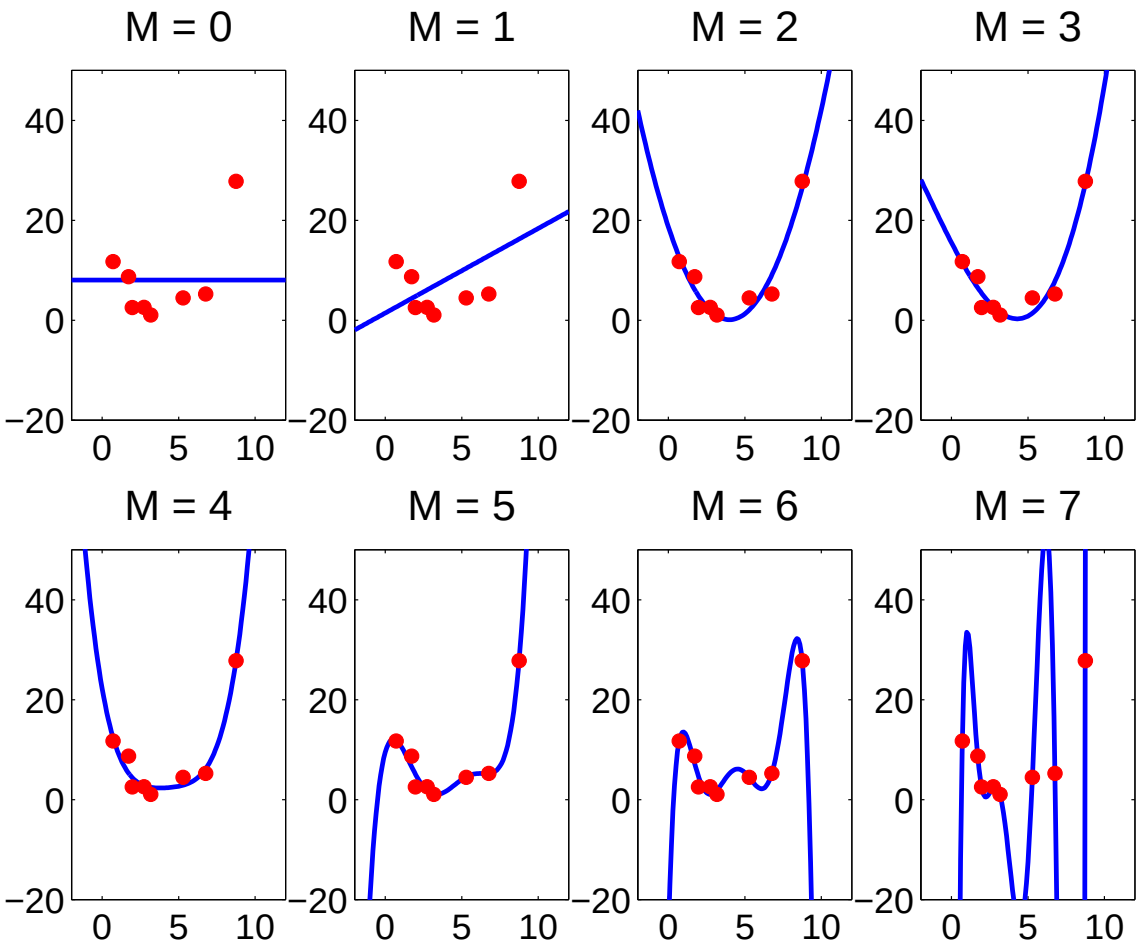
Bayesian Learning of Model Structure

Zoubin Ghahramani
Gatsby Computational Neuroscience Unit
University College London

December 2000

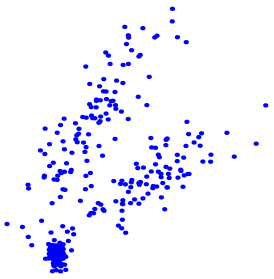
http://www.gatsby.ucl.ac.uk/

Model structure and overfitting: a simple example

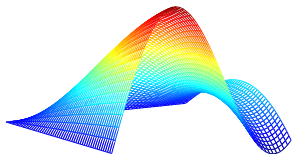


Model Selection Questions

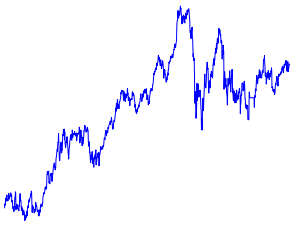
How many clusters in the data?



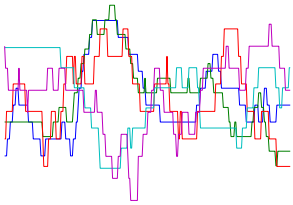
What is the intrinsic dimensionality of the data?



Is this input relevant to predicting that output?



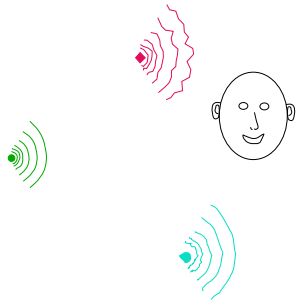
What is the order of this dynamical system?



How many states for this hidden Markov model?

SVYDAAAQLTADVKKDLRDSWKVIGSDKKGNGVALMTTY

How many auditory sources in the input?



Bayesian Learning

data Y
models $\mathcal{M}_1 \ldots, \mathcal{M}_n$
parameter sets $\theta_1 \ldots, \theta_n$

(let's ignore hidden variables X for the moment, this will just introduce another level of averaging/integration)

Model Selection:

$$P(\mathcal{M}_i|Y) = \frac{P(Y|\mathcal{M}_i)P(\mathcal{M}_i)}{P(Y)}$$

Model Averaging:

$$P(y|Y) = \sum_i P(y|Y, \mathcal{M}_i)P(\mathcal{M}_i|Y)$$

Ockham's Razor

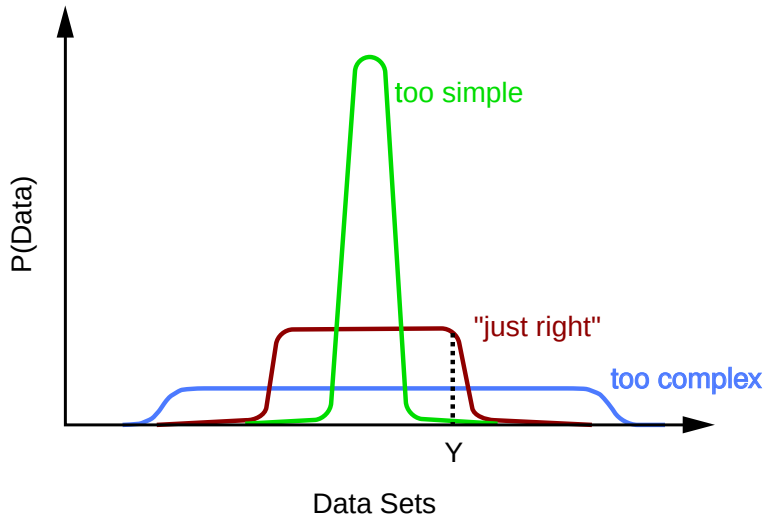
$$P(\mathcal{M}_i|Y) = \frac{P(Y|\mathcal{M}_i)P(\mathcal{M}_i)}{P(Y)}$$

$$P(Y|\mathcal{M}_i) = \int_{\theta_i} P(Y|\theta_i, \mathcal{M}_i)P(\theta_i|\mathcal{M}_i)$$

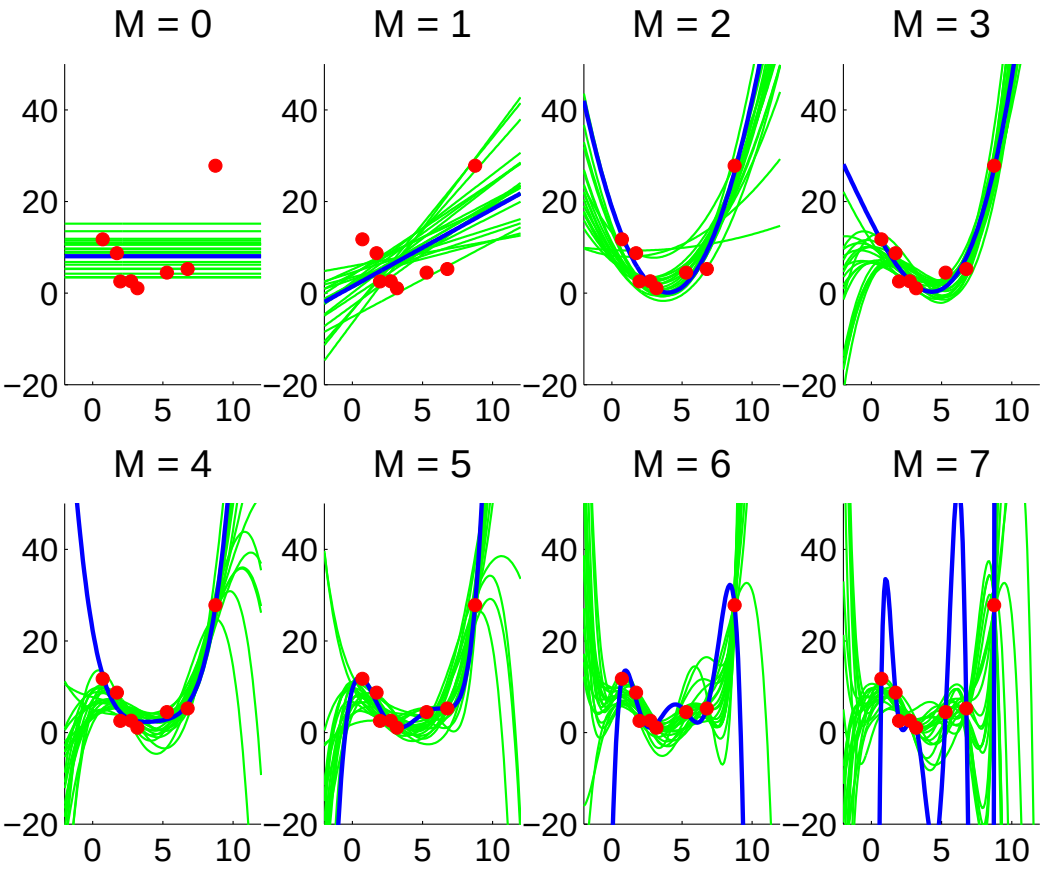
What is the probability that if you *randomly selected* parameter values from your model class you would generate data set Y ?

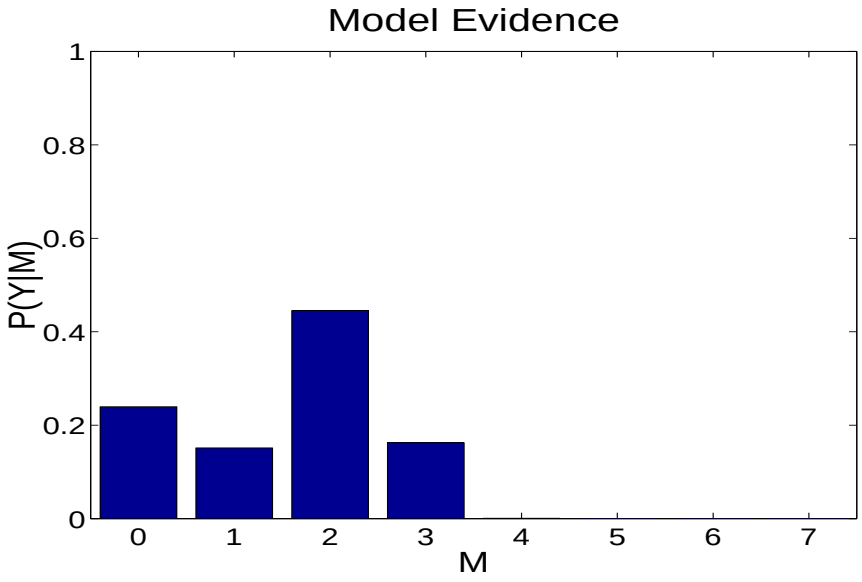
Model classes that are too simple will be very unlikely to generate that particular data set.

Model classes that are too complex can generate many possible data sets, so again, they are unlikely to generate that particular data set at random.



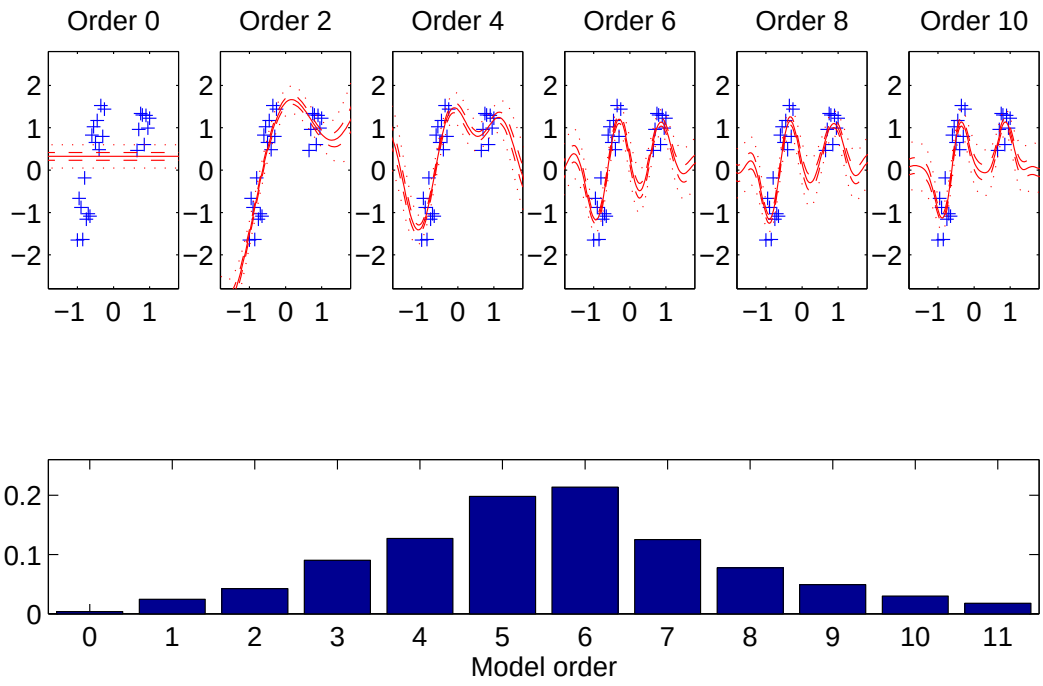
Bayesian Model Selection



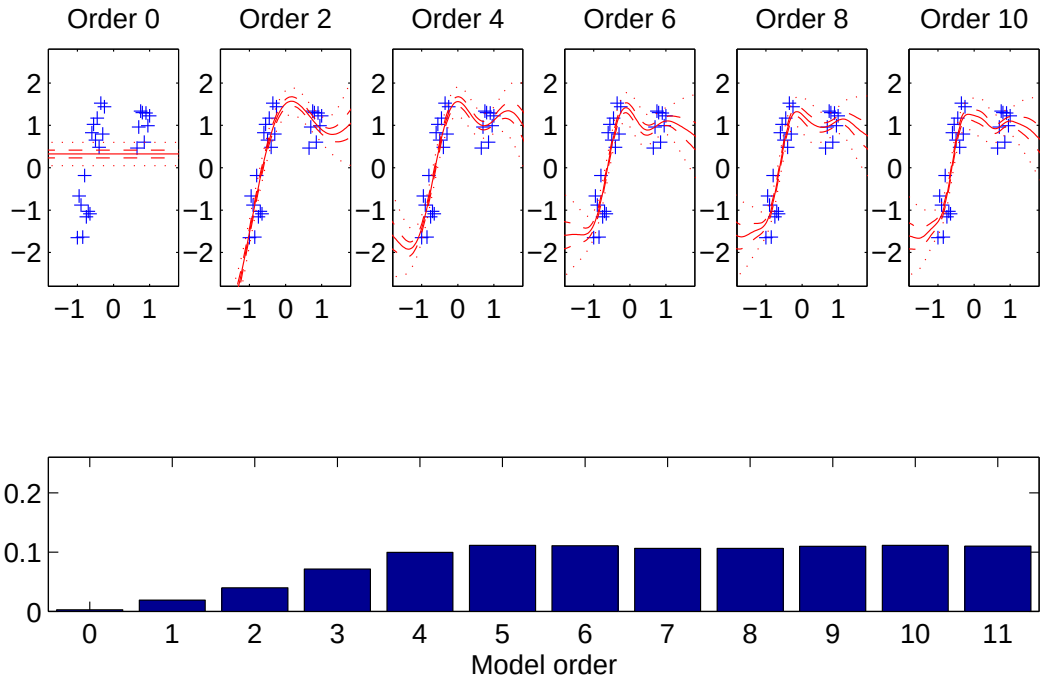


A subtle point about Ockham's Hill

Unscaled models:



Scaled models:



Practical Bayesian approaches

- Laplace approximations:
  - Appeals to Central Limit Theorem making a Gaussian approximation about maximum *a posteriori* parameter estimate.
- Large sample approximations (e.g. BIC).
- Markov chain Monte Carlo methods (MCMC):
  - In the limit are guaranteed to converge, but:
  - Many samples required to ensure accuracy.
  - Hard to assess convergence.
- Variational approximations...

Variational Bayesian Learning

Let the hidden states be $\mathbf{x}$, data $\mathbf{y}$ and the parameters $\boldsymbol{\theta}$.
We can lower bound the evidence (Jensen's inequality):

$$\begin{aligned}
\ln P(\mathbf{y}|\mathcal{M}) &= \ln \int d\mathbf{x}\, d\boldsymbol{\theta}\; P(\mathbf{y},\mathbf{x},\boldsymbol{\theta}|\mathcal{M}) \\
&= \ln \int d\mathbf{x}\, d\boldsymbol{\theta}\; Q(\mathbf{x},\boldsymbol{\theta}) \frac{P(\mathbf{y},\mathbf{x},\boldsymbol{\theta})}{Q(\mathbf{x},\boldsymbol{\theta})} \\
&\geq \int d\mathbf{x}\, d\boldsymbol{\theta}\; Q(\mathbf{x},\boldsymbol{\theta}) \ln \frac{P(\mathbf{y},\mathbf{x},\boldsymbol{\theta})}{Q(\mathbf{x},\boldsymbol{\theta})}.
\end{aligned}$$

Use a simpler, factorised approximation to $Q(\mathbf{x},\boldsymbol{\theta})$:

$$\begin{aligned}
\ln P(\mathbf{y}) &\geq \int d\mathbf{x}\, d\boldsymbol{\theta}\; Q_{\mathbf{x}}(\mathbf{x}) Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \frac{P(\mathbf{y},\mathbf{x},\boldsymbol{\theta})}{Q_{\mathbf{x}}(\mathbf{x}) Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \\
&= \mathcal{F}(Q_{\mathbf{x}}(\mathbf{x}), Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \mathbf{y}).
\end{aligned}$$

Maximising this lower bound, $\mathcal{F}$, leads to **EM-like** updates:

$$\begin{aligned}
Q^*_{\mathbf{x}}(\mathbf{x}) &\propto \exp \langle \ln P(\mathbf{x},\mathbf{y}|\boldsymbol{\theta}) \rangle_{Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} && E-like\ step \\
Q^*_{\boldsymbol{\theta}}(\boldsymbol{\theta}) &\propto P(\boldsymbol{\theta}) \exp \langle \ln P(\mathbf{x},\mathbf{y}|\boldsymbol{\theta}) \rangle_{Q_{\mathbf{x}}(\mathbf{x})} && M-like\ step
\end{aligned}$$

Equivalent to minimizing KL-divergence between the *approximating* and *true* posteriors.

Conjugate-Exponential models

Condition (1). The joint probability over *variables* is in the exponential family:

$$P(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) = f(\mathbf{x}, \mathbf{y})\ g(\boldsymbol{\theta}) \exp\left\{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{x}, \mathbf{y})\right\}$$

where $\boldsymbol{\phi}(\boldsymbol{\theta})$ is the vector of *natural parameters*.

Condition (2). The prior over *parameters* is conjugate to this joint probability:

$$P(\boldsymbol{\theta}|\eta, \boldsymbol{\nu}) = h(\eta, \boldsymbol{\nu})\ g(\boldsymbol{\theta})^\eta \exp\left\{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \boldsymbol{\nu}\right\}$$

where η and $\boldsymbol{\nu}$ are hyperparameters of the prior.

Conjugate-exponential (**CE**) models satisfy **(1)** and **(2)**.

- Conjugate priors: η: number of pseudo-observations,
  $\boldsymbol{\nu}$: values of pseudo-observations.
- Usually (2) implies (1).

Conjugate-Exponential examples

In the **CE** family:

- Gaussian mixtures
- factor analysis, probabilistic PCA
- hidden Markov models and factorial HMMs
- linear dynamical systems and switching models
- discrete-variable belief networks

Other as yet undreamt-of models can combine Gaussian, Gamma, Poisson, Dirichlet, Wishart, Multinomial and others.

Not in the **CE** family:

- Boltzmann machines (no conjugacy)
- logistic regression (no conjugacy)
- sigmoid belief networks (not exponential)
- independent components analysis (not exponential)

Note: one can often approximate these models with models in the **CE** family.

Theoretical Results

Theorem 1 Given an iid data set $\mathbf{y} = (\mathbf{y}_1, \ldots \mathbf{y}_n)$, if the model is **CE** then:

(a) $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ is also conjugate, *i.e.*

$$Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = h(\tilde{\eta}, \tilde{\boldsymbol{\nu}}) g(\boldsymbol{\theta})^{\tilde{\eta}} \exp\left\{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \tilde{\boldsymbol{\nu}}\right\}$$

(b) $Q_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^n Q_{\mathbf{x}_i}(\mathbf{x}_i)$ is of the same form as in the E step of regular EM, but using pseudo parameters computed by averaging over $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$

$$\begin{aligned} Q_{\mathbf{x}_i}(\mathbf{x}_i) &\propto f(\mathbf{x}_i, \mathbf{y}_i) \exp\left\{\overline{\boldsymbol{\phi}}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{x}_i, \mathbf{y}_i)\right\} \\ &= P(\mathbf{x}_i | \mathbf{y}_i, \overline{\boldsymbol{\phi}}(\boldsymbol{\theta})) \end{aligned}$$

KEY points:
(a) the approximate parameter posterior is of the same form as the prior;
(b) the *approximate* hidden variable posterior, averaging over *all* parameters, is of the same form as the *exact* hidden variable posterior for a *single* setting of the parameters.

The Variational EM algorithm

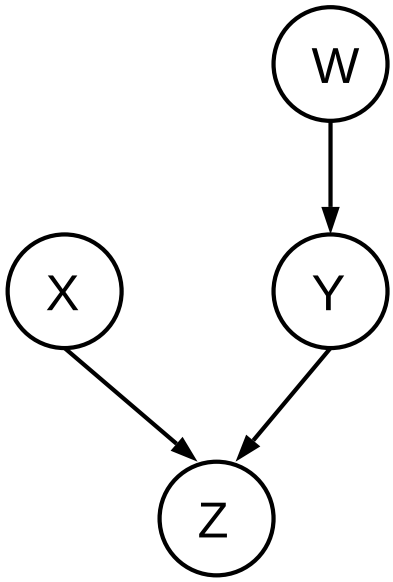
VE Step: Compute the expected sufficient statistics $\mathbf{t}(\mathbf{y}) = \sum_i \overline{\mathbf{u}}(\mathbf{x}_i, \mathbf{y}_i)$ under the hidden variable distributions $Q_{\mathbf{x}_i}(\mathbf{x}_i)$.

VM Step: Compute expected natural parameters $\overline{\boldsymbol{\phi}}(\boldsymbol{\theta})$ under the parameter distribution given by $\tilde{\eta}$ and $\tilde{\boldsymbol{\nu}}$.

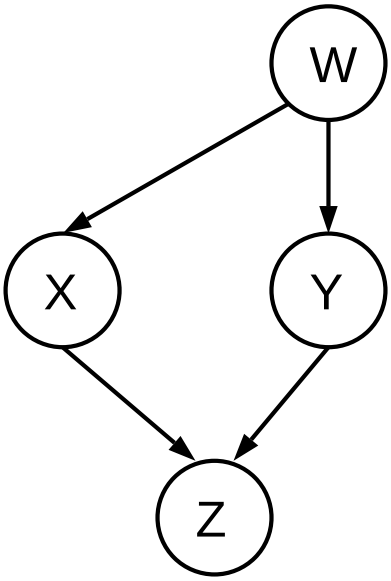
Properties:

- VE step has same complexity as corresponding E step.
- Reduces to the EM algorithm if $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$. M step then involves re-estimation of $\boldsymbol{\theta}^*$.
- $\mathcal{F}$ increases monotonically, and incorporates the model complexity penalty.

Graphical models and propagation algorithms



Singly-connected nets
The *belief propagation* algorithm.



Multiply-connected nets
The *junction tree* algorithm.

These are efficient ways of applying Bayes rule using the conditional independence relationships implied by the graphical model.

Propagation Algorithms for VEM

Corollary 1: CE Belief Networks. If the model is **CE**, with hidden and visible variables $\mathbf{z} = (\mathbf{x}, \mathbf{y})$, and satisfies a belief network factorisation

$$P(\mathbf{z}|\boldsymbol{\theta}) = \prod_j P(z_j|\mathbf{z}_{p_j}, \boldsymbol{\theta})$$

then the approximate joint satisfies the same BN factorisation but with $\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \overline{\boldsymbol{\phi}}(\boldsymbol{\theta})$, i.e.

$$Q_{\mathbf{Z}}(\mathbf{z}) = \prod_j Q(z_j|\mathbf{z}_{p_j}, \tilde{\boldsymbol{\theta}})$$

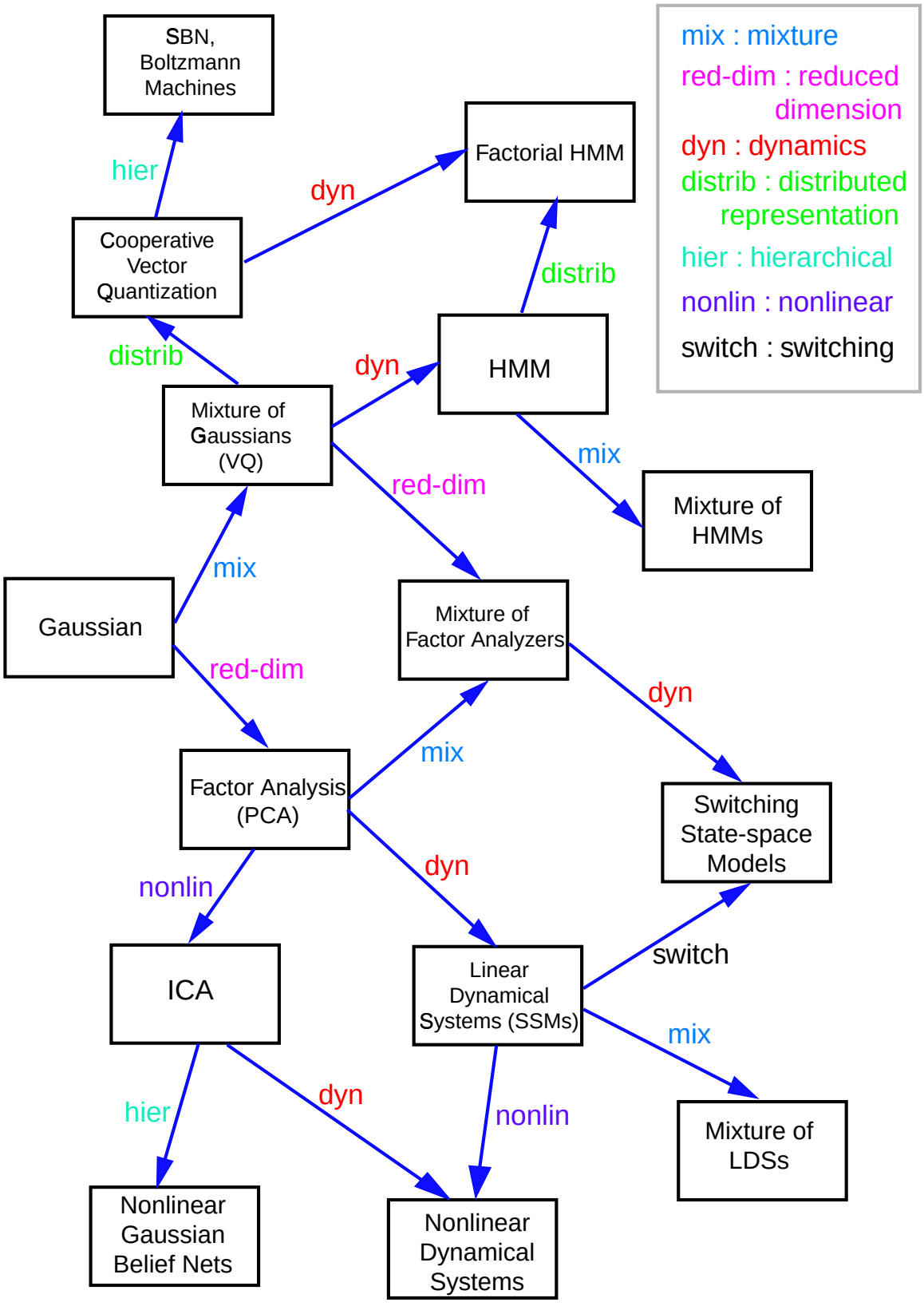
Corollary 2: CE Markov Networks. If the model is a CE Markov network then the approximate joint distribution is

$$Q_{\mathbf{Z}}(\mathbf{z}) = \tilde{g} \prod_j \psi_j(C_j, \tilde{\boldsymbol{\theta}})$$

where the clique potentials have exactly the same form as in the model, but with natural parameters $\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \overline{\boldsymbol{\phi}}(\boldsymbol{\theta})$.

Intuition: We can use the junction tree, belief propagation, Kalman filter, etc, algorithms in the VE step of VEM, but *using expected natural parameters*.

A Generative Model for Generative Models



Variational Bayes & Ensemble Learning

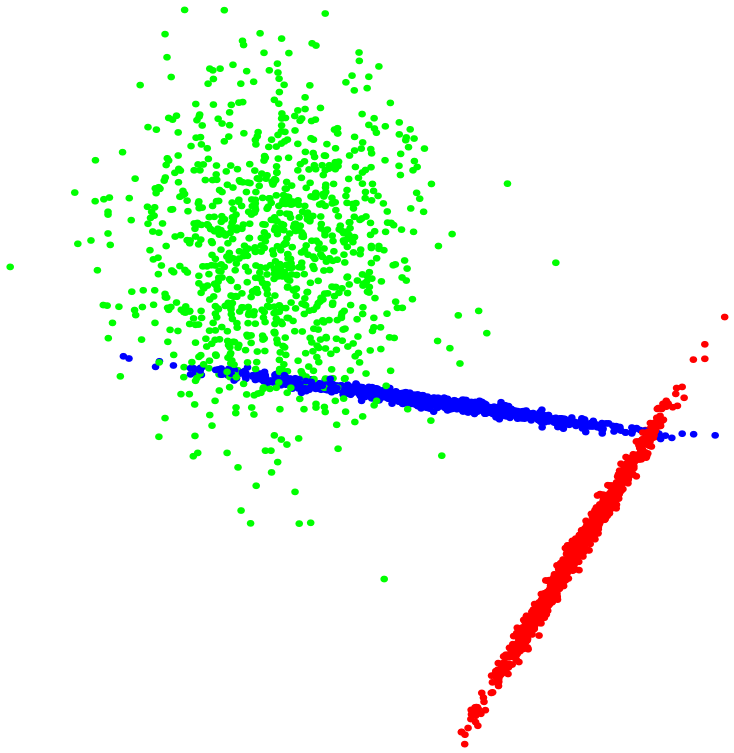
- multilayer perceptrons (Hinton & van Camp, 1993)
- mixture of experts (Waterhouse, MacKay & Robinson, 1996)
- hidden Markov models (MacKay, 1995)
- other work by Jaakkola, Barber, Bishop, Tipping, etc

Examples of VB Learning Model Structure

Model learning has been treated with variational Bayesian techniques for:

- mixtures of factor analysers (Ghahramani & Beal, 1999)
- mixtures of Gaussians (Attias, 1999)
- independent components analysis (Attias, 1999; Miskin & MacKay, 2000; Valpola 2000)
- principal components analysis (Bishop, 1999)
- linear dynamical systems (Ghahramani & Beal, 2000)
- mixture of experts (Ueda & Ghahramani, 2000)
- hidden Markov models (Ueda & Ghahramani, in prep)

Mixture of Factor Analysers



Goal:

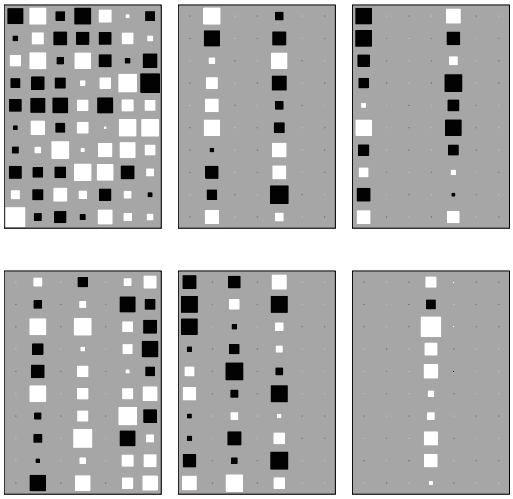
- Infer number of clusters
- Infer intrinsic dimensionality of each cluster

Under the assumption that each cluster is Gaussian

Mixture of Factor Analysers

True data: 6 Gaussian clusters with dimensions:
(1 7 4 3 2 2) embedded in 10 dimensions

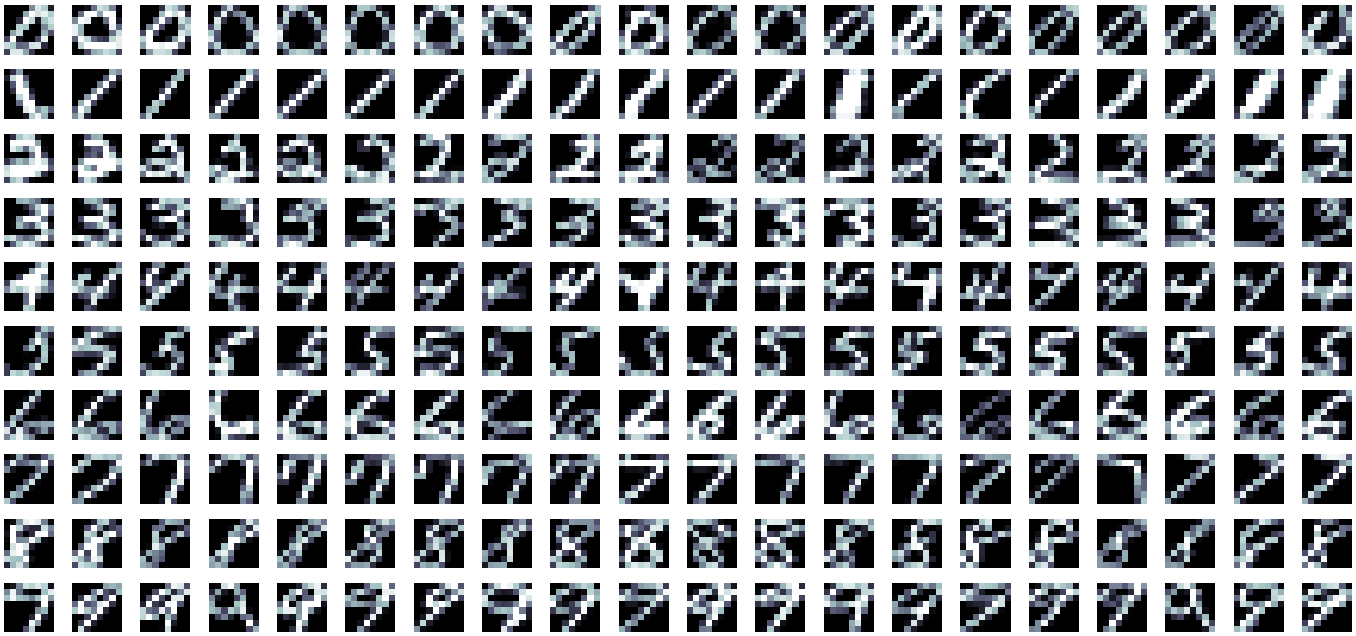
Inferred structure:



number of points per cluster	intrinsic dimensionalities					
	1	7	4	3	2	2
8	2				1	
8	1		2			
16	1		4			2
32	1	6	3	3	2	2
64	1	7	4	3	2	2
128	1	7	4	3	2	2

- Finds the clusters and dimensionalities efficiently.
- The model complexity reduces in line with the lack of data support.

Digit Clustering



- Trained on 700 8x8 images of each digit (CEDAR ROM).
- Determines the number of clusters (styles) required
- ARD determines the number of deformations
- The number to the right of each digit is the dimension


2 4 3 4 3 3
3 5 5 2 4
1 5 3 4 3 4 3 3 2 2 1 4 5 5 5 4 4 5 5 5
5 4 5 4 5 4 5 4 5
3 4 5 5 4 2 4 5 3 2
2 4 4 5 5 4 3 5
5 5 3 4 4 4 4 4 4
3 5 5 3 4 3 4 3
4 4 5 3 4 5 5 4 4
4 5 3 5 3 5 4 3

Classified

True	0	1	2	3	4	5	6	7	8	9
0	687	7	.	2	.	1	3	.	.	.
1	.	699	.	.	.	1	.	.	.	.
2	1	7	671	1	2	.	7	4	7	.
3	.	3	7	629	.	27	1	2	30	1
4	.	3	4	.	609	.	3	14	1	66
5	3	7	5	46	.	618	5	.	16	.
6	.	3	.	.	32	1	664	.	.	.
7	.	2	3	1	3	.	.	589	2	100
8	1	14	1	27	2	43	1	4	603	4
9	.	4	1	.	13	.	.	65	3	614

Training data

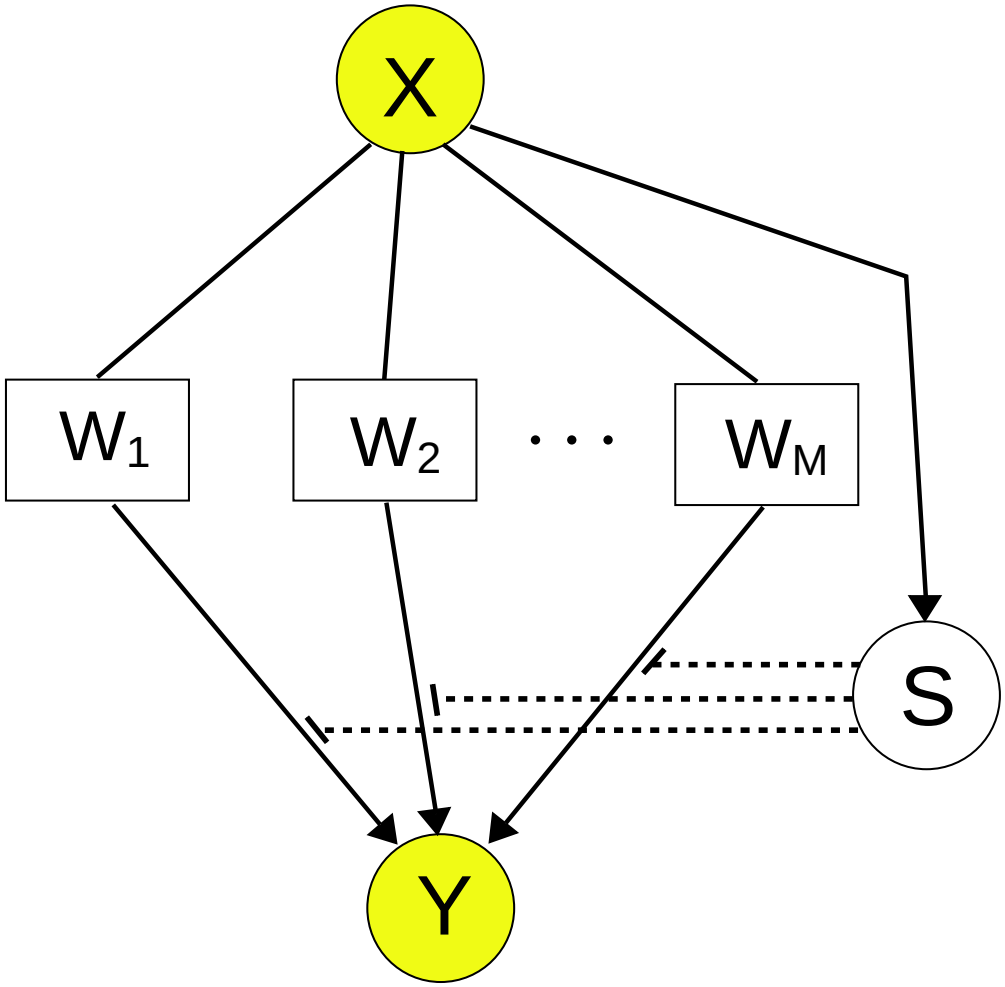
Classified

True	0	1	2	3	4	5	6	7	8	9
0	196	1	.	.	.	1	2	.	.	.
1	.	200	.	.	.	.	.	.	.	.
2	.	1	186	1	1	1	2	1	6	1
3	.	2	1	181	.	10	.	.	5	1
4	.	1	.	.	175	.	.	1	.	23
5	1	2	.	13	.	180	.	1	2	1
6	.	1	.	.	5	1	193	.	.	.
7	.	2	1	.	.	.	.	176	.	21
8	.	1	.	5	.	9	.	1	179	5
9	.	4	.	.	3	.	.	17	.	176

Test data

- Each image is classified using hard assignment
- Unsupervised classif: 8.8% train, 7.9% test error.
- K-means (same # of clusters): 12.2%, 13.3% error.

Mixture of Experts



Learning Mixture of Experts Structure

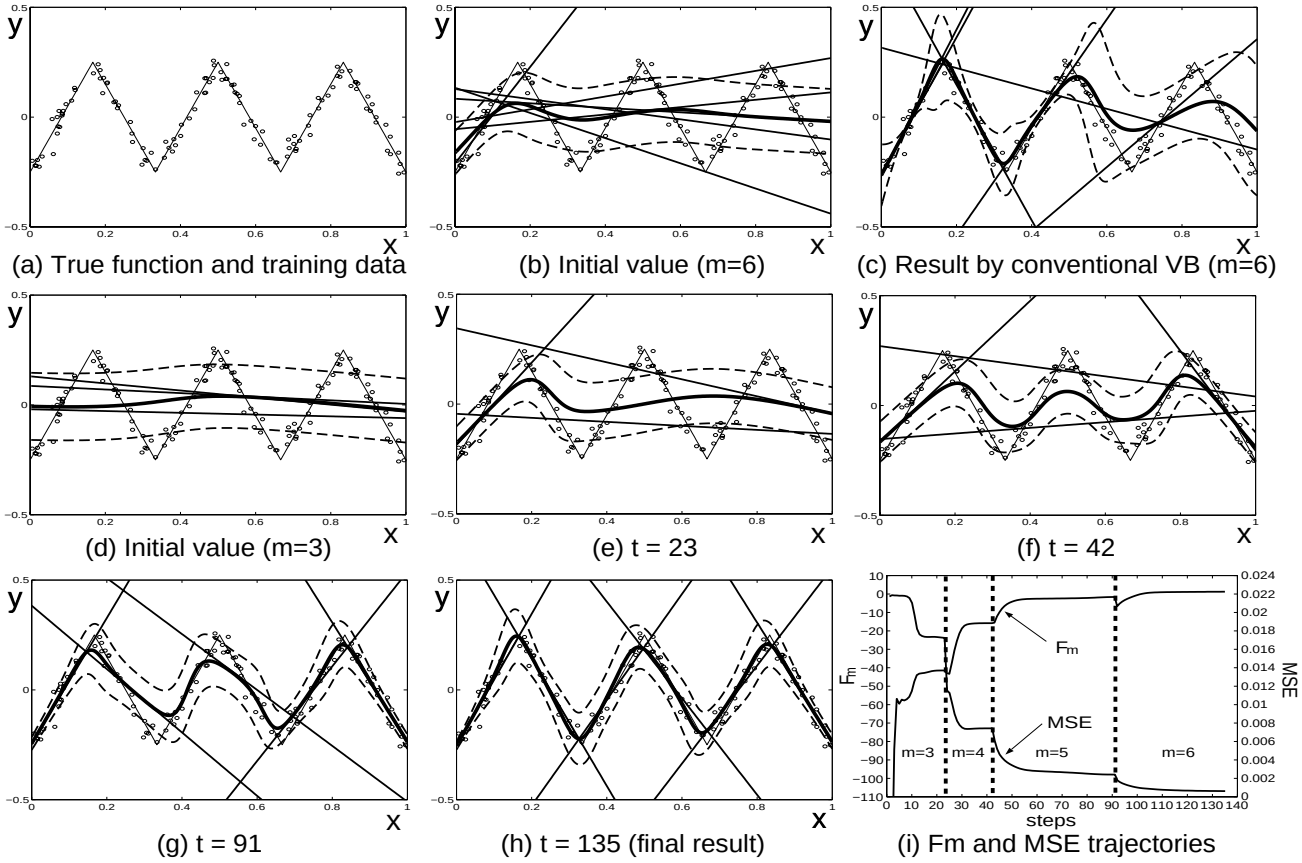
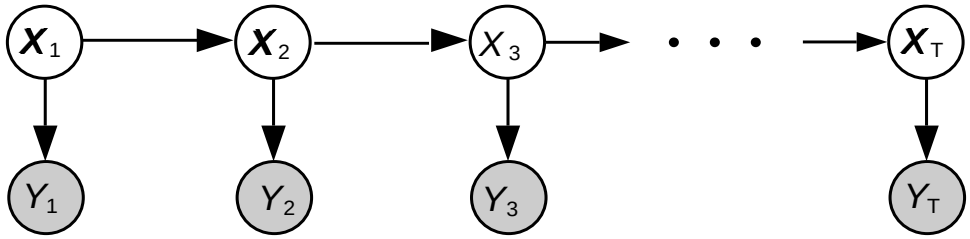


Figure 1: Result for synthetic data.

Linear Dynamical Systems



- Assumes $\mathbf{y}_t$ generated from a *hidden* state variable $\mathbf{x}_t$, and that the sequence of $\mathbf{x}_{1:T}$ is Markov.
- If transition and output functions are linear, time-invariant, and noise distributions are Gaussian, this is a Linear-Gaussian state-space model:

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

- Dynamic generalisation of factor analysis.
- Three levels of inference:

I Given data, structure and parameters,
Kalman smoothing $\rightarrow$ hidden state;

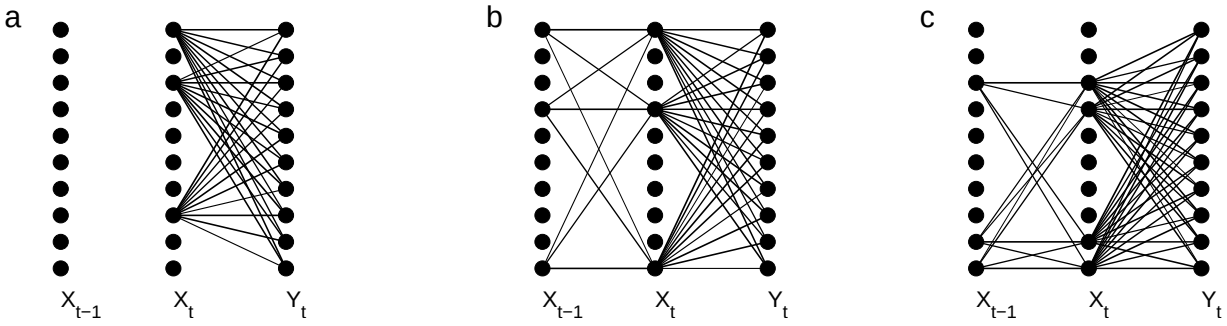
II Given data and structure,
EM $\rightarrow$ hidden state and parameter point estimates;

III Given data only, **VEM** $\rightarrow$ **model structure** *and* **distributions over parameters** *and* **hidden state**.

Linear Dynamical Systems Results

Inferring model structure (synthetic):

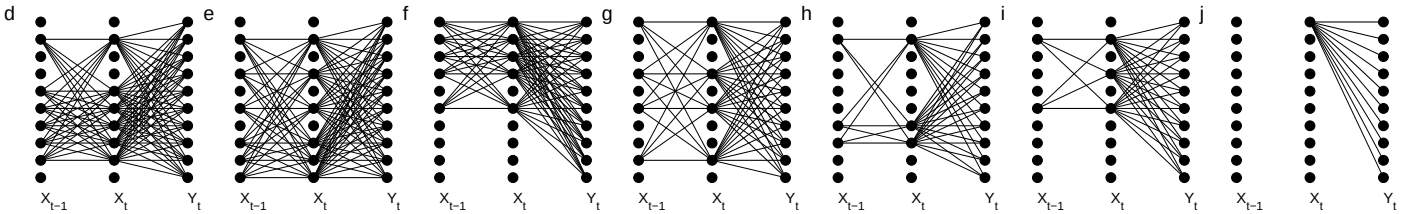
a) SSM(0,3) i.e. FA b) SSM(3,3) c) SSM(3,4)



Inferred model complexity reduces with less data:

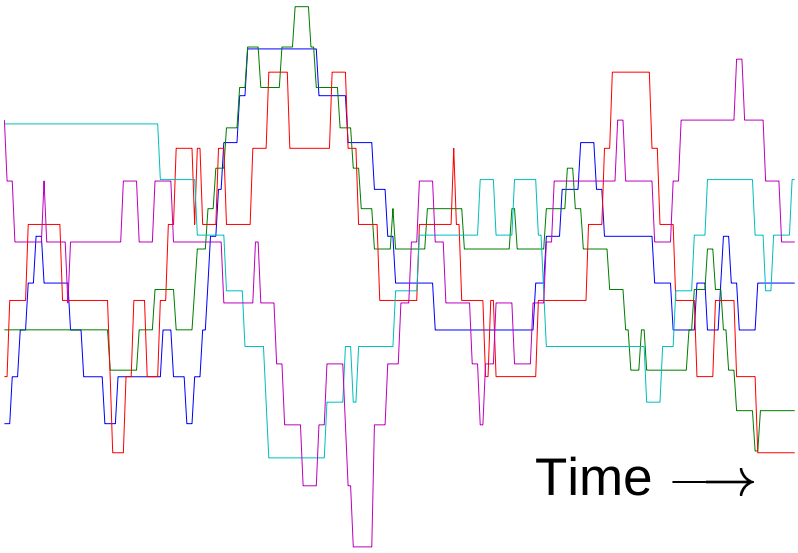
True model: • SSM(6,6) • 10-dim observation vector.

400 350 250 100 30 20 10



Steel Plant Data

- 38 sensors (temperatures, pressures, etc.) sampled at 2Hz from a continuous casting process for 150 secs.
- Sensors *covaried* and were *temporally correlated*, suggesting an LDS could capture some of its structure.



- **True model**: **???**.
- **Inferred model**: **16** state variables required, of which **14** emitted outputs.

Sampling from Variational Approximations

Sampling $\theta_m \sim Q(\theta)$ gives us estimates of:

- The Exact Predictive Density:

$$\begin{aligned} P(y|Y) &= \int d\theta \; P(y|\theta)P(\theta|Y) \\ &= \int d\theta \; Q(\theta)P(y|\theta)\frac{P(\theta|Y)}{Q(\theta)} \\ &\approx \sum_{m=1}^{M} P(y|\theta_m)\,\omega_m \end{aligned}$$

weights: $\omega_m = \frac{1}{\Omega}\frac{P(\theta_m,Y)}{Q(\theta_m)}$, with Ω s.t. $\sum \omega_m = 1$

- The True Evidence:

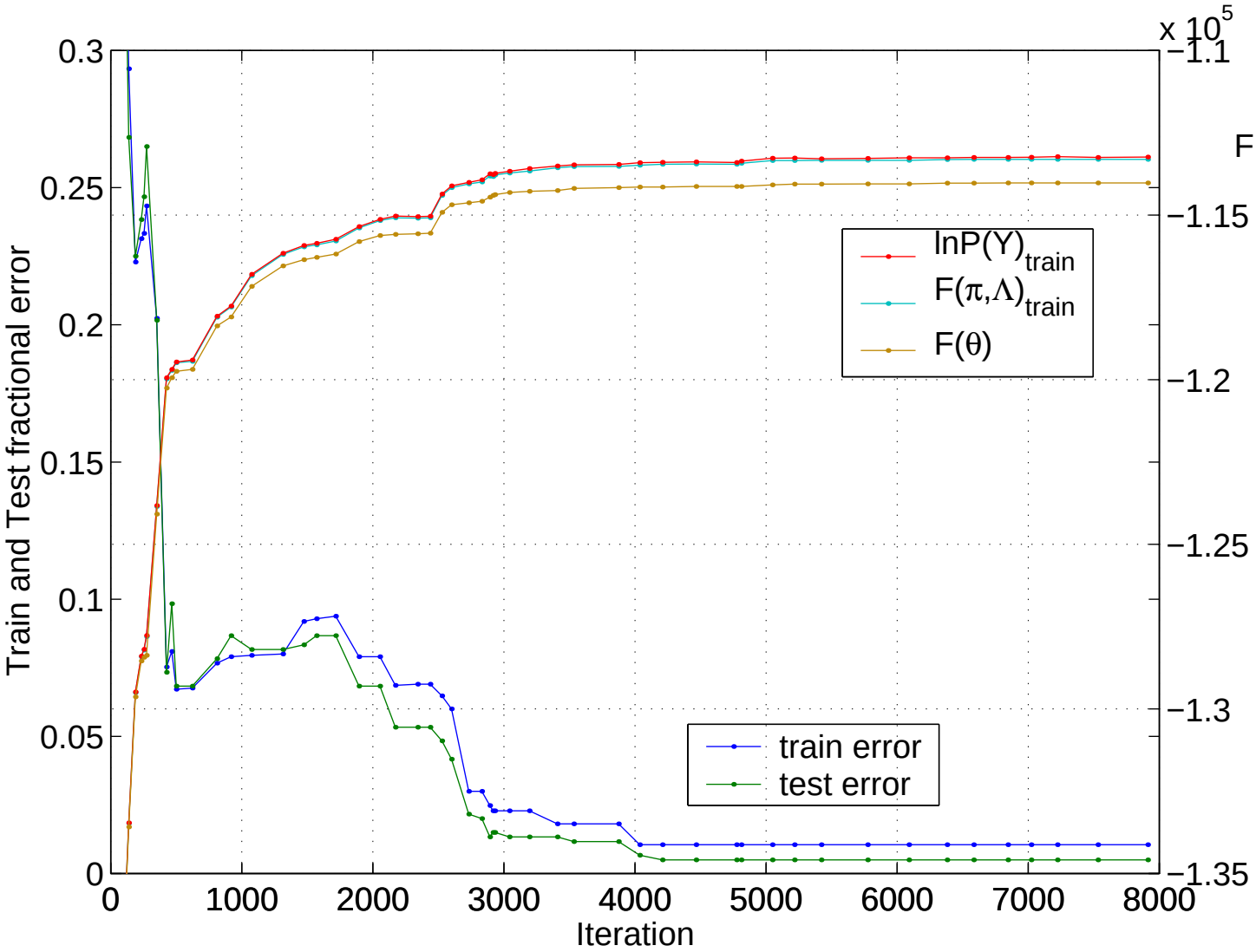
$$P(Y|\mathcal{M}) = \int d\theta \; Q(\theta)\frac{P(\theta,Y)}{Q(\theta)} = \langle \Omega\omega \rangle$$

- The KL Divergence:

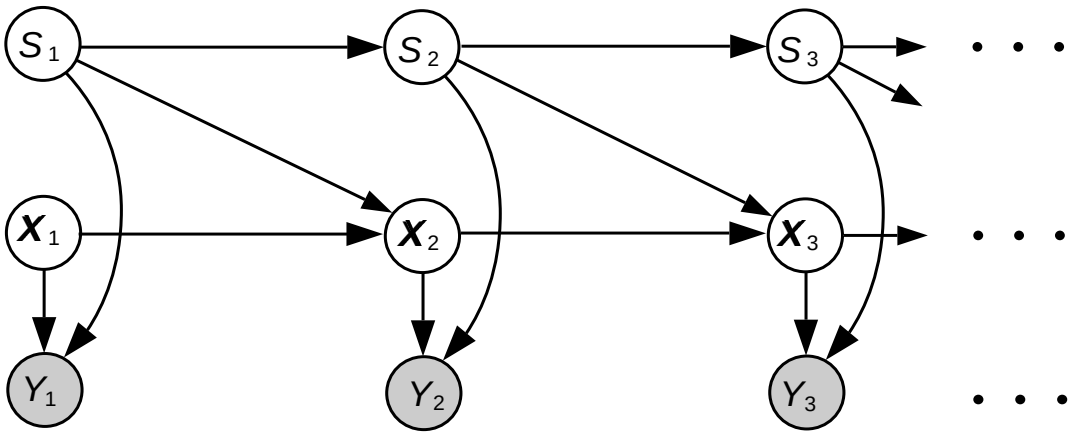
$$\mathrm{KL}(Q(\theta)\|P(\theta|Y)) = \ln\langle\omega\rangle - \langle \ln \omega\rangle.$$

Note: same weights can be used for all three!

Evolution of $\mathcal{F}$, true evidence and KL-divergence



Switching state-space model



Switch transitions:

$$P(s_t = i | s_{t-1} = j) = T_{ij}$$

Hidden state dynamics:

$$P(\mathbf{x}_t | s_{t-1}, \mathbf{x}_{t-1}) = \mathsf{N}(A_{s_{t-1}} \mathbf{x}_{t-1}, Q_{s_{t-1}})$$

Output function:

$$P(\mathbf{y}_t | s_t, \mathbf{x}_t) = \mathsf{N}(C_{s_t} \mathbf{x}_t, R_{s_t})$$

Contains as special cases: mixtures of factor analysers, mixtures of linear dynamical systems, Gaussian-output HMMs, mixtures of Gaussians, ...

is a conjugate-exponential belief network

Summary & Conclusions

- Bayesian learning avoids overfitting and can be used to learn model structure
- Tractable Bayesian learning using variational methods
- Conjugate-exponential families
- Variational EM and Propagation theorems
- Some examples
- Sampling from variational approximation estimates:
  - the true evidence
  - the KL divergence
  - the exact predictive density
- Combining variational methods and sampling: best of both worlds, fast and reliable algorithms for Bayesian learning?

Application Areas

- computational molecular biology
- financial time series prediction
- speech and video processing
- analysis of functional neuro-imaging data

Future Directions and Other Interests

- extensions to other models (HMMs, hierarchies)
- combination with other approximations (MCMC, loopy)
- extension to influence diagrams for decision/control/RL
- inferring causality
- human motor control and computational neuroscience