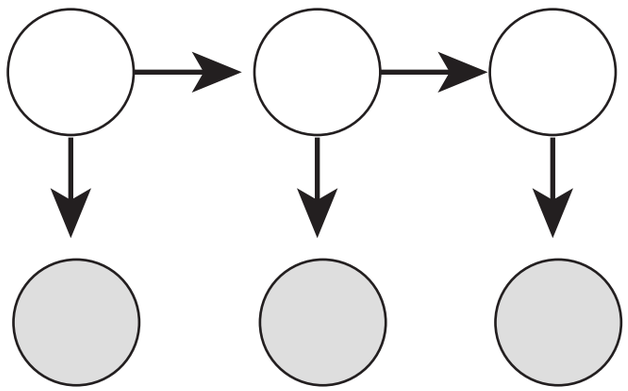
DS-GA 3001.008 Modelling time series data

L6. An unified view of linear models. Beyond linear

Instructor: Cristina Savin

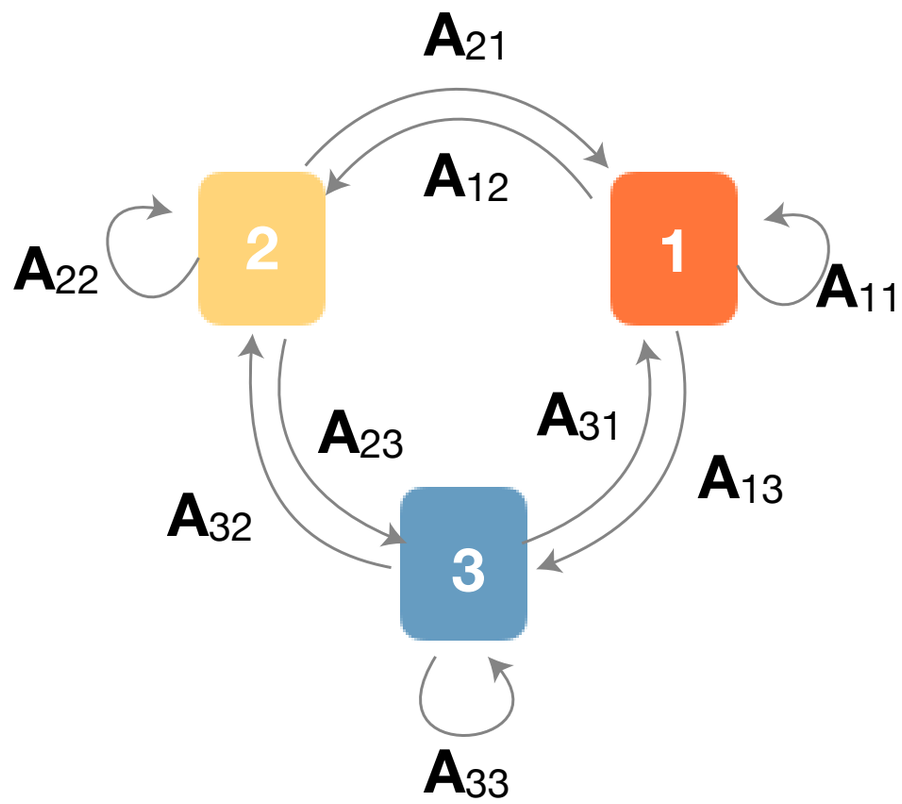
NYU, CNS & CDS

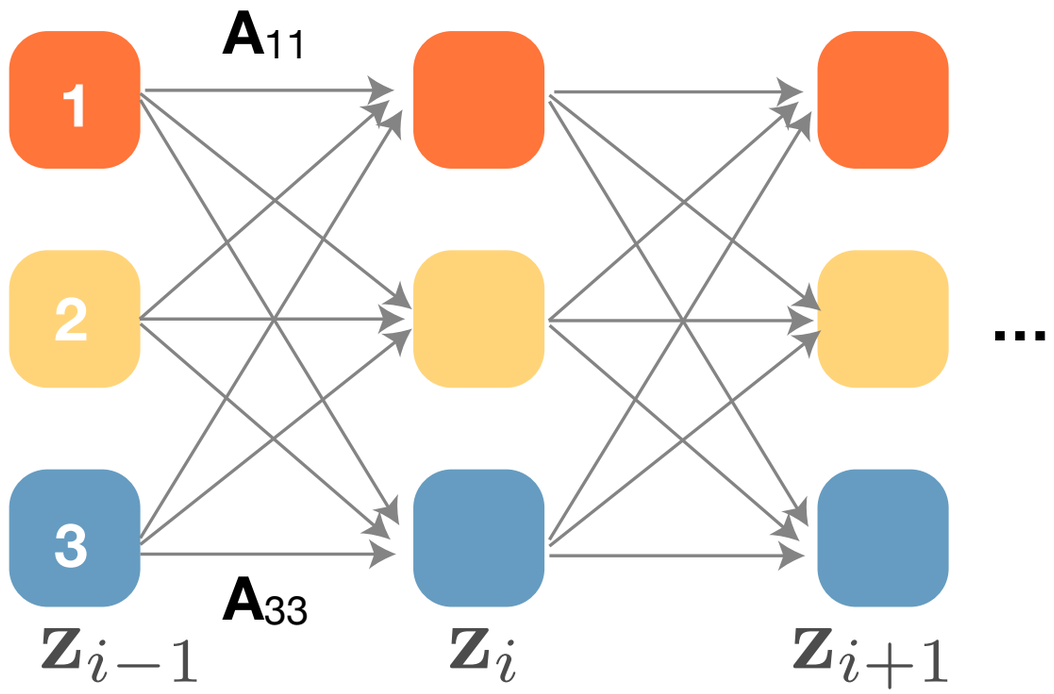
HMMs recap

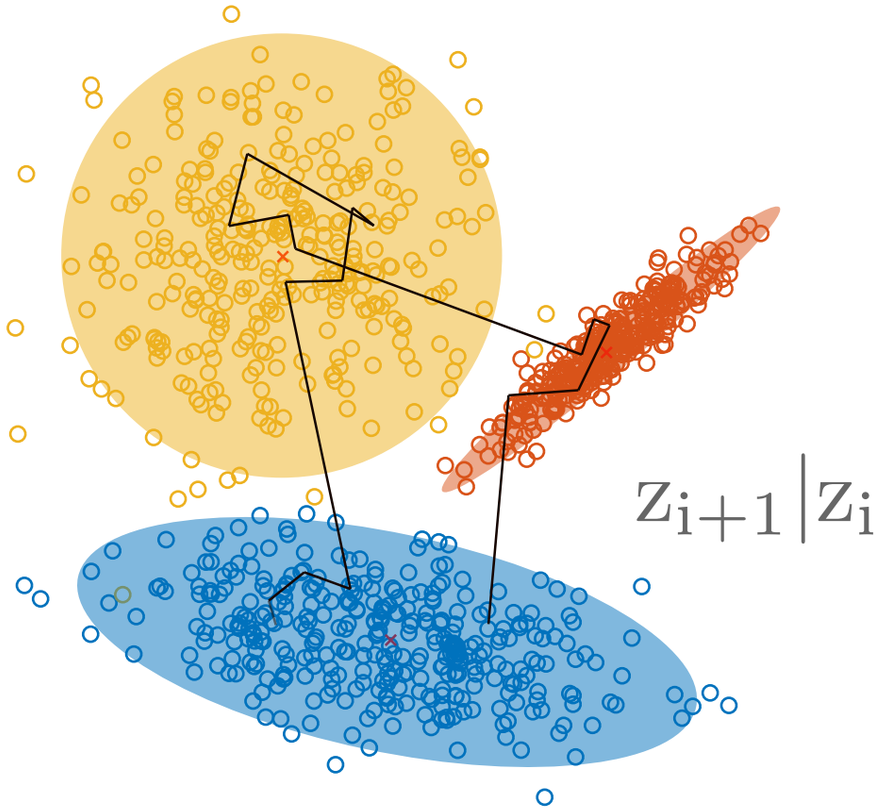


discrete latent states

application-specific observation model







1. Inference

compute exact **marginals**: alpha-beta algorithm
(as we did in LDS with Kalman smoother)

We may also want to determine the most
likely sequence **(MAP)**
Viterbi algorithm

2. Parameter learning

This is just more **EM**

1. Inference: marginals

We represent **posterior marginals** and joints using

$$\begin{aligned} \gamma(\mathbf{z}_i) &= \mathrm{P}\left(\mathbf{z}_i|\mathbf{x}_*, \theta^{\mathrm{old}}\right) \\ \xi(\mathbf{z}_i, \mathbf{z}_{i+1}) &= \mathrm{P}\left(\mathbf{z}_i, \mathbf{z}_{i+1}|\mathbf{x}_*, \theta^{\mathrm{old}}\right) \end{aligned}$$

dimensionality?

With the corresponding **expectations**:

$$\begin{aligned} \gamma(z_{ik}) &= \mathbb{E}[z_{i,k}|\mathbf{x}_*, \theta^{\mathrm{old}}] = \sum_{\mathbf{z}_i} \gamma(\mathbf{z}_i) z_{i,k} \\ \xi(z_{i,j}, z_{i+1,k}) &= \mathbb{E}[z_{i,j} z_{i+1,k}|\mathbf{x}_*, \theta^{\mathrm{old}}] = \sum_{\mathbf{z}_i, \mathbf{z}_{i+1}} \xi(\mathbf{z}_i, \mathbf{z}_{i+1}) z_{i,j} z_{i+1,k} \end{aligned}$$

We seek an efficient way of computing these by using recursions
(dynamic programming, as for LDS)

Recursion equations:

$$\alpha(\mathbf{z}_i) = \mathrm{P}(\mathbf{x}_{1:i}|\mathbf{z}_i)\,\mathrm{P}(\mathbf{z}_i) = \mathrm{P}(\mathbf{x}_i|\mathbf{z}_i)\sum_{\mathbf{z}_{i-1}}\alpha(\mathbf{z}_{i-1})\mathrm{P}(\mathbf{z}_i|\mathbf{z}_{i-1})$$

$$\beta(\mathbf{z}_i) = \mathrm{P}(\mathbf{x}_{i+1:t}|\mathbf{z}_i) = \sum_{\mathbf{z}_{i+1}}\beta(\mathbf{z}_{i+1})\mathrm{P}(\mathbf{x}_{i+1}|\mathbf{z}_{i+1})\,\mathrm{P}(\mathbf{z}_{i+1}|\mathbf{z}_i)$$

With initial conditions:

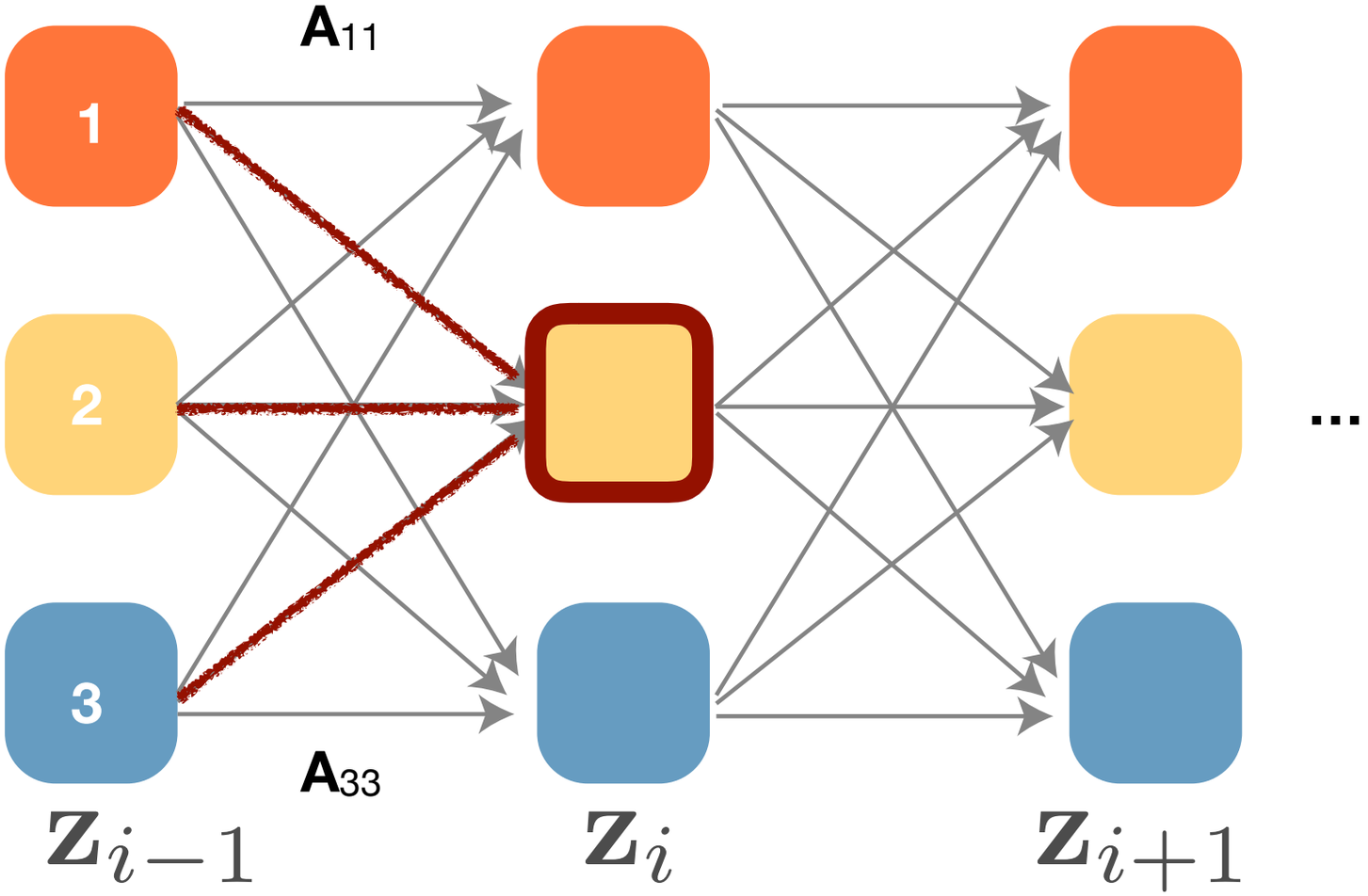
$$\alpha(\mathbf{z}_1) = \mathrm{P}(\mathbf{x}_1|\mathbf{z}_1)\,\mathrm{P}(\mathbf{z}_1) = \prod_k(\pi_k\mathrm{P}(\mathbf{x}_1|\phi_k))^{z_{1k}}$$

$$\beta(\mathbf{z}_t) = 1$$

These quantities are enough to also compute the joint:

$$\mathrm{P}(\mathbf{z}_i, \mathbf{z}_{i+1}|\mathbf{x}_{1:t}) = \frac{\alpha(\mathbf{z}_i)\mathrm{P}(\mathbf{z}_{i+1}|\mathbf{z}_i)\,\mathrm{P}(\mathbf{x}_{i+1}|\mathbf{z}_{i+1})\,\beta(\mathbf{z}_{i+1})}{\mathrm{P}(\mathbf{x}_{1:t})}$$

Interpretation: sum-product



$$\alpha(\mathbf{z}_i) \quad = \quad \mathrm{P}\left(\mathbf{x}_i|\mathbf{z}_i\right) \sum_{\mathbf{z}_{i-1}} \alpha(\mathbf{z}_{i-1}) \mathrm{P}\left(\mathbf{z}_i|\mathbf{z}_{i-1}\right)$$

consider **all possible ways** to reach state $\mathbf{z}_{i,k}$ and **sum** them up,
then combine with local evidence

Implementational caveat (a rather important one)

$$\alpha(\mathbf{z}_i) \quad = \quad \mathrm{P}\left(\mathbf{x}_{1:i}, \mathbf{z}_i\right)$$

joint distribution over increasingly many things,
the probability of any particular configuration is vanishingly small,
easy to go under machine precision

Solution: rescale messages to keep things in sensible range

$$\hat{\alpha}(\mathbf{z}_i) \quad = \quad \mathrm{P}\left(\mathbf{z}_i | \mathbf{x}_{1:i}\right) = \frac{\alpha(\mathbf{z}_i)}{\mathrm{P}\left(\mathbf{x}_{1:i}\right)}$$

$$\hat{\beta}(\mathbf{z}_i) \quad = \quad \frac{\mathrm{P}\left(\mathbf{x}_{i+1:t} | \mathbf{z}_i\right)}{\mathrm{P}\left(\mathbf{x}_{\mathbf{i+1:t}} | \mathbf{x}_{1:i}\right)} = \frac{\beta(\mathbf{z}_i)}{\prod_{j=i+1:t} c_j}$$

where we've introduced an intermediate variable

$$c_i \quad = \quad \mathrm{P}\left(\mathbf{x}_i | \mathbf{x}_{1:i-1}\right)$$

$$\mathrm{P}\left(\mathbf{x}_{1:i}\right) \quad = \quad \prod_{j=1:i} c_j$$

The updated recursion equation for $\hat{\alpha}$, and $\hat{\beta}$ become:

$$\begin{aligned} c_i\hat{\alpha}(\mathbf{z}_i) &= \mathrm{P}\left(\mathbf{x}_i|\mathbf{z}_i\right)\sum_{\mathbf{z}_{i-1}}\hat{\alpha}(\mathbf{z}_{i-1})\mathrm{P}\left(\mathbf{z}_i|\mathbf{z}_{i-1}\right) \\ c_{i+1}\hat{\beta}(\mathbf{z}_i) &= \sum_{\mathbf{z}_{i+1}}\hat{\beta}(\mathbf{z}_{i+1})\mathrm{P}\left(\mathbf{x}_{i+1}|\mathbf{z}_{i+1}\right)\mathrm{P}\left(\mathbf{z}_{i+1}|\mathbf{z}_i\right) \end{aligned}$$

Finally the new expressions for the posterior marginals are:

$$\begin{aligned} \gamma(\mathbf{z}_i) &= \hat{\alpha}(\mathbf{z}_i)\hat{\beta}(\mathbf{z}_i) \\ \xi(z_{i,j}, z_{i+1,k}) &= {c_{i+1}}^{-1}\hat{\alpha}(\mathbf{z}_i)\mathrm{P}\left(\mathbf{z}_{i+1}|\mathbf{z}_i\right)\mathrm{P}\left(\mathbf{x}_{i+1}|\mathbf{z}_{i+1}\right)\hat{\beta}(\mathbf{z}_{i+1}) \end{aligned}$$

Learning: EM

Log likelihood: $\mathcal{L}(\theta) = \log \mathrm{P}(\mathbf{x}_*|\theta)$

$$\mathrm{P}(\mathbf{x}_*, \mathbf{z}_*|\theta) = \mathrm{P}_\theta(\mathbf{z}_0) \prod_i \mathrm{P}_\theta(\mathbf{z}_{i+1}|\mathbf{z}_i) \prod_i \mathrm{P}_\theta(\mathbf{x}_i|\mathbf{z}_i)$$

Pretend z known, then take expectations under q*

Separable, total loss is sum of separate components

Look for a peak of the corresponding loss term

Learning: EM

M-step:

In general:

$$\pi_k^{\text{new}} = \frac{\gamma(z_{1,k})}{\sum_j \gamma(z_{1,j})}$$

$$A_{jk}^{\text{new}} = \frac{\sum_i \xi(z_{i,j}, z_{i+1,k})}{\sum_{i,l} \xi(z_{i,j}, z_{i+1,l})}$$

For a simple *gaussian* observation model

$$\mu_k^{\text{new}} = \frac{1}{\sum_i \gamma(z_{i,k})} \sum_i \gamma(z_{i,k}) \mathbf{x}_i$$

$$\Sigma_k^{\text{new}} = \frac{1}{\sum_i \gamma(z_{i,k})} \sum_i \gamma(z_{i,k}) \mathbf{x}_i \mathbf{x}_i^{\text{t}} - \mu_k \mu_k^{\text{t}}$$

This is just a weighted version of empirical mean and covariance

Plan

Unified view of basic model, links to iid counterparts

Reading: Rowise & Gahrahmani 1999

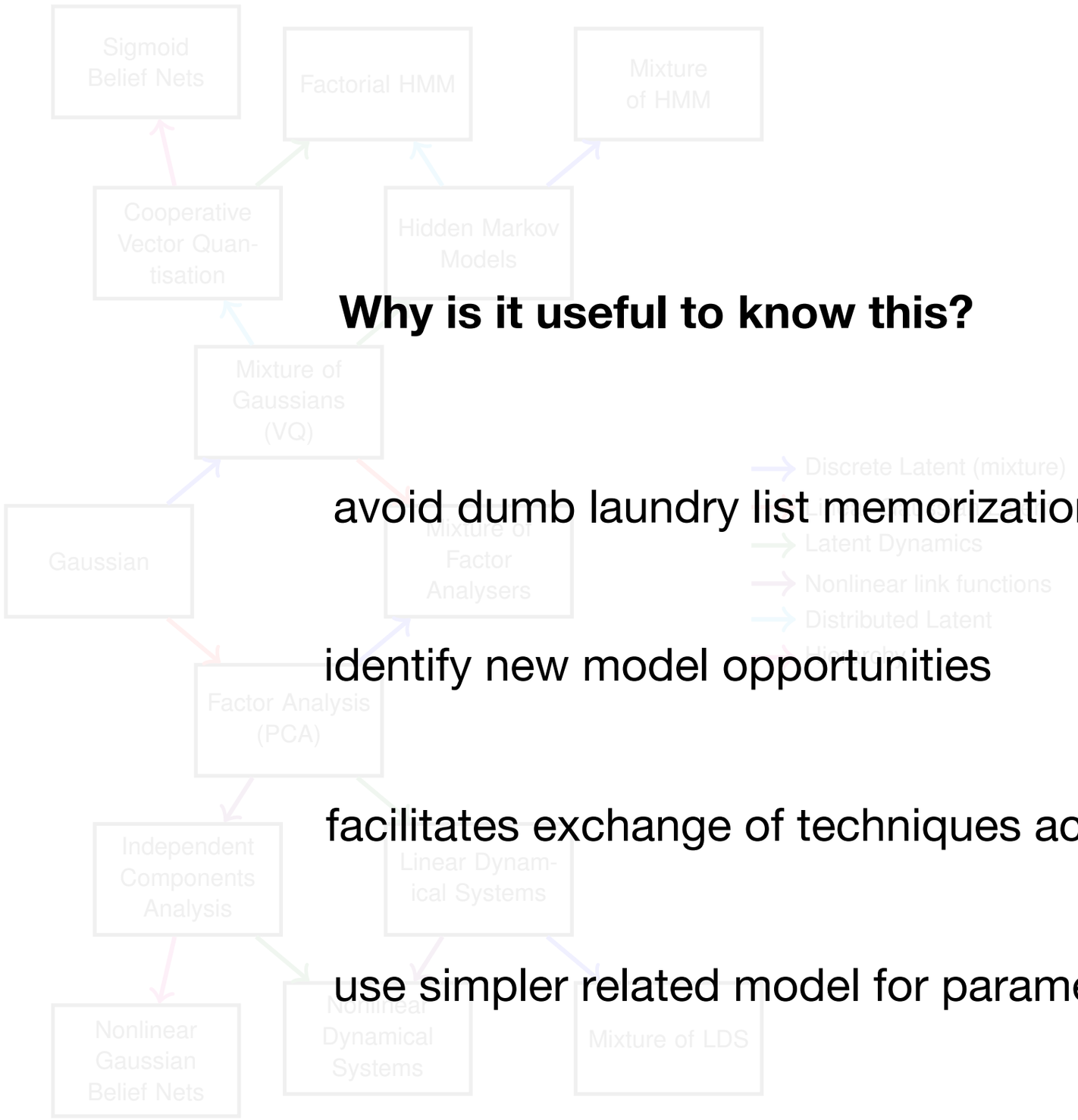
Examples generalizations/ extensions

Reading: Bishop Chp.13

Organizing the more complicated models

The friendship network for generative models

aka The generative model of generative models



Why is it useful to know this?

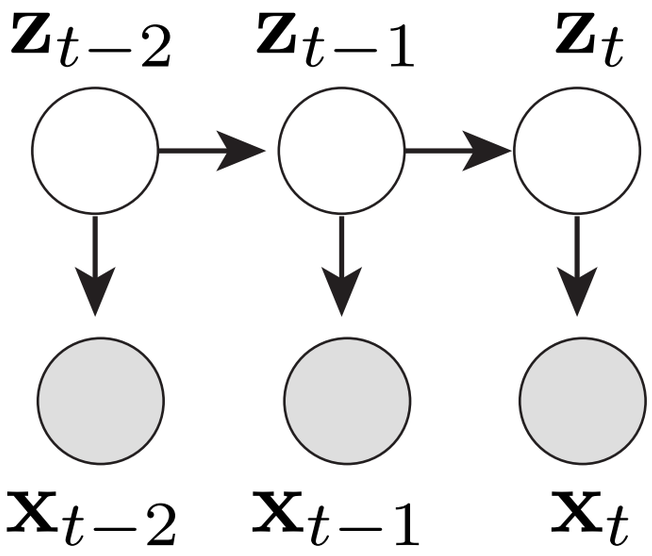
avoid dumb laundry list memorization

identify new model opportunities

facilitates exchange of techniques across models

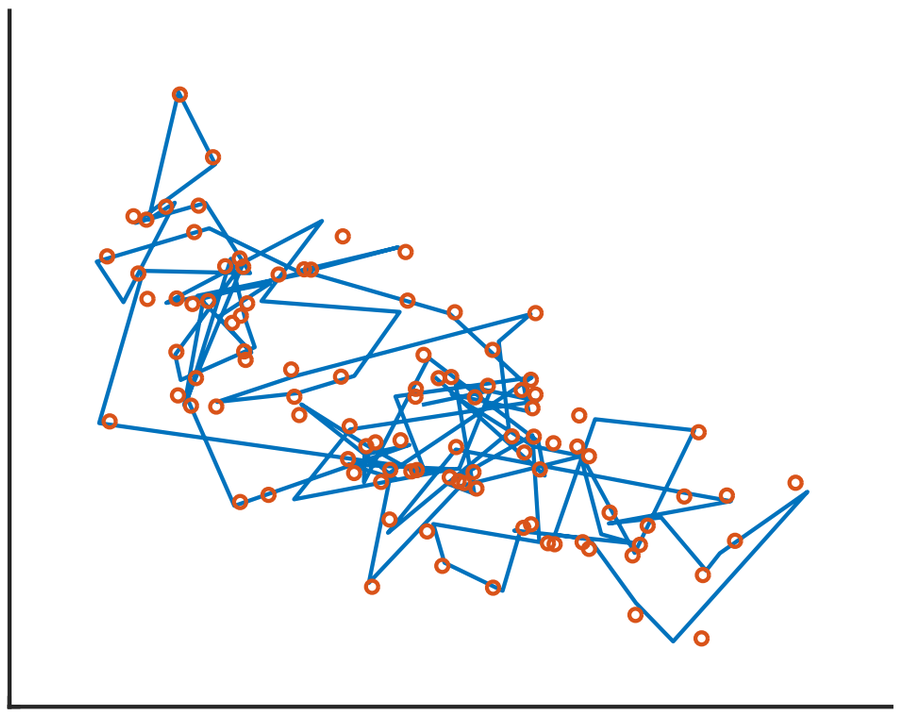
use simpler related model for parameter initialization

The root is classic LDS



$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{w}_t$$
$$\mathbf{x}_t = \mathbf{C}\mathbf{z}_t + \mathbf{v}_t$$

$$\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q})$$
$$\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R})$$
$$\mathbf{z}_0 \sim \mathcal{N}(\mu_0, \Sigma)$$



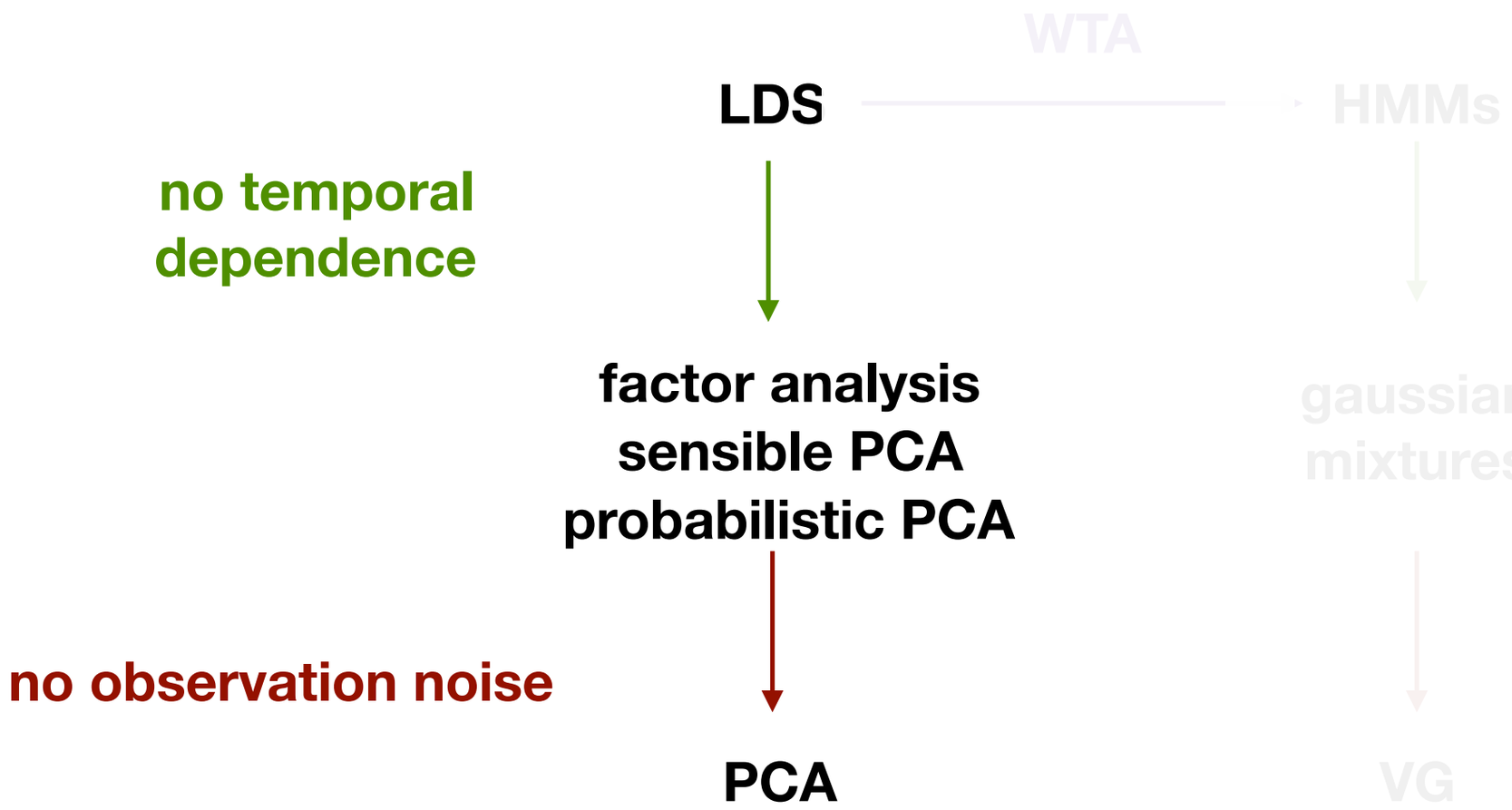
1. Inference

compute exact **marginals**:
Kalman smoother

2. Parameter learning

Use **EM**

Putting together the basic models (Roweis & Ghahramani, ’99)



Recovering iid case: A=0

$$\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$

$$\mathbf{x}_t = \mathbf{C}\mathbf{z}_t + \mathbf{v}_t$$

$$\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R})$$

marginalize out latent:

$$\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{C}\mathbf{Q}\mathbf{C}^{\mathrm{t}} + \mathbf{R})$$

Intuition: pancakes!

Underspecified, just set Q=I

Trivial fit: R = cov data, C=0

How to make this less boring?

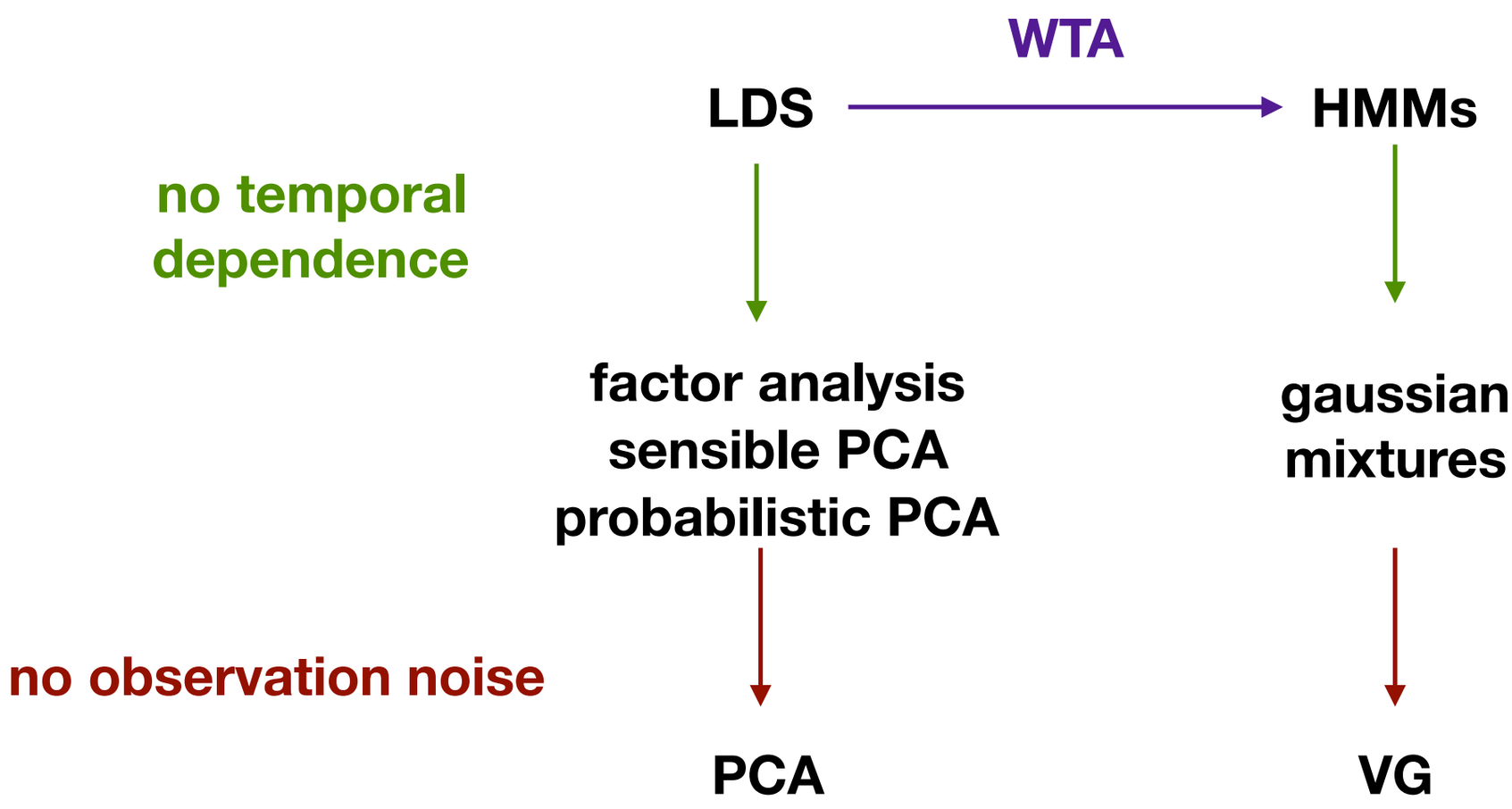
Idea: restrict R so that C can capture some interesting structure

Option 1: R is diagonal -> **factor analysis** (FA)

Option 2: R is spherical -> sensible **principle component analysis** (sPCA)

FA and PCA variants differ in the constraints they impose on the observation noise

Putting together the basic models (Roweis & Ghahramani, ’99)



Potentially useful links to autoencoders, see paper if interested

HMM and nonlinear LDS

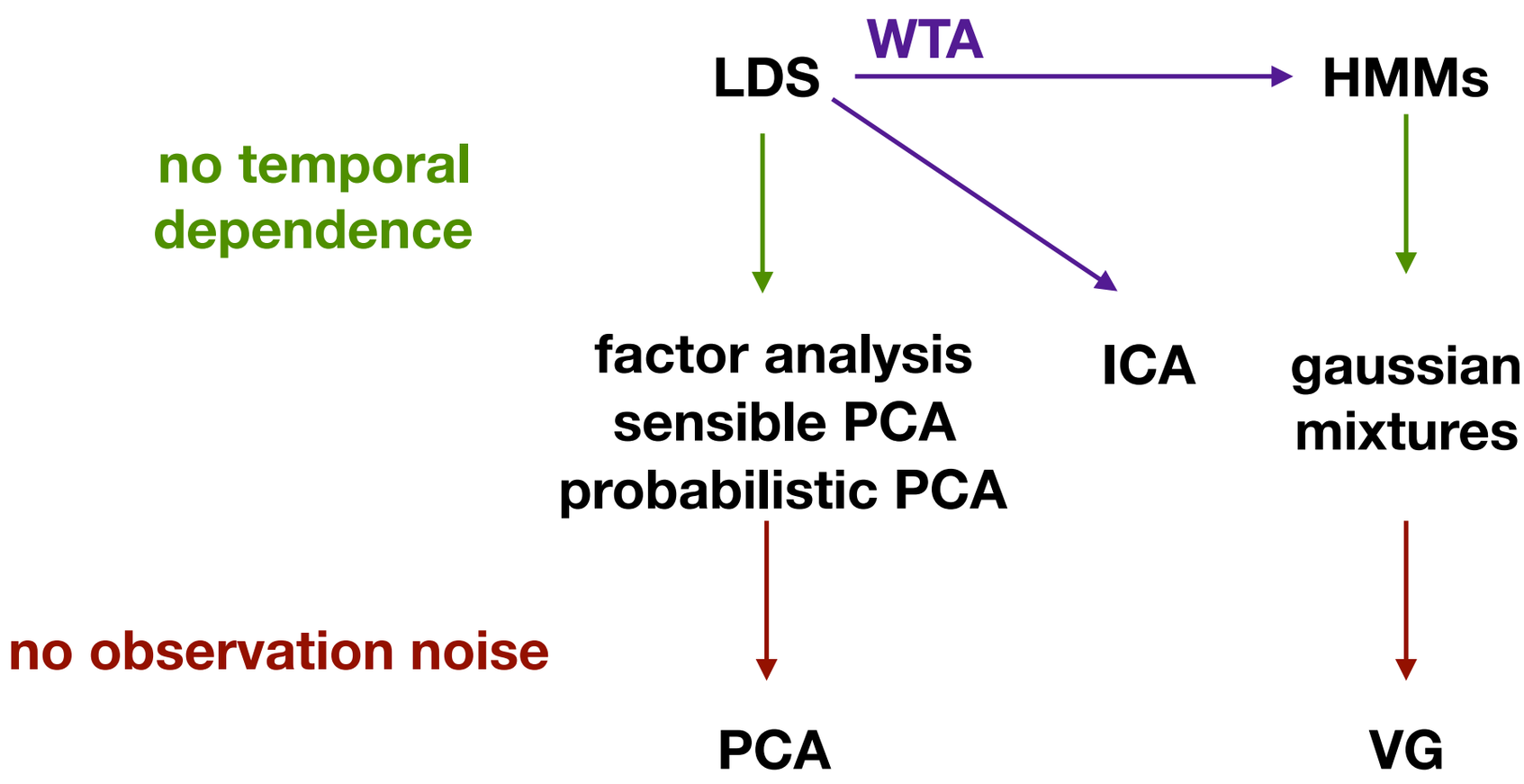
making a discrete version of LSD

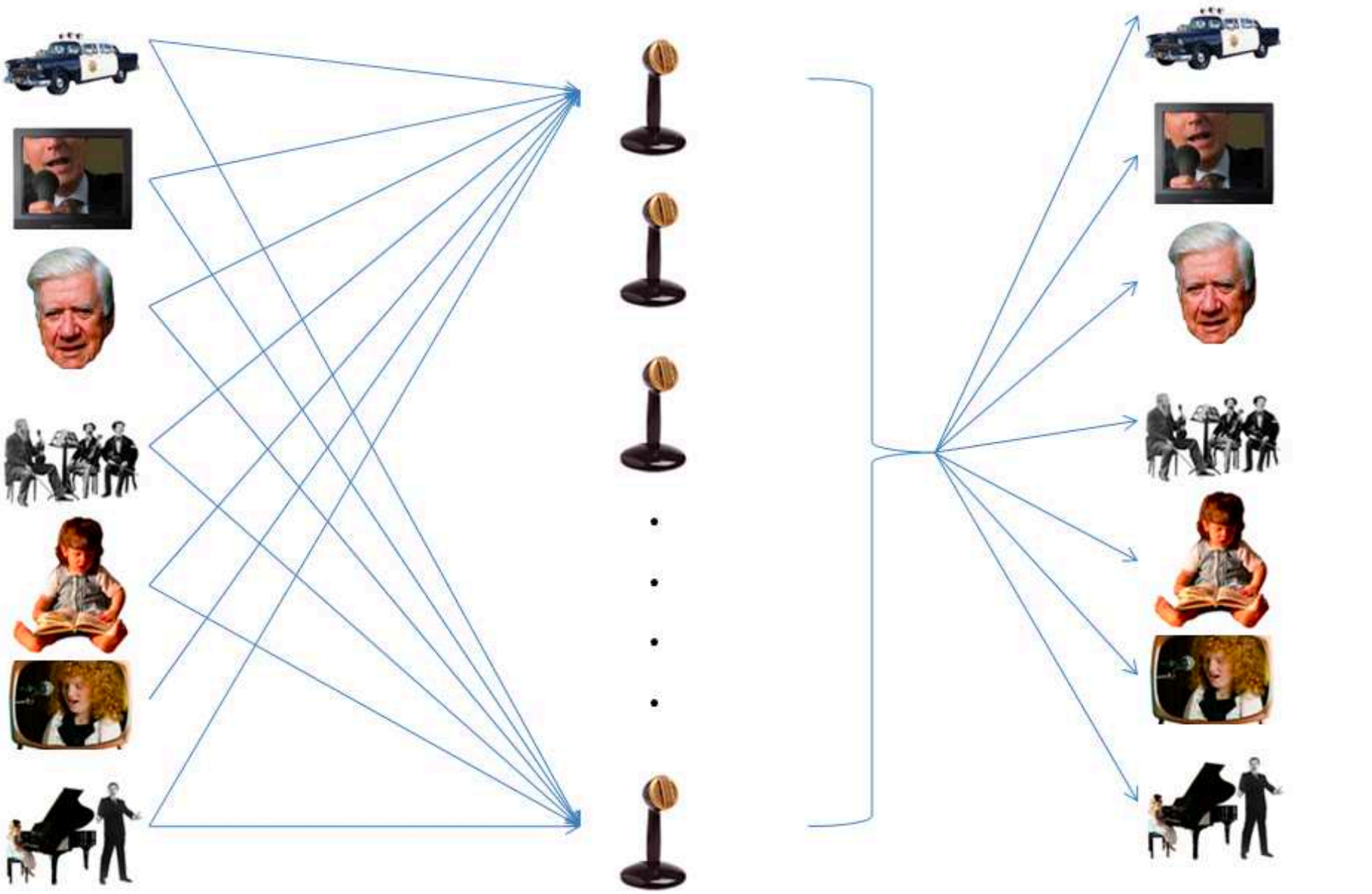
$$\mathbf{x}_{t+1} = \mathbf{WTA}[\mathbf{A}\mathbf{x}_t + \mathbf{w}_t] = \mathbf{WTA}[\mathbf{A}\mathbf{x}_t + \mathbf{w}_\bullet]$$
$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{v}_t = \mathbf{C}\mathbf{x}_t + \mathbf{v}_\bullet$$

$$\mathbf{y} = \mathrm{WTA}(\mathbf{x}): \quad y_i = 1, i = \mathrm{argmax_j}[x_j],$$
$$y_i = 0, \text{ otherwise}$$

Not exactly trivial, but there is a 1-to-1 map
between this definition and an HMM with Gaussian observation noise

Putting together the basic models (Roweis & Ghahramani, '99)





$$\mathbf{x}_\bullet = g(\mathbf{w}_\bullet)$$

$$\mathbf{y}_\bullet = \mathbf{C}\mathbf{x}_\bullet + \mathbf{v}_\bullet$$

$$\mathbf{w}_\bullet \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$

$$\mathbf{v}_\bullet \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$$

Plan

Unified view of basic model, links to iid counterparts

Reading: Rowise & Gahrahmani 1999*

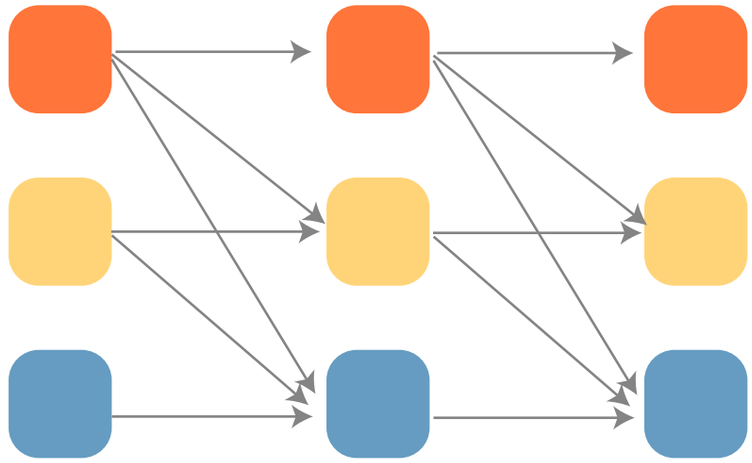
Examples generalizations/ extensions

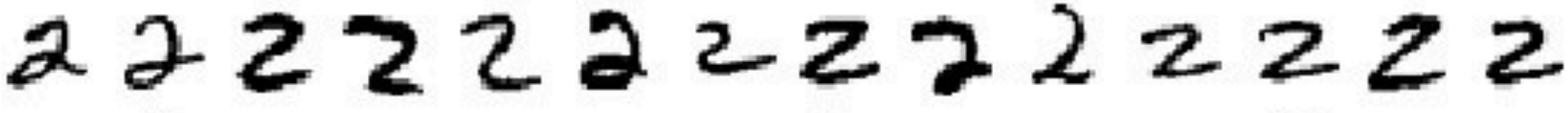
Reading: Bishop Chp.13

**Postponing link to auto-encoders to neural nets lecture*

Special cases: HMM variants

natural ordering of components: **left-to-right** HMMs

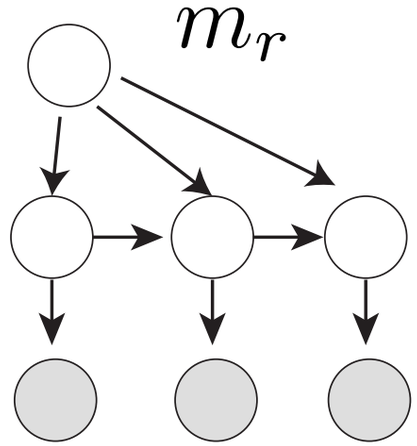




does not capture different ‘styles’: a **mixture** of HMMs

collection of HMMs, with separate parameters

consider multiple HMM parameter settings



Special cases: HMM variants

HMMs constrain the amount of time spent in the same state to be:

$$\mathrm{P}(t) = A_{kk}^{t}(1 - A_{kk})$$ (exponential in t)

Solution:

$$A_{kk} = 0$$

separately define a ‘time you stay in state’ model $\mathrm{P}(t|k)$

needs relatively straightforward modifications for inference and learning

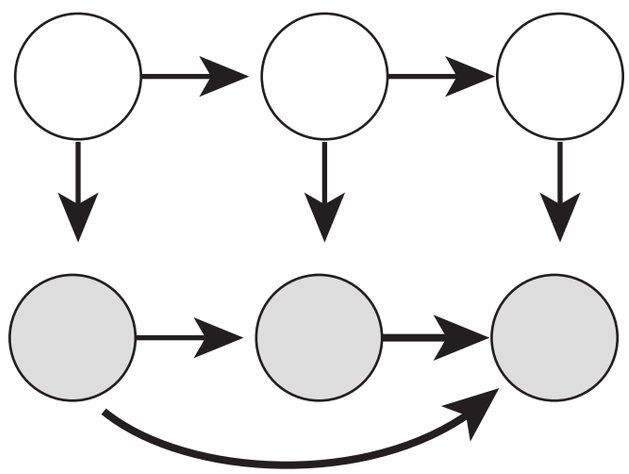
e.g. proteins, patterns of looking

Special cases: HMM variants

HMMs can be weak at capturing long range dependencies

autoregressive HMM

mixes ARMA-like ideas with latent space models



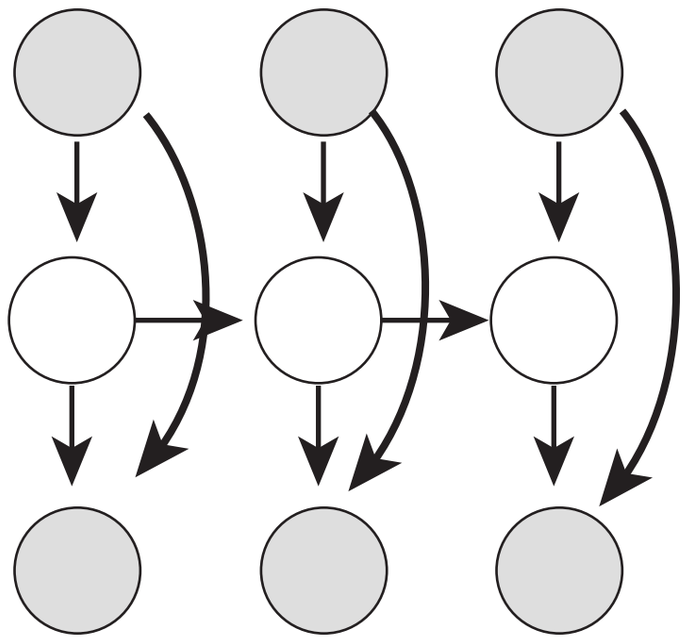
inference and learning are still (sort of) tractable

e.g. speech recognition/ synthesis

Special cases: HMM variants

A slightly different scenario:
latent state constrained by some other observed variables $\mathbf{u}_i$

input-output HMMs

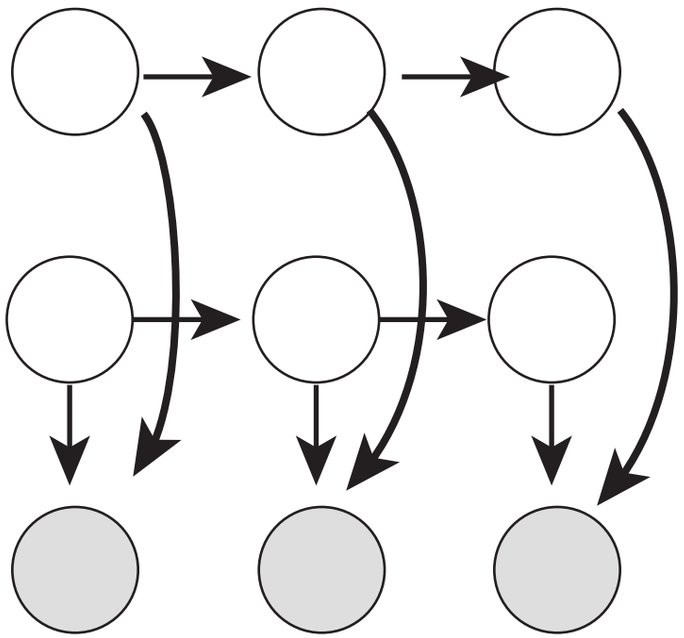


e.g. a plant

we need to model joint statistics of **u**,**x**, but it's too complicated to directly consider their temporal dependencies, we still want to use the trick of finding a latent representation that condenses the relevant history down to 1st order Markov dependencies

Special cases: HMM variants

factorial HMMs



multiple independent
latent chains (usually binary)
jointly determining the observations

why would this be a good idea?

‘digitization’

disadvantage: harder to do learning

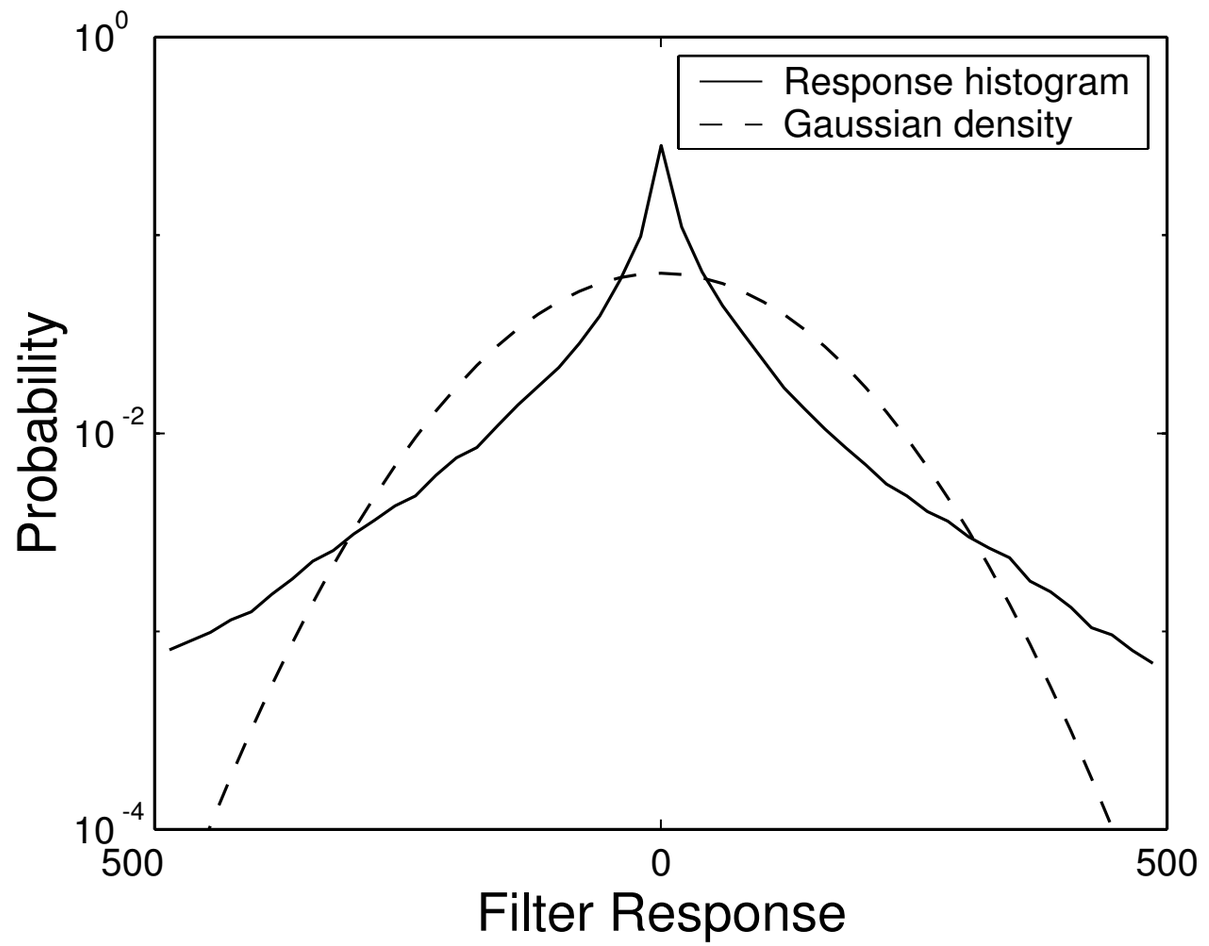
some neat variational-based solutions

Special cases: LDS extensions

Fundamental limitation: joint of observed is multivariate gaussian

How do we see this?

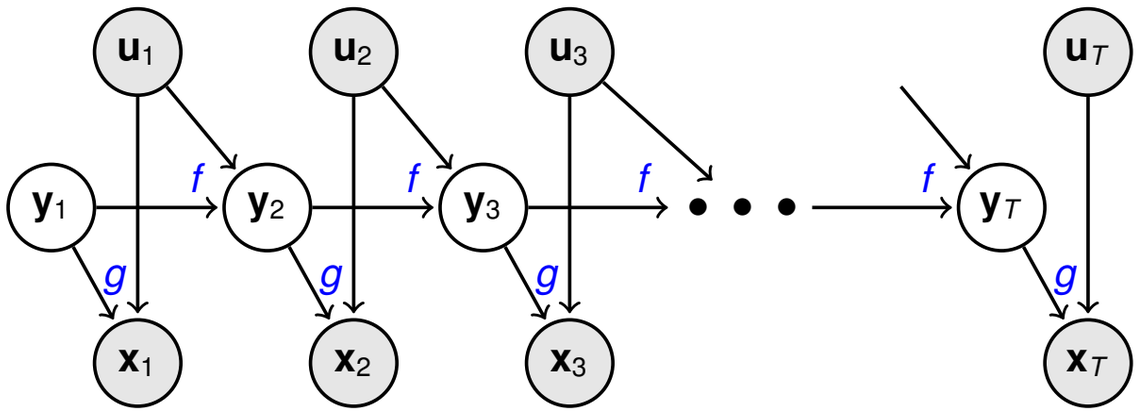
Much of the world is neither linear nor Gaussian



Special cases: LDS extensions

Extended Kalman filter

nonlinear ->intractable, but we can *locally* linearize around current estimate
use normal Kalman filtering

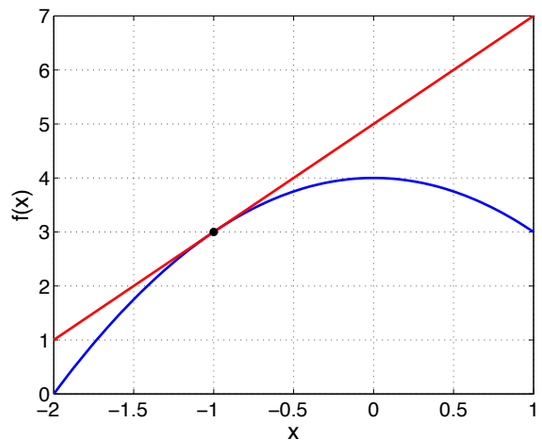


$$\mathbf{y}_{t+1} = f(\mathbf{y}_t, \mathbf{u}_t) + \mathbf{w}_t$$
$$\mathbf{x}_t = g(\mathbf{y}_t, \mathbf{u}_t) + \mathbf{v}_t$$

$\mathbf{w}_t, \mathbf{v}_t$ usually assumed Gaussian.

$$\mathbf{y}_{t+1} \approx f(\hat{\mathbf{y}}_t^t, \mathbf{u}_t) + \left.\frac{\partial f}{\partial \mathbf{y}_t}\right|_{\hat{\mathbf{y}}_t^t} (\mathbf{y}_t - \hat{\mathbf{y}}_t^t) + \mathbf{w}_t$$

$$\mathbf{x}_t \approx g(\hat{\mathbf{y}}_t^{t-1}, \mathbf{u}_t) + \left.\frac{\partial g}{\partial \mathbf{y}_t}\right|_{\hat{\mathbf{y}}_t^{t-1}} (\mathbf{y}_t - \hat{\mathbf{y}}_t^{t-1}) + \mathbf{v}_t$$



no guarantees

if not:
particle filtering
EP, etc

Special cases: hybrid models

When do these make sense: some aspects of the data are discrete, others not

switching state space models

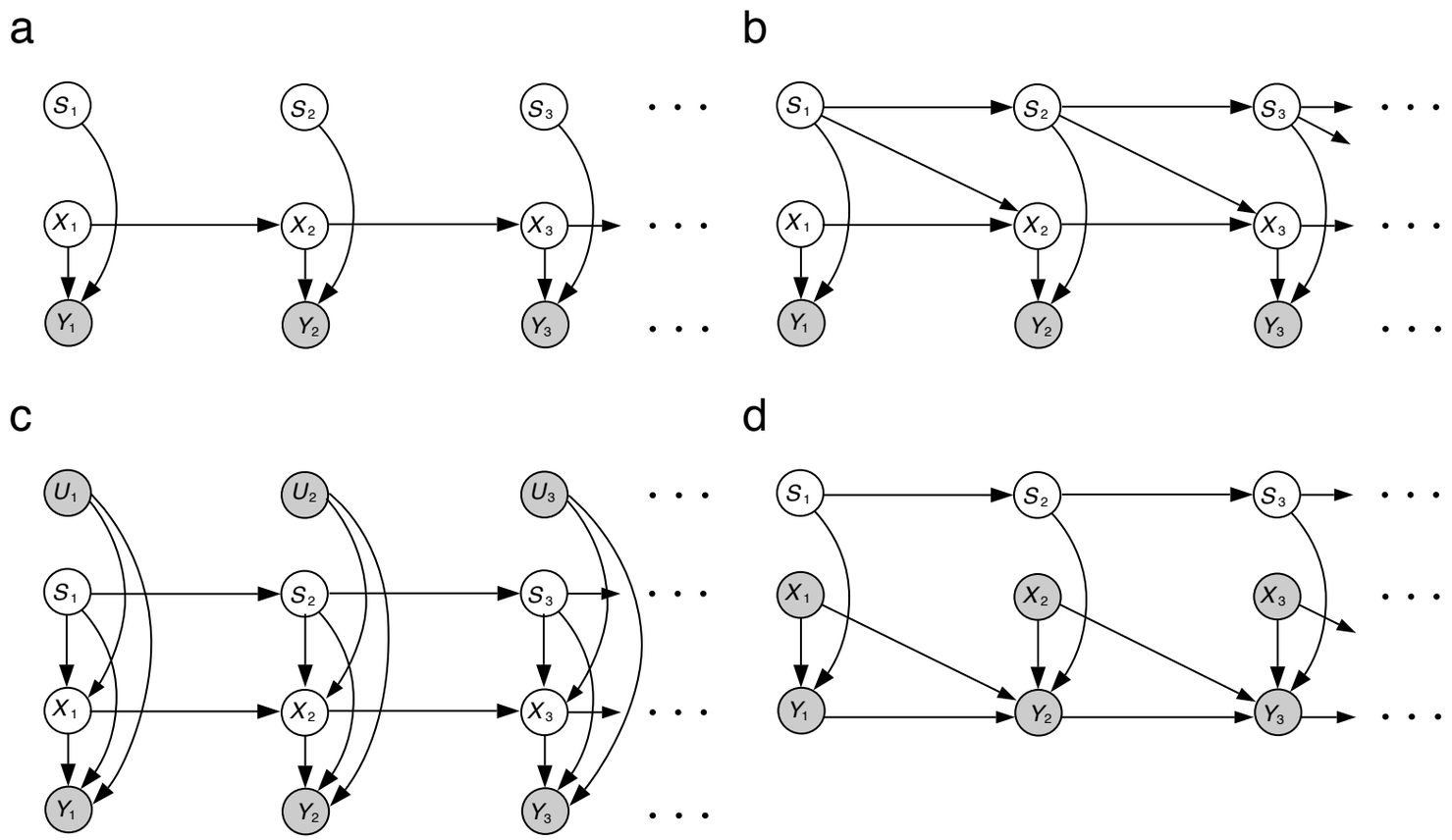
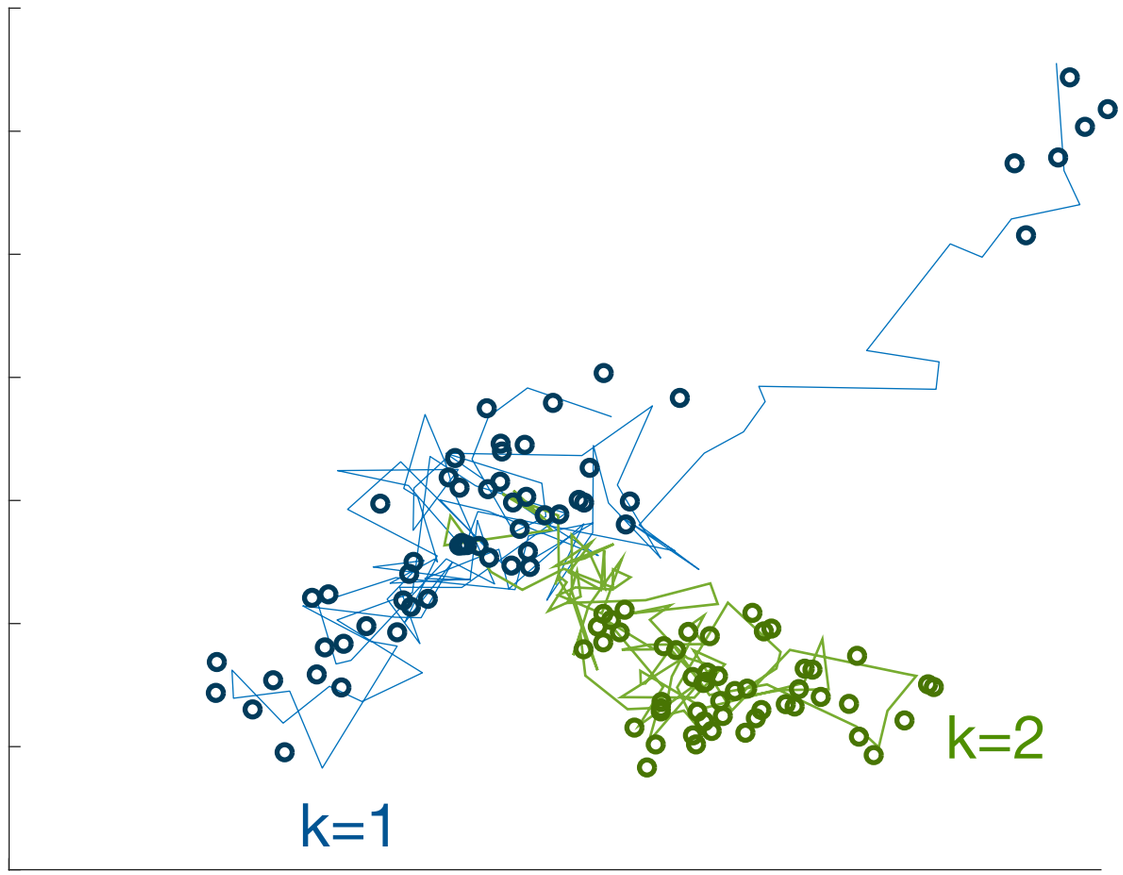


Figure 2: Directed acyclic graphs specifying conditional independence relations for various switching state-space models. (a) Shumway and Stoffer (1991): the output matrix (C in equation (3)) switches independently between a fixed number of choices at each time step. Its setting is represented by the discrete hidden variable S_t; (b) Bar-Shalom and Li (1993): both the output equation and the dynamic equation can switch and the switches are Markov; (c) Kim (1994); (d) Fraser and Dimitriadis (1993): outputs and states are observed.

is the latents are all discrete **switching HMM model**

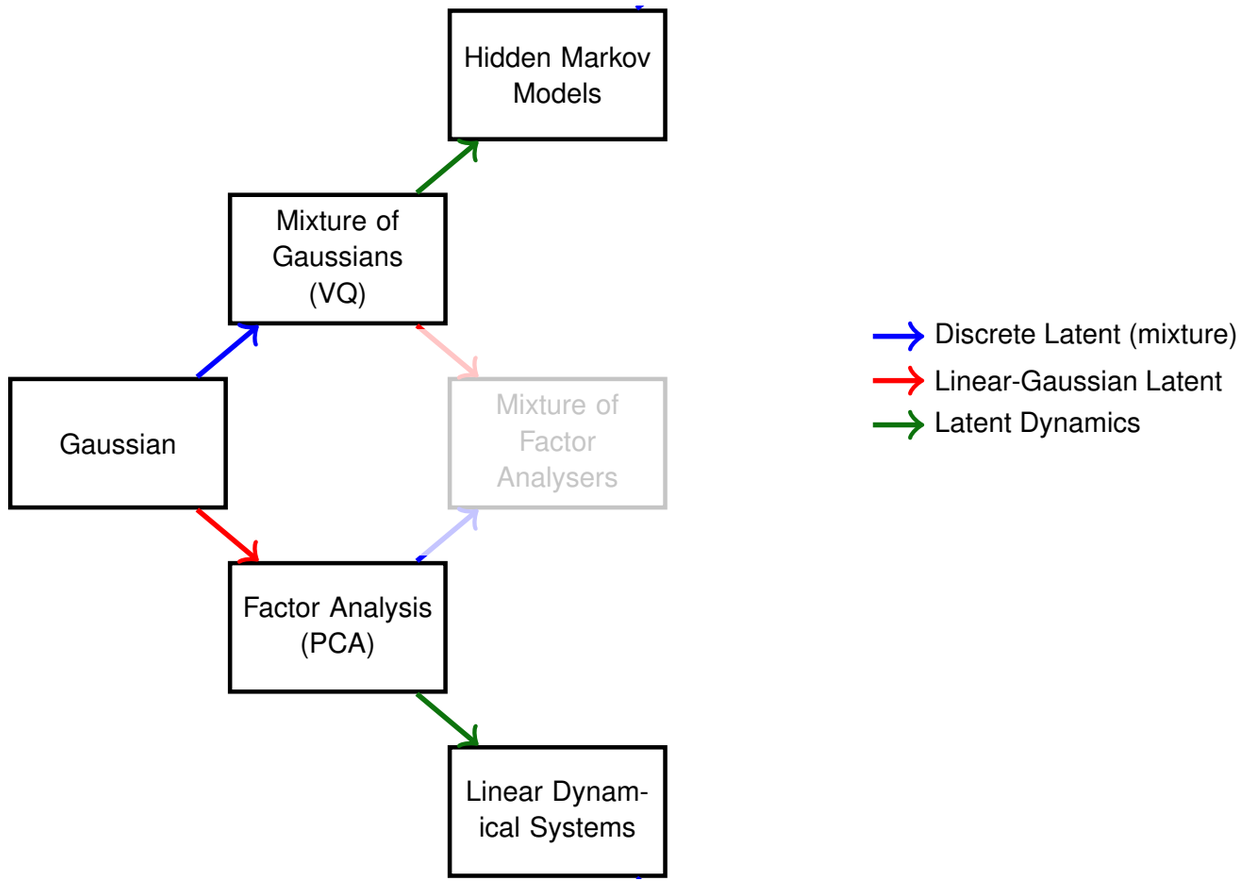
Example: **Gahrahmani & Hinton 1998**

multiple parallel linear gaussian chains +
a HMM deciding which is responsible for generating the current output



impossible to do inference exactly in this model,
but some nice variational approximations,
independent forward backward recursions along each chain

A Generative Model for Generative Models



Adapted from Roweis & Ghahramani (1999). A Unifying Review of Linear Gaussian Models. *Neural Comput.* **11**(2).

General lessons

Basic models are nice but fairly restricted

Some general ideas about how to go forward:

hierarchical models, e.g. mixtures

distributed models, e.g. factorial HMMs, DBNs

General lessons

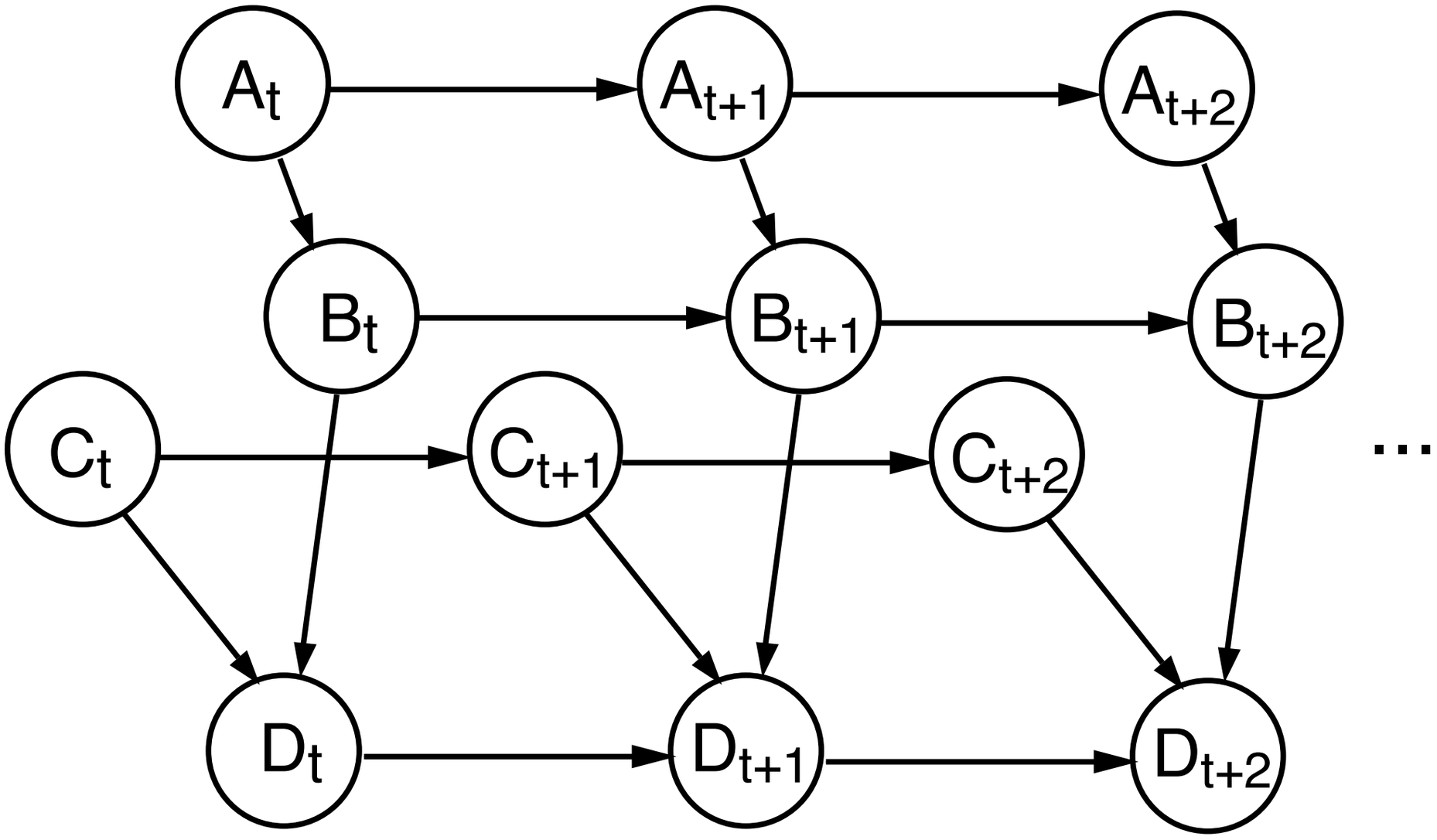
Basic models are nice but fairly restricted

Some general ideas about how to go forward:

hierarchical models, e.g. mixtures

distributed models, e.g. factorial HMMs, DBNs

Dynamic Bayesian Nets



distributed HMMs, with structured dependencies between latents

General lessons

Basic models are nice but fairly restricted

Some general ideas about how to go forward:

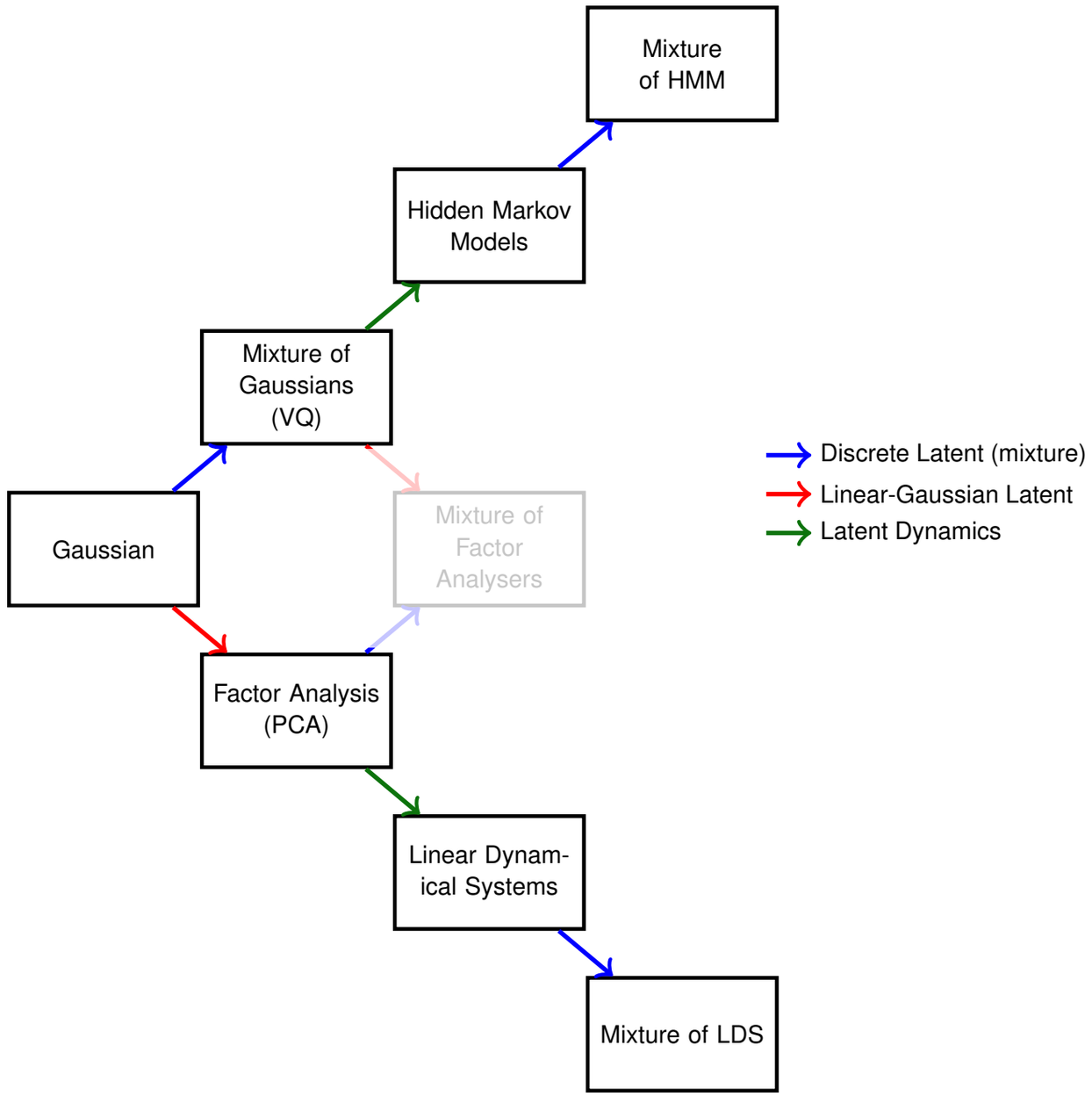
hierarchical models, e.g. mixtures

distributed models, e.g. factorial HMMs

nonlinear models, e.g. extended Kalman filtering

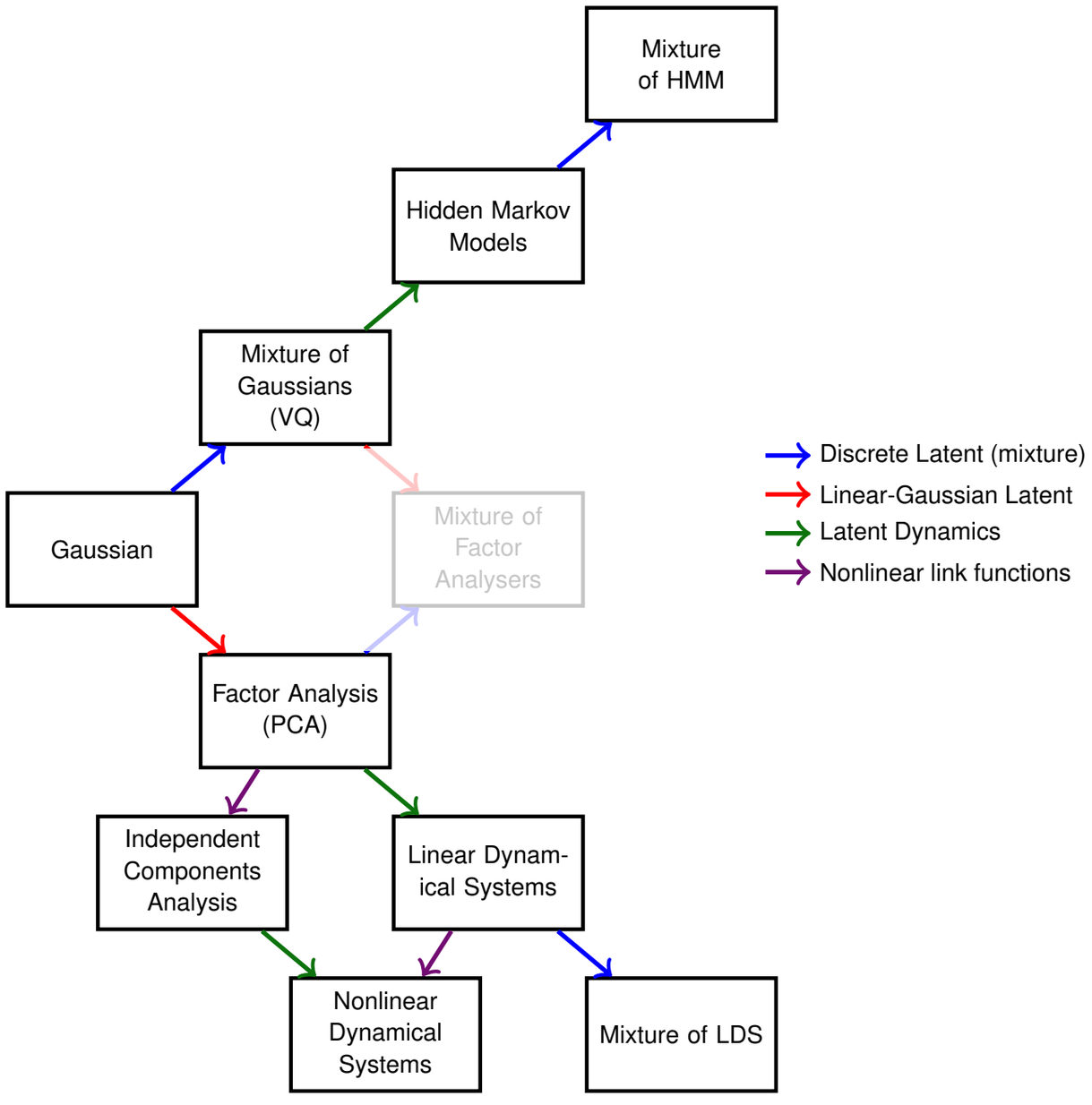
non-gaussian models, HMMs with non-gaussian observations

A Generative Model for Generative Models



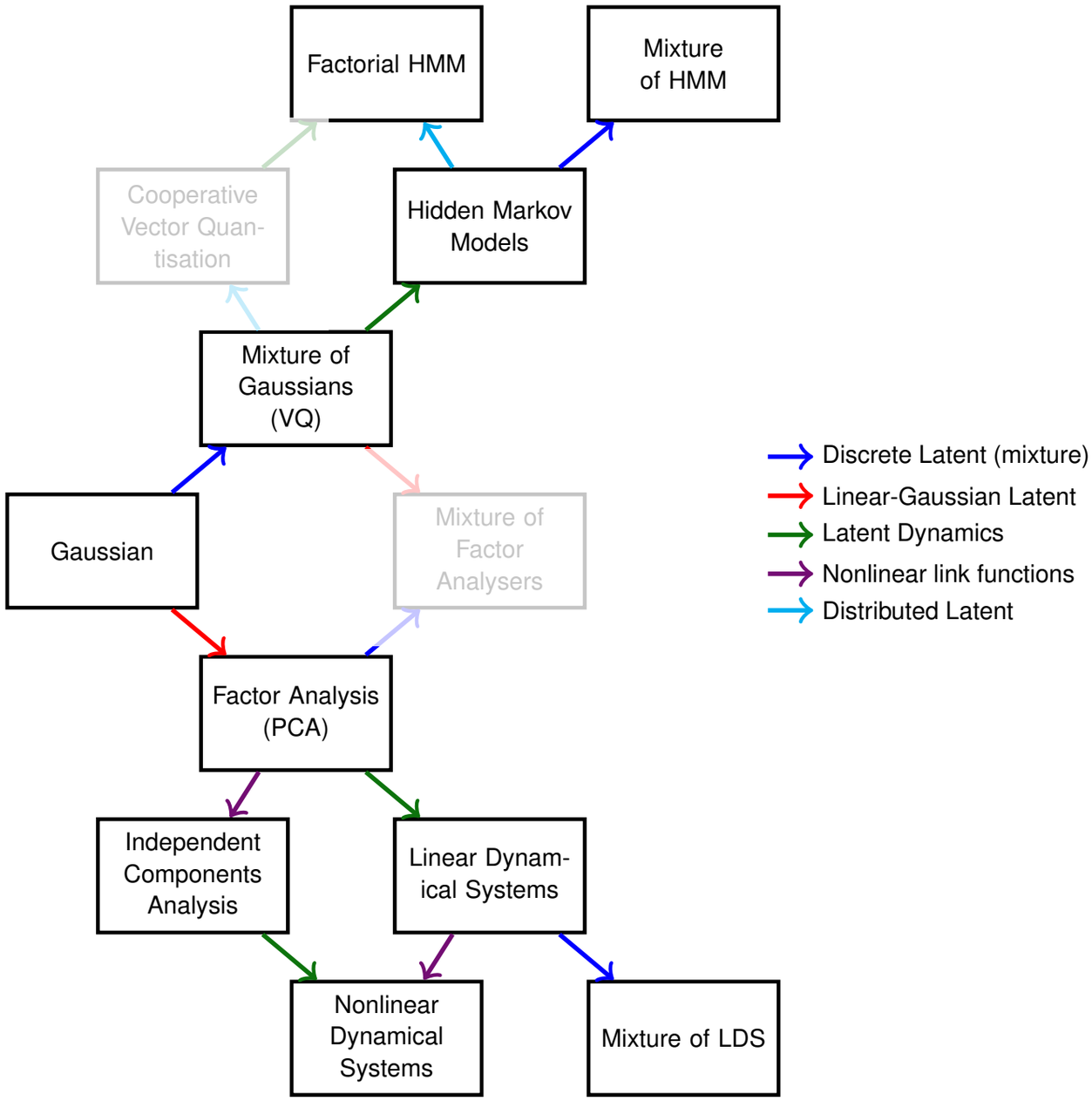
Adapted from Roweis & Ghahramani (1999). A Unifying Review of Linear Gaussian Models. *Neural Comput.* **11**(2).

A Generative Model for Generative Models



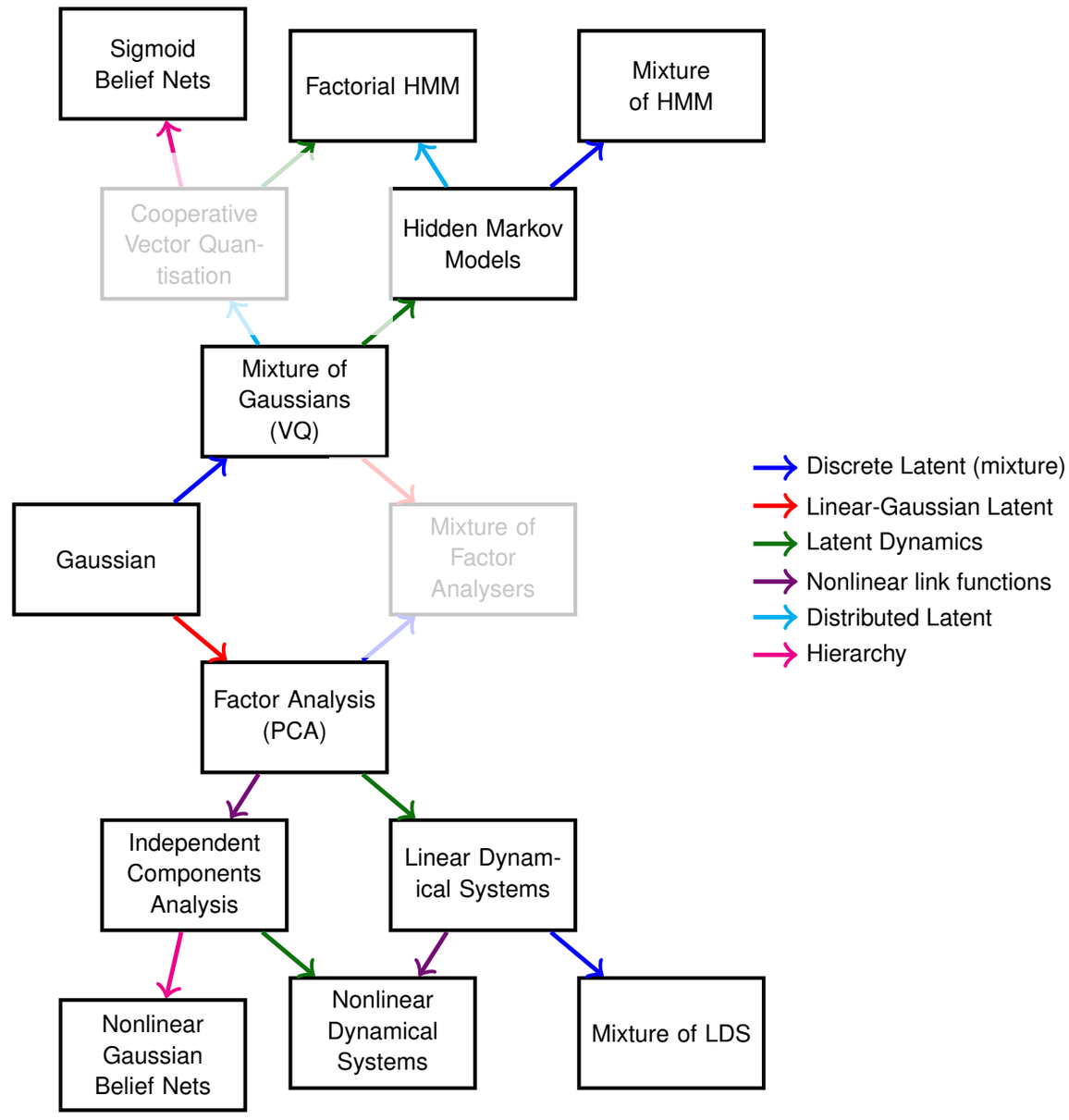
Adapted from Roweis & Ghahramani (1999). A Unifying Review of Linear Gaussian Models. *Neural Comput.* **11**(2).

A Generative Model for Generative Models



Adapted from Roweis & Ghahramani (1999). A Unifying Review of Linear Gaussian Models. *Neural Comput.* **11**(2).

A Generative Model for Generative Models



Adapted from Roweis & Ghahramani (1999). A Unifying Review of Linear Gaussian Models. *Neural Comput.* **11**(2).