{\text{parameters} \atop \downarrow}{\theta} | \underset{\uparrow \atop \text{observations}}{y}) = \frac{p(y|x,\theta)p(x,\theta)}{p(y)} \approx q(x,\theta)$$

- Minimize $\mathrm{KL}(q||p) \leftrightarrow$ maximize "ELBO":

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(y,x,\theta)] - \mathbb{E}_q[\log q(x,\theta)] \leq \log p(y)$$

- Common to make mean-field assumption:

$$q(x,\theta) = q(x)q(\theta)$$

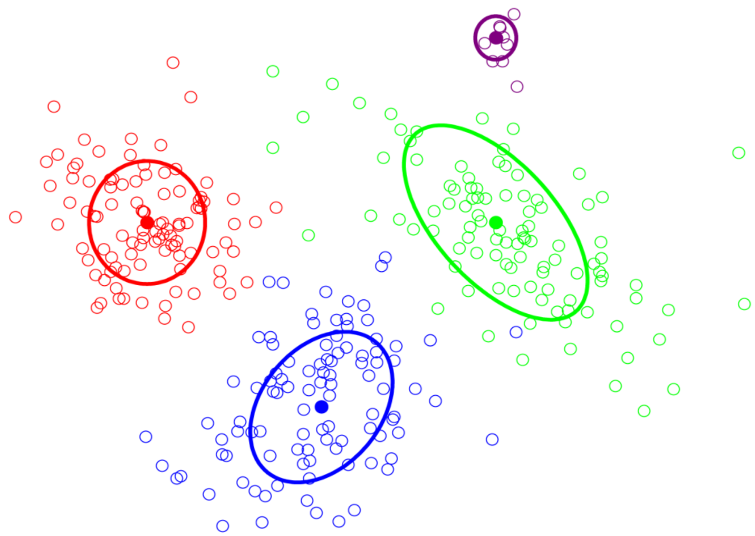
Variational Methods Cartoon

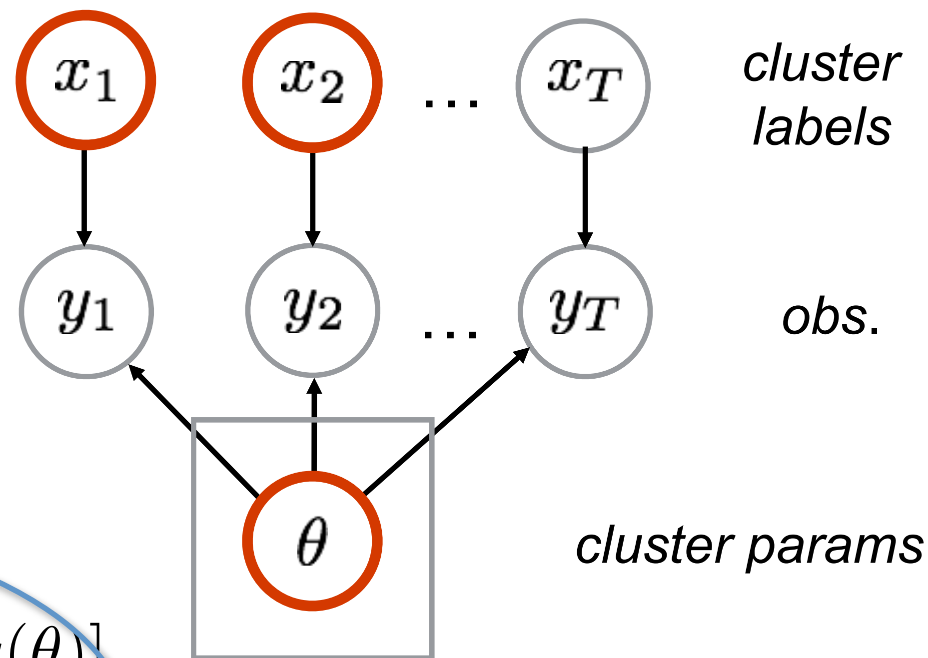
- Cartoon of goal:

- Variational distribution parameterized by *variational free parameters*
- *Objective: optimize over free parameters to find "closest" distribution in variational family*

VB Example: Mixture of Gaussians

Maximize ELBO with *coordinate-ascent* $\frac{\partial \mathcal{L}}{\partial q(\mathbf{x})} = 0 \leftrightarrow \frac{\partial \mathcal{L}}{\partial q(\theta)} = 0$





$$\mathcal{L} = E_{q(\theta)}\left[\ln p(\theta)\right] - E_{q(\theta)}\left[\ln q(\theta)\right] + \sum_{i=1}^{T} E_{q(x_i)}\left[\ln p(y_i, x_i|\theta)\right] - E_{q(x_i)}\left[\ln q(x_i)\right]$$

Stochastic Variational Inference (SVI)

- Batch VB global step requires touching *all of the data*

$$\mathcal{L} = E_{q(\theta)}\left[\ln p(\theta)\right] - E_{q(\theta)}\left[\ln q(\theta)\right]$$
$$+ \boxed{\sum_{i=1}^{T} E_{q(x_i)}\left[\ln p(y_i, x_i | \theta)\right]} - E_{q(x_i)}\left[\ln q(x_i)\right]$$

- SVI uses stochastic gradient descent (SGD) for global update [Hoffman, et. al. 2013]
  - Sample observation: $x^S \sim \mathrm{Unif}(x_1, \ldots, x_T)$
  - Follow noisy, unbiased estimate of natural gradient of $\mathcal{L}$.

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \rho_t \tilde{\nabla}_{\mathbf{w}} \mathcal{L}^S \qquad \mathbb{E}_S[\tilde{\nabla}_{\mathbf{w}} \mathcal{L}^S] = \tilde{\nabla}_{\mathbf{w}} \mathcal{L}$$

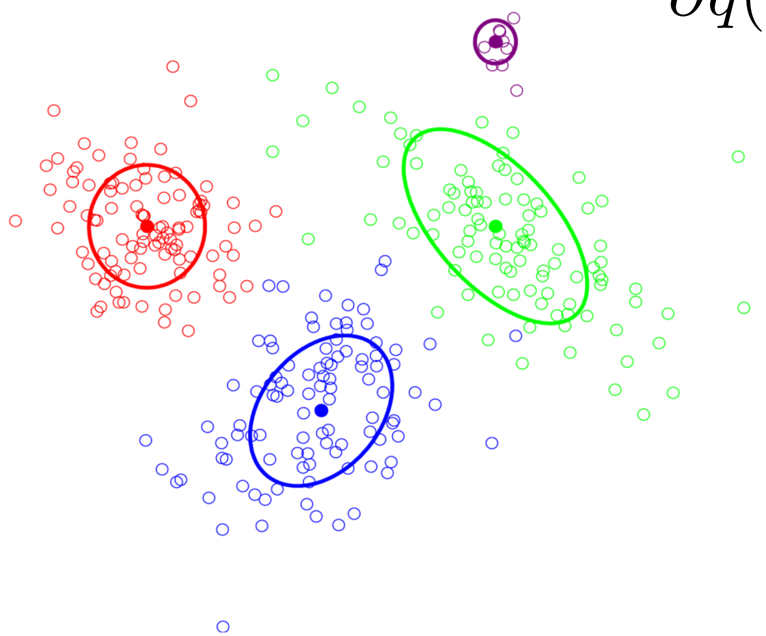
Variational parameters defining $q(\theta)$

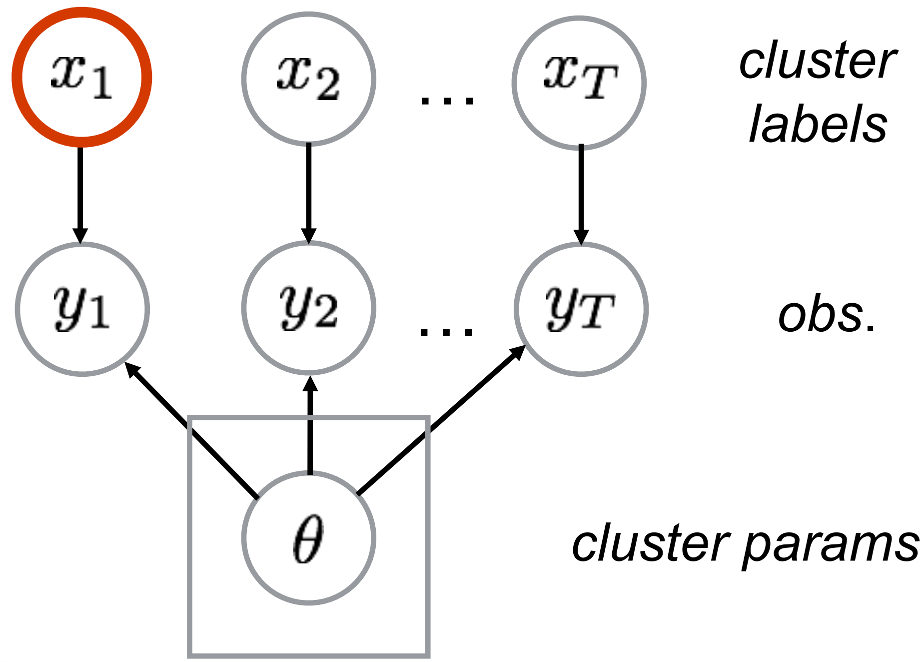
SVI Example: Mixture of Gaussians

Maximize ELBO with *stochastic gradient descent*

$$\frac{\partial \mathcal{L}^s}{\partial q(x_s)} = 0 \longleftrightarrow \mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \rho_t \tilde{\nabla}_{\mathbf{w}} \mathcal{L}^S$$

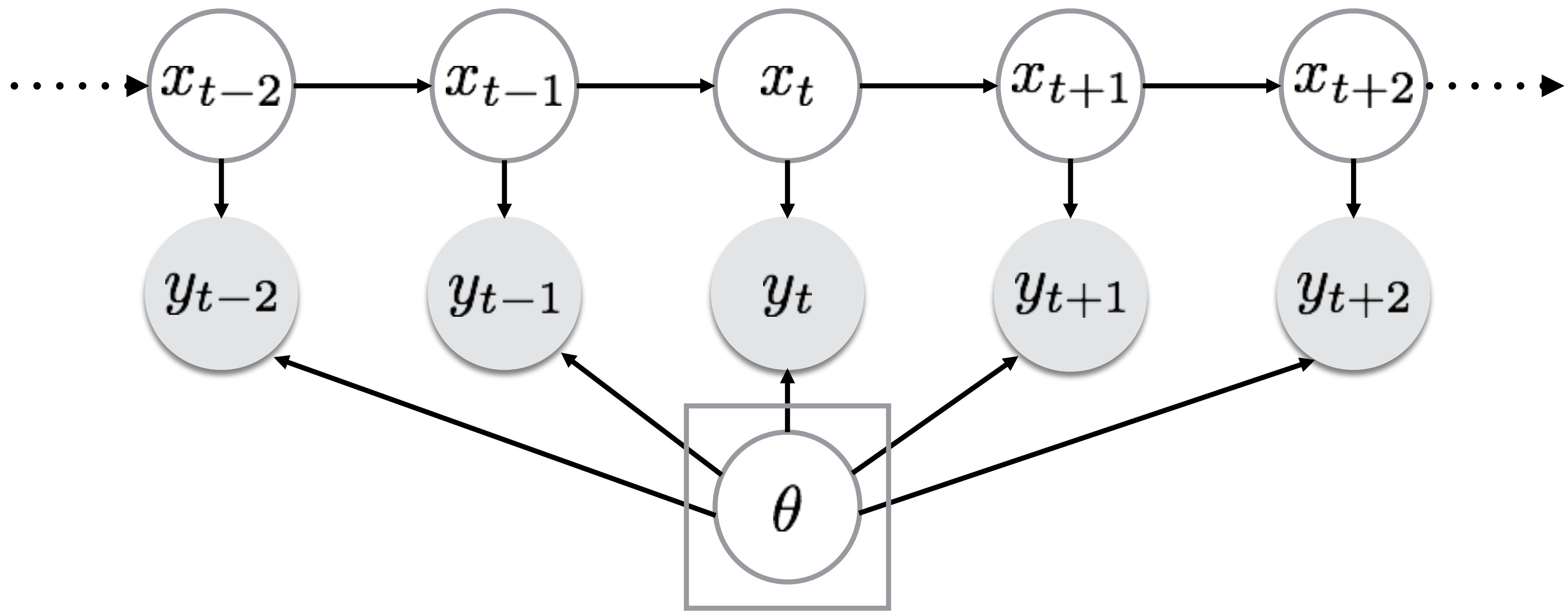
Hyperparams for q(θ)





$$\mathcal{L}^s = E_{q(\theta)}\left[\ln p(\theta)\right] - E_{q(\theta)}\left[\ln q(\theta)\right] + T \cdot \left(E_{q(x_s)}\left[\ln p(y_s, x_s|\theta)\right] - E_{q(x_s)}\left[\ln q(x_s)\right]\right)$$

Structured Mean Field Approximation



- Use structured mean-field approximation:

$$p(x_1, x_2, \ldots, x_T, \theta \mid y_1, y_2 \ldots, y_T) \approx q(x_1, x_2, \ldots, x_T)q(\theta)$$

SVI for HMMs

(Approx) coordinate ascent step:

Function of $q(\theta)$

$$\alpha_{t+1,k} = \tilde{p}(y_{t+1} \mid x_{t+1} = k) \sum_{j=1}^{K} \alpha_{t,j} \tilde{A}_{j,k}$$

$$\beta_{t,k} = \sum_{j=1}^{K} \tilde{p}(y_{t+1} \mid x_{t+1} = j) \tilde{A}_{k,j} \beta_{t+1,k}$$

$$q(\theta)$$

Stochastic natural gradient step:

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \rho_t \tilde{\nabla}_{\mathbf{w}} \mathcal{L}^S$$

Function of $q(\mathbf{x})$

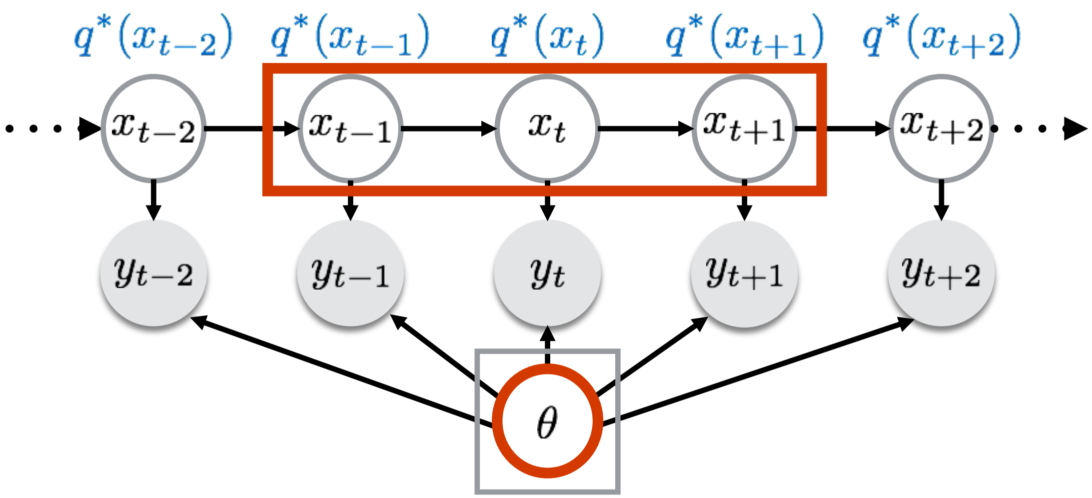
Foti, Xu, Laird, Fox, *NIPS 2014*

Differences from i.i.d. Case

- Minibatches are *correlated*
  - Data in one is not independent of data in another
- Minibatch marginals ≠ batch marginals
  - Impact of latent chain
  - Mitigated by buffering

Correlated Minibatches

- Pretend we have exact local distribution $q^*(x^S)$



As if we had run batch forward-backward

- Typical arguments for convergence to local mode rely on *unbiased* + *independent* noisy gradients [c.f., Bottou 1998, Hoffman 2013]
  - Our SGs are *dependent* since subchains are correlated
- Using [Polyak and Tsypkin 1973], unbiasedness suffices for convergence of $\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \rho_t \tilde{\nabla}_{\mathbf{w}} \mathcal{L}^S$

Global Update – Unbiasedness

- In mixture model case with **uniform** sampling of **observation *s***, unbiasedness was preserved via:

$$\mathcal{L}^s = E_{q(\theta)}\left[\ln p(\theta)\right] - E_{q(\theta)}\left[\ln q(\theta)\right] + T \cdot \left(E_{q(x_s)}\left[\ln p(y_s, x_s|\theta)\right] - E_{q(x_s)}\left[\ln q(x_s)\right]\right)$$

- In HMM case, our ELBO data term is

$$\ln p(\mathbf{y}, \mathbf{x}|\theta) = \ln \pi(x_1) + \sum_{t=2}^{T} \ln A_{x_{t-1},x_t} + \sum_{i=1}^{T} \ln p(y_t|x_t)$$

  – Does not decompose over individual x_t

  – Need to scale transition and emission terms separately

- Straightforward for **uniform** sampling of **subchains *S*** of length *L*, assuming chain is observed at *stationarity*

Effect of Approximated Marginals

SVI-HMM iterates:

buffer minibatches to **approx *q(x)*** $\longleftrightarrow$ **update *q(Θ)***

coordinate gradient step — *stochastic (natural) gradient step*

For ϵ sufficiently small (sufficiently long buffer)

- Approximate marginals "close enough" to true marginals
- Noisy gradient in same half-plane as true gradient

$$\Downarrow$$

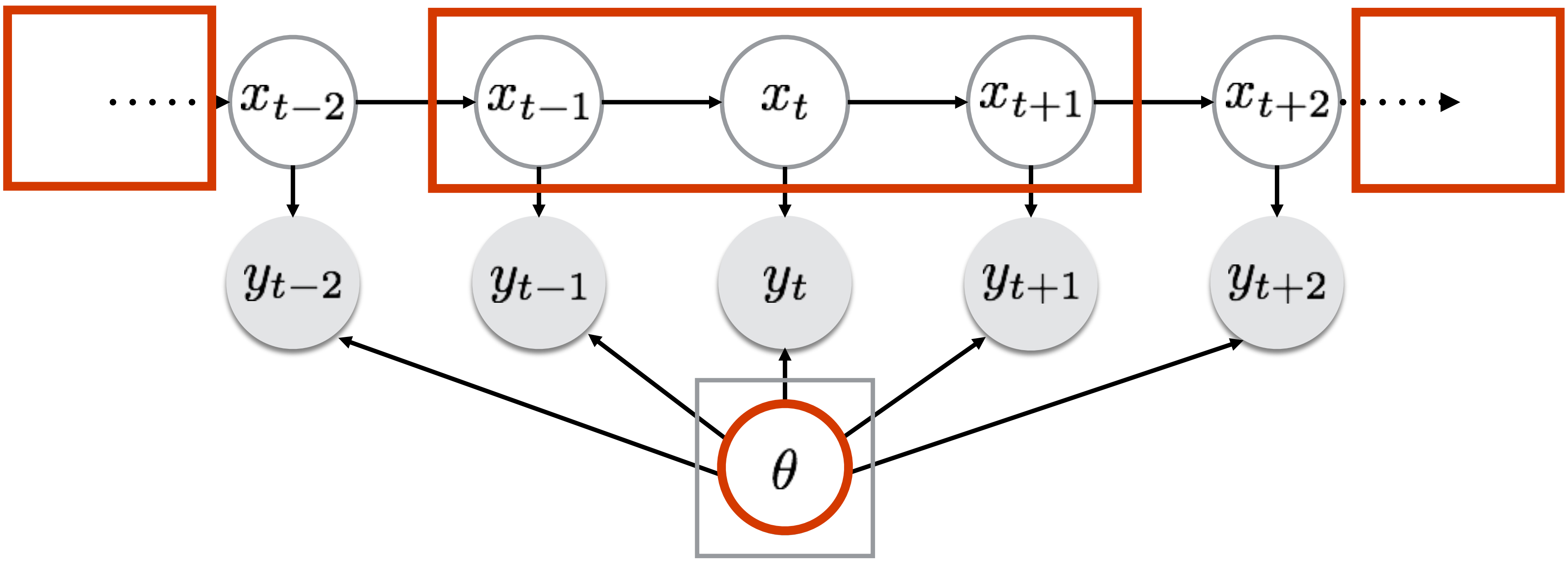
iterative algorithm converges to local mode of ELBO

Foti, Xu, Laird, Fox, *NIPS 2014*

Experiments

- Synthetic data:
  - **Diagonally Dominant**: Long memory chain with large self-transitions
  - **Reversed Cycles**: Two overlapping cycles with opposite directions
- **Human chromatin application**

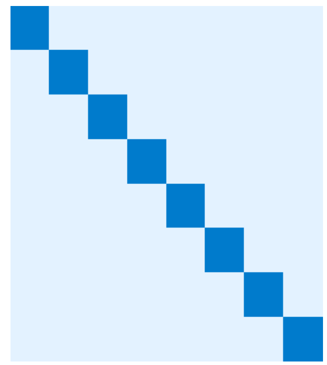
Minibatch of Subchains

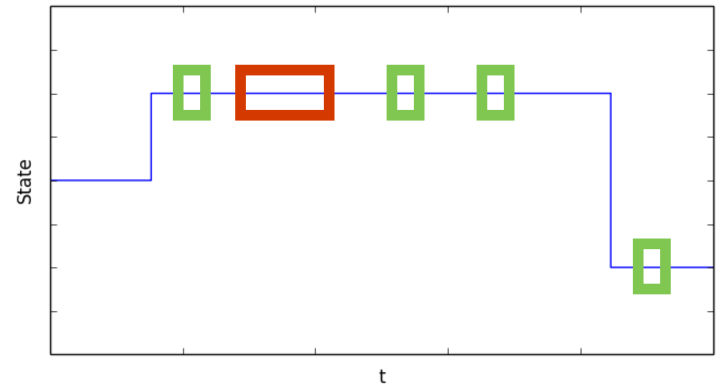


Minibatch consists of *M* subchains each of length *L*

Diagonally Dominant

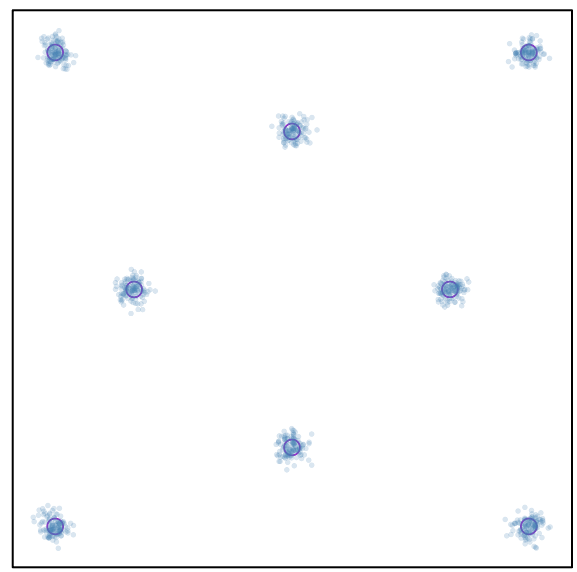
- 8 latent states
- 2d Gaussian emissions





- *High auto-correlation*
  → few long subchains
  converge slowly
  (small *M*, large *L*)

- *Emissions identifiable*
  → many small subchains
  perform better
  (large *M*, small *L*)

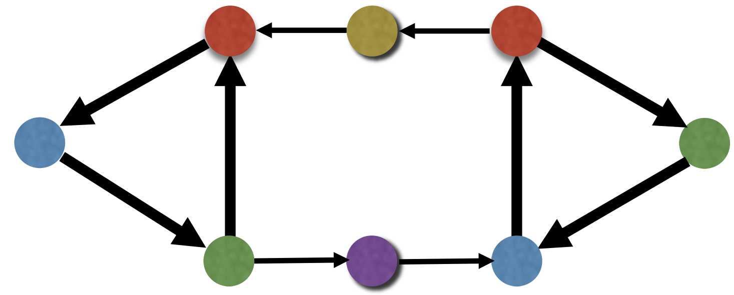


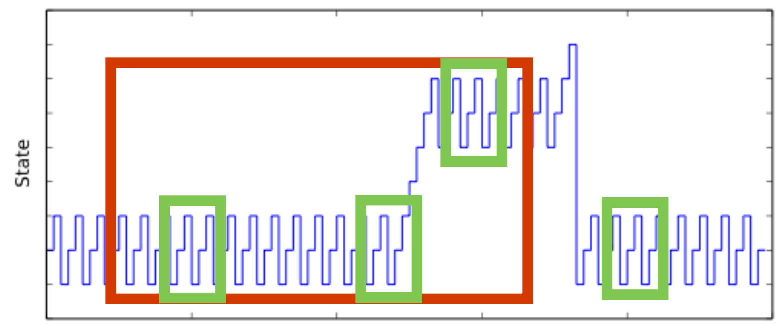
Reversed Cycles

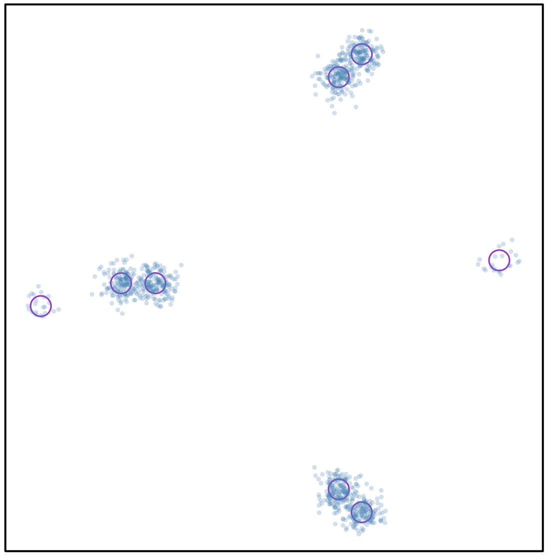
- 8 latent states
- 2d Gaussian emissions

- *Emission distributions overlap*
- *Direction* of cycles important to identify states
  - Singleton observations insufficient
  - Without buffering, need $L > 3$ to learn effectively

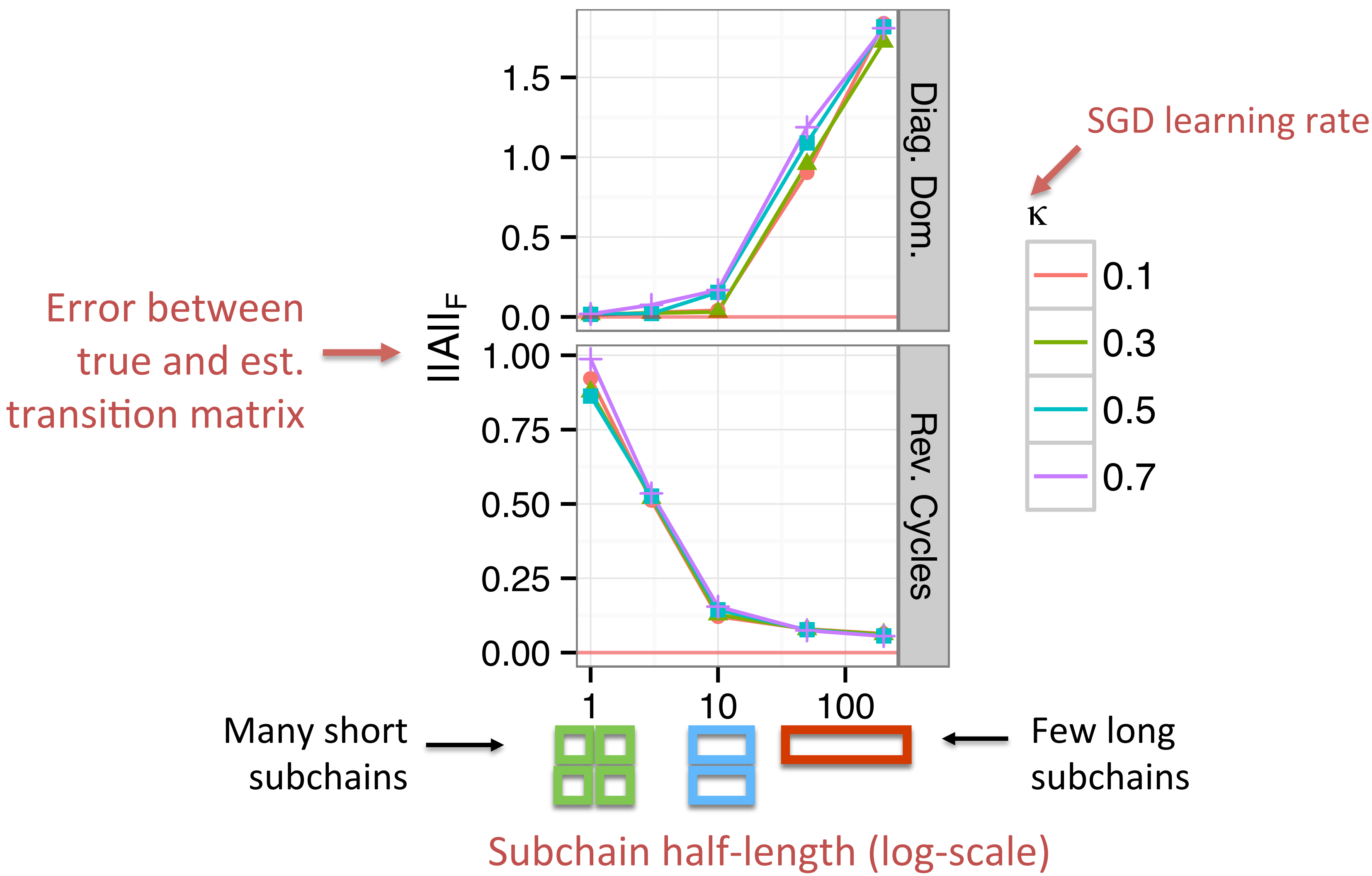
- Longer subchains more likely to capture structure



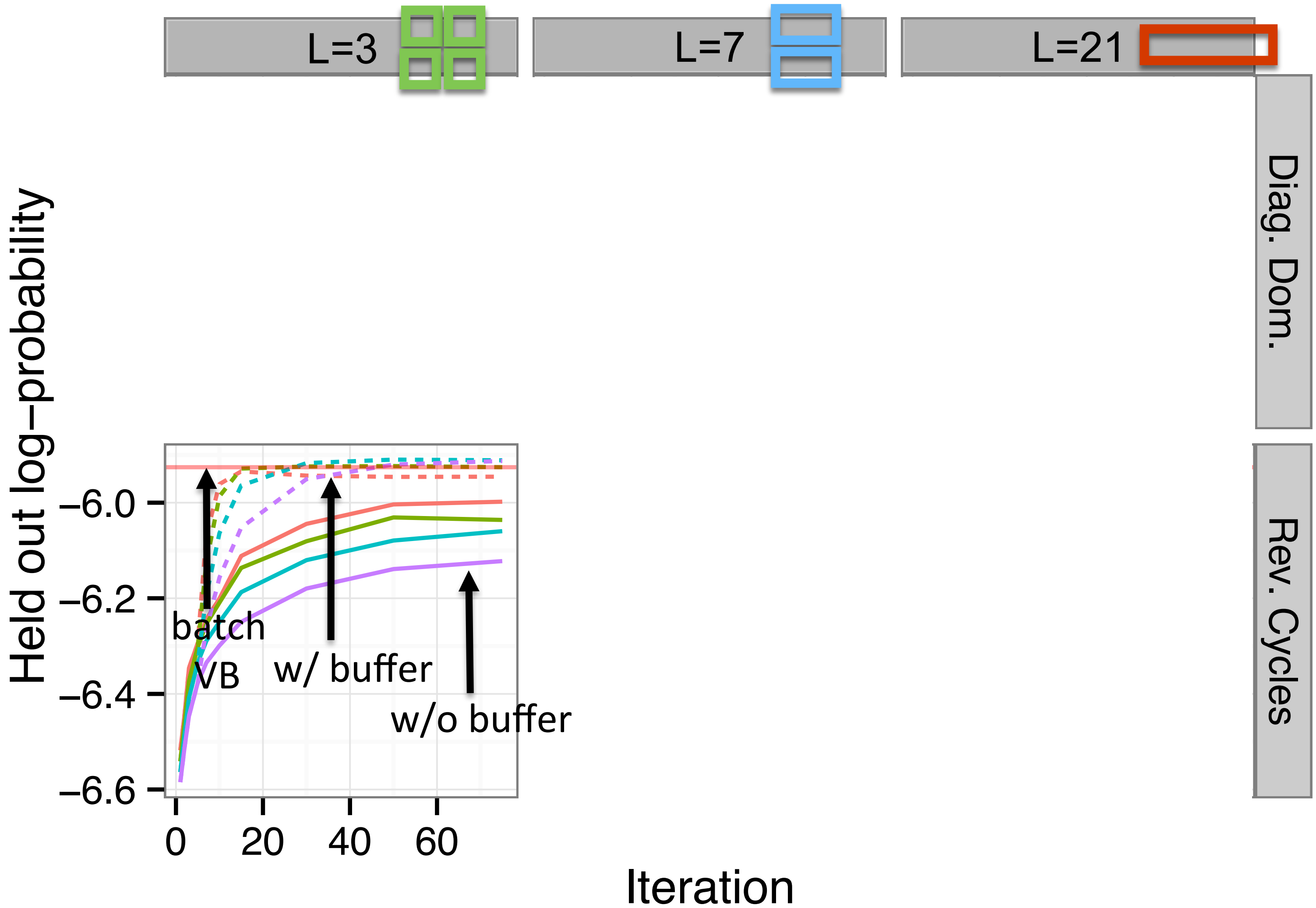




Transition Matrix Recovery

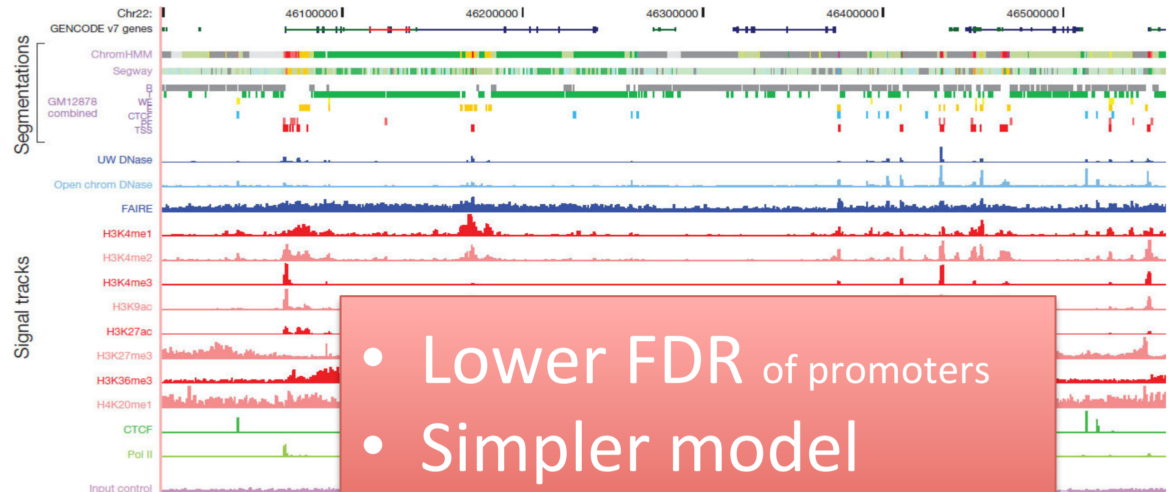


Subchain Buffering



Human Chromatin Segmentation

- Chromosome data from ENCODE project
- 12 dimensional observations
- **Goal:** segment sequences
- **T = 250 million**



- Lower FDR of promoters
- Simpler model
- Uses all of the data

- [Hoffman et. al. 2012] used **dynamic Bayesian network**
  - Broke sequence into pieces to perform inference via EM
  - Severs long-range dependencies

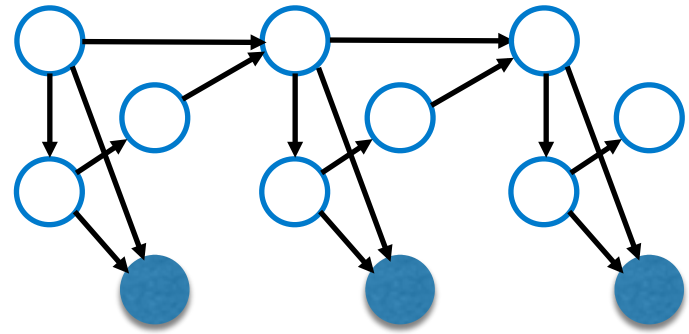
Runtime = **days**

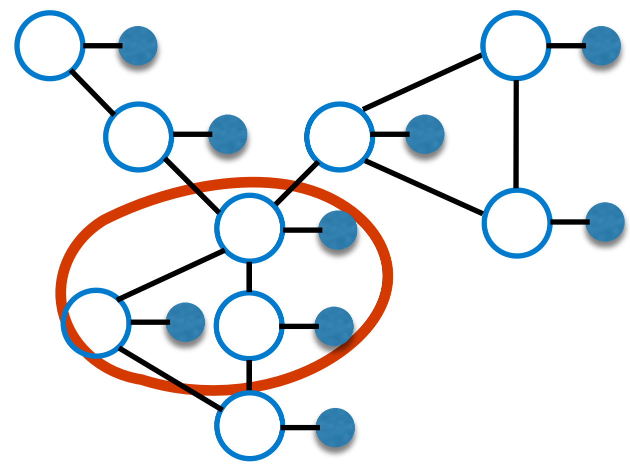
- Adaptive subsampling on **HMM** (*simpler model*)

Runtime = under **1 hr**

BNP and Other Extensions

- Presented finite HMM case, but ideas could generalize to:
  - Nonparametric HMMs
    - Truncation plus split-merge to change the number of states [Bryant & Sudderth, 2012]
  - DBN and MRF models
- Applications to:
  - Large spatial fields
  - Spatio-temporal data, etc.





Overall Summary

- Scalable Bayesian dynamic modeling:
  - *Low-dimensional embeddings* with application to MEG word classification
  - *Clusters* for forming high-resolution housing value index
  - *Graphs of time series* with application to stocks + functional connectivity
- Scalable Bayesian computations in dynamic models
  - Harness *memory decay* to use subset-based methods in HMMs