An introduction to Expectation Propagation

Tom Minka

Microsoft Research, Cambridge, UK

Predictive Multiscale Materials Modelling

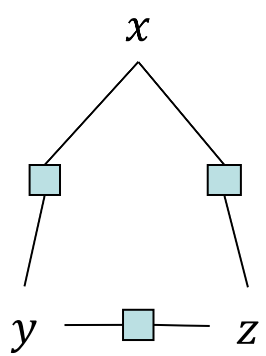
2015

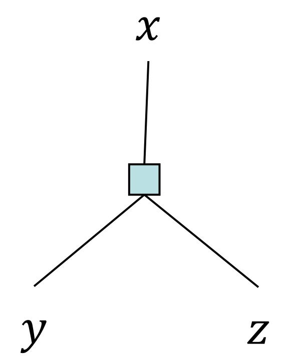
Bayesian paradigm

- Consistent use of probability theory for representing unknowns (parameters, latent variables, missing data, choice of model)
- Unifies the problems of prediction, state estimation, parameter estimation, model selection
  - All reduce to computing posterior marginals
  - Can be solved using the same algorithms

Factor graphs

- Shows how a function of several variables can be factored into a product of simpler functions
- f(x,y,z) = (x+y)(y+z)(x+z)

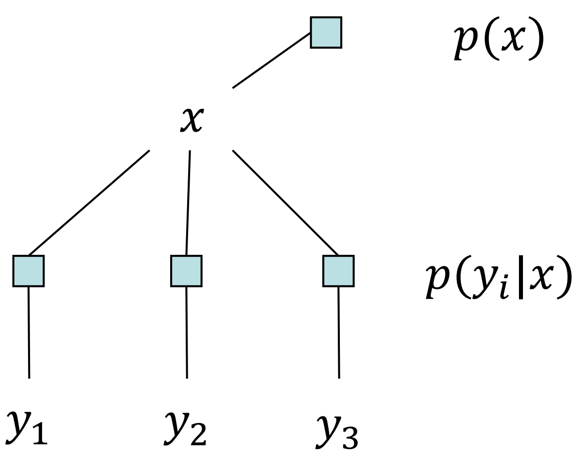




Example factor graph (Parameter estimation)

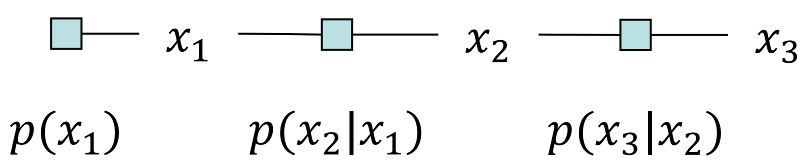
$$p(x, y_1, \ldots, y_n) = p(x) \prod_i p(y_i | x)$$

$$p(y_i | x) = N(y_i; x, 1)$$



Example factor graph (Markov chain)

$$p(x_1, \ldots, x_n) = p(x_1) \prod_i p(x_i | x_{i-1})$$



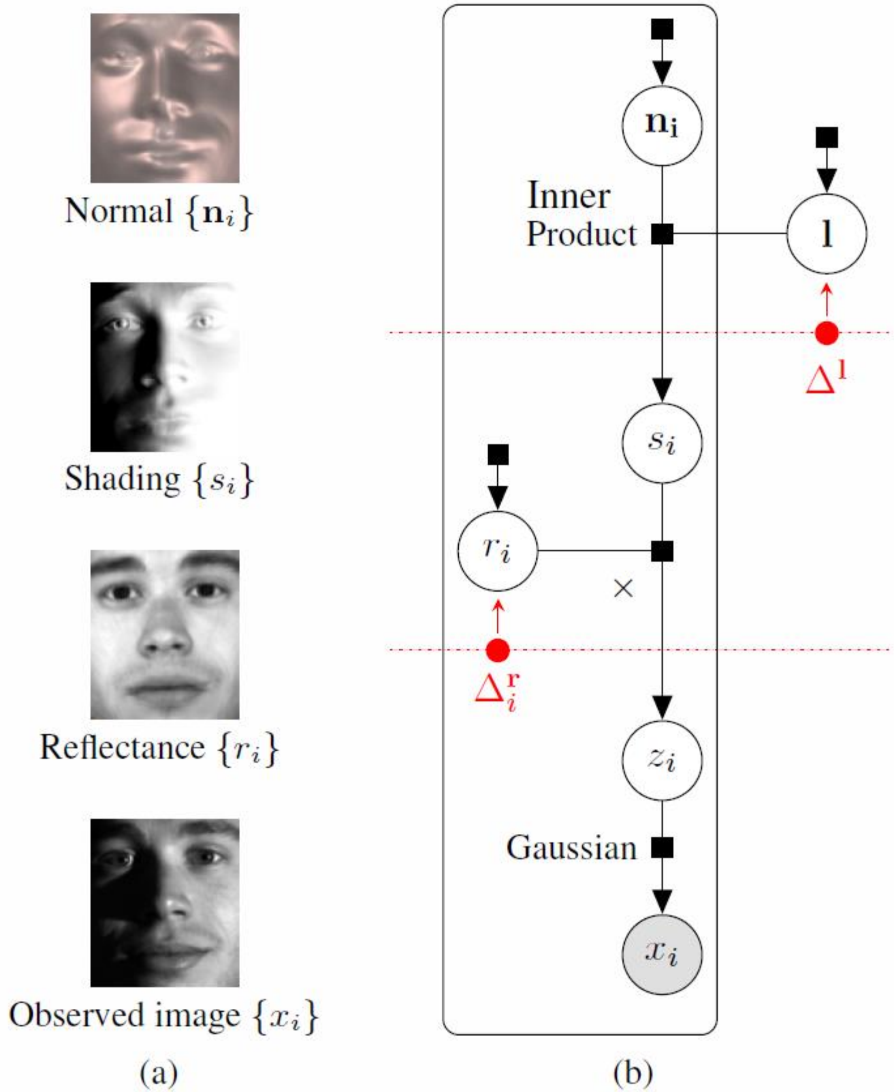


Figure 6: **The face problem.** (a) We observe an image and wish to infer the corresponding reflectance map and normal map (visualized here as 3D shape). (b) A graphical model for this problem. Symmetry priors not shown.

$$\delta(s_i - \boldsymbol{n_i} \cdot \boldsymbol{l})$$

$$x_i = (\mathbf{n_i} \cdot \mathbf{l}) \times r_i + \epsilon$$

Jampani et al, AISTATS 2015

Two tasks

- Modeling
  - What graph should I use for this data?
- Inference
  - Given the graph and data, what is the marginal of variable x?
  - Algorithms:
    - Monte Carlo
    - Variable elimination
    - Message-passing (Expectation Propagation, Variational Bayes, …)

I will contrast this with "multi-stage inference"

Multi-stage inference

1. Draw samples from the model
2. Using samples as training data, locally approximate each component of the model
3. Combine the local approximations to form a surrogate model
4. Perform exact inference in the surrogate model

Seems to be popular in materials modeling. How does it compare?

Multi-stage inference

Pros:

- Computation is amortized
- Modular development and re-use

Cons:

- Brittle – must re-train when model changes in any way
- Surrogate may miss crucial properties of the model

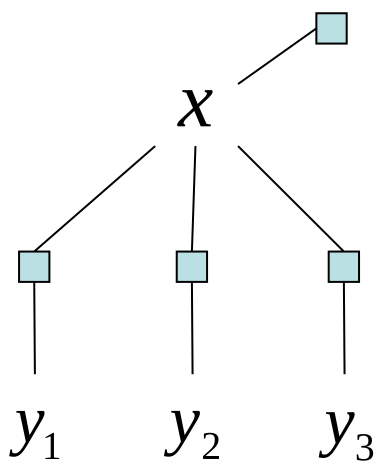
A simple example

Clutter problem

- Want to estimate x given multiple y's

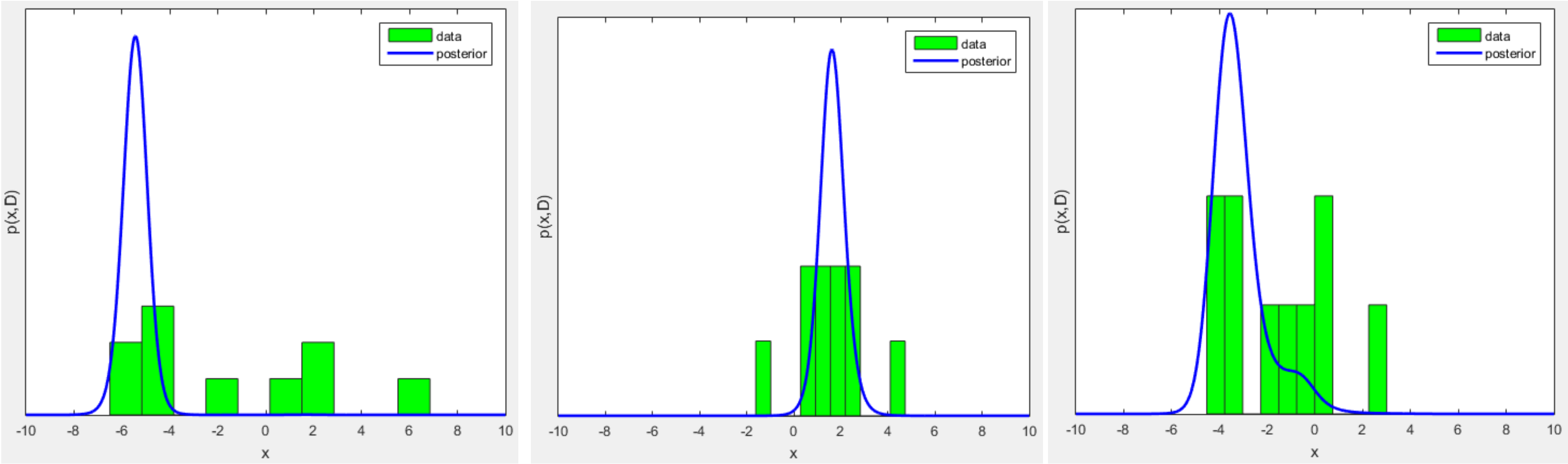
$$p(x) = \mathcal{N}(x; 0, 100)$$

$$p(y_i|x) = (0.5)\mathcal{N}(y_i; x, 1) + (0.5)\mathcal{N}(y_i; 0, 10)$$



$$p(x|y_1, \ldots, y_n) \propto p(x) \prod_i p(y_i|x)$$

Exact posterior



Multi-stage inference

- Surrogate model: each factor $p(y_i|x)$ is replaced with $\widetilde{f}_i(x) = N\big(x; m(y_i), v(y_i)\big)$
- Stage 1: Learn m, v functions
- Stage 2: Map data into Gaussian factors, multiply together to get posterior on x

What could go wrong?

Multi-stage inference

- Surrogate model no longer has the ability to reject outliers
- Regardless of how m,v functions are tuned

Strategy

- Each factor $p(y_i|x)$ is replaced with $\tilde{f}_i(x) = N(x; m_i, v_i)$
- (m_i, v_i) depend on y_i *and* the current posterior on x (excluding this factor)
  – Call this the *context*

$$q^{\setminus \mathrm{i}}(x) = p(x) \prod_{j \neq i} \tilde{f}_j(x)$$

$\tilde{f}_i(x)$ is computed by *divergence minimization*

Global divergence to local divergence

- Global divergence:

$$D(p(x) \mid\mid q(x)) =$$

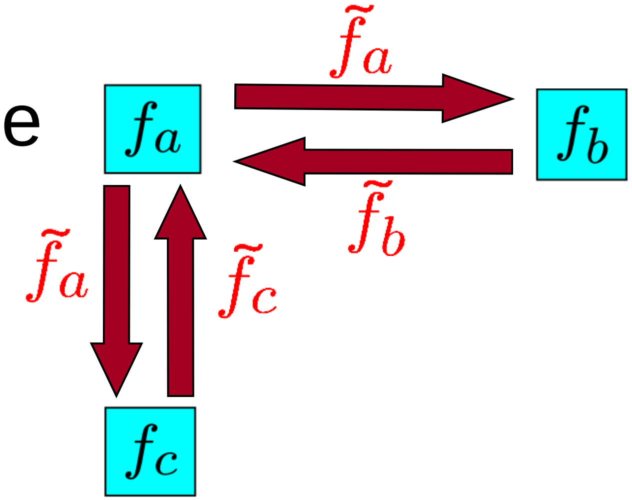
$$D(f_a(x) \prod_{b \neq a} f_b(x) \mid\mid \tilde{f}_a(x) \prod_{b \neq a} \tilde{f}_b(x))$$

- Local divergence:

$$D(f_a(x) \prod_{b \neq a} \tilde{f}_b(x) \mid\mid \tilde{f}_a(x) \prod_{b \neq a} \tilde{f}_b(x))$$

Message passing

- Messages are passed between *factors*
- Messages are factor approximations: $\tilde{f}_a(x)$
- Factor a receives $\tilde{f}_b(x), b \neq a$
  - Minimize local divergence to get $\tilde{f}_a(x)$
  - Send to other factors
  - Repeat until convergence



Approximating a factor

$$\text{proj}[p(x)] = \text{argmin}_{q \in Q} D(p||q)$$

We want $$\tilde{f}_a(x) q^{\backslash a}(x) = \text{proj}\left[f_a(x) q^{\backslash a}(x)\right]$$

Therefore $$\tilde{f}_a(x) = \frac{\text{proj}\left[f_a(x) q^{\backslash a}(x)\right]}{q^{\backslash a}(x)}$$

Divergence measures

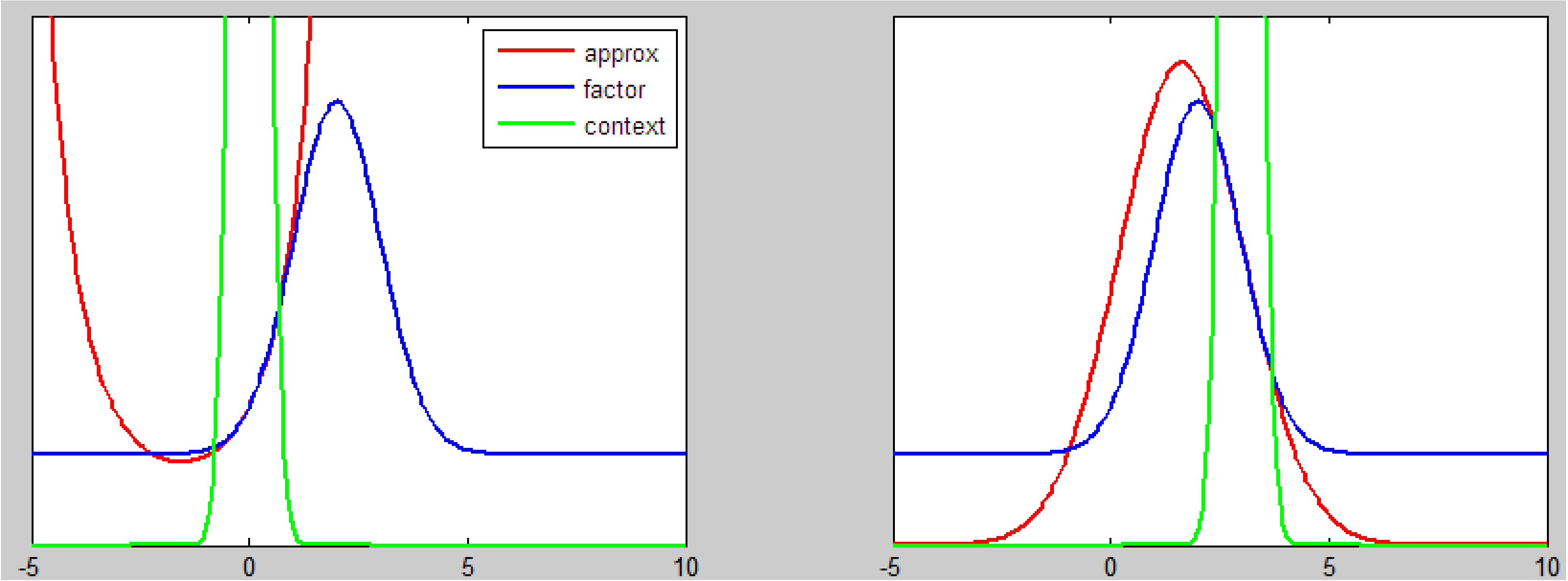
- KL divergence:
$D(p||q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx$
- Minimizing KL over Gaussians reduces to matching the mean and variance of $p(x)$
- KL can be replaced with other measures, usually to increase efficiency

Gaussian multiplication formula

$$
\begin{aligned}
\mathcal{N}(x; m_1, v_1)\mathcal{N}(x; m_2, v_2) &= \mathcal{N}(m_1; m_2, v_1 + v_2)\mathcal{N}(x; m, v) \\
\text{where } v &= \frac{1}{\frac{1}{v_1} + \frac{1}{v_2}} \\
m &= v\left(\frac{m_1}{v_1} + \frac{m_2}{v_2}\right)
\end{aligned}
$$

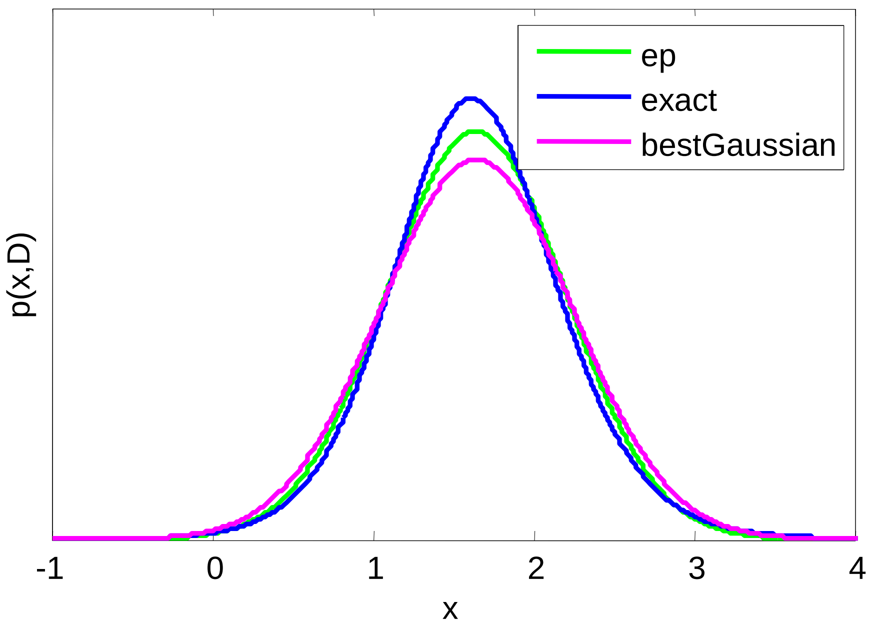
$$
\begin{aligned}
\mathcal{N}(x; m_1, v_1)/\mathcal{N}(x; m_2, v_2) &= \frac{v_2\mathcal{N}(x; m, v)}{(v_2 - v_1)\mathcal{N}(m_1; m_2, v_2 - v_1)} \\
\text{where } v &= \frac{1}{\frac{1}{v_1} - \frac{1}{v_2}} \\
m &= v\left(\frac{m_1}{v_1} - \frac{m_2}{v_2}\right)
\end{aligned}
$$

Approximation changes with context



$$p(y_i|x) = (0.5)\mathcal{N}(y_i; x, 1) + (0.5)\mathcal{N}(y_i; 0, 10)$$

Gaussian found by EP



Accuracy

Posterior mean:

exact = 1.649

ep = 1.645

laplace = 1.619

vb = 1.618

Posterior variance:

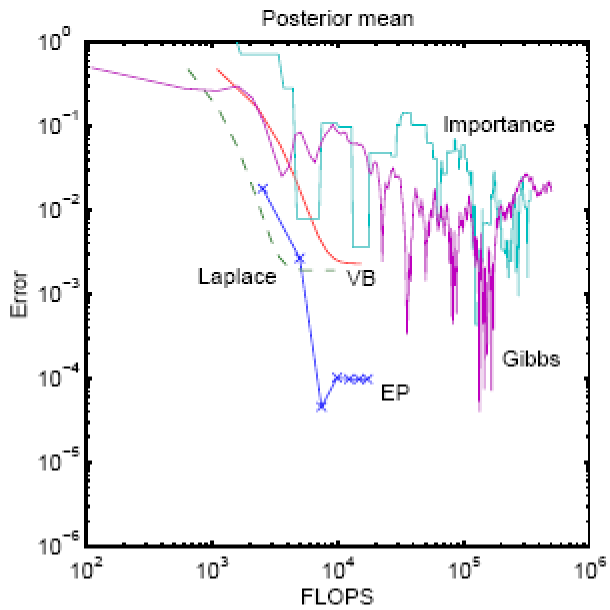
exact = 0.360

ep = 0.311

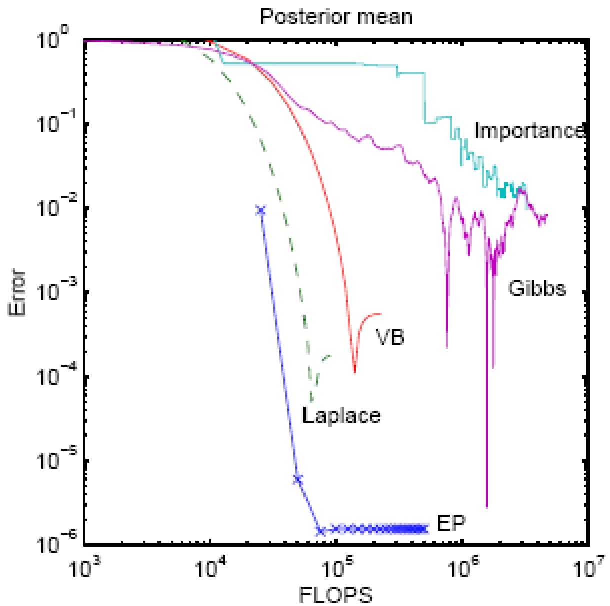
laplace = 0.235

vb = 0.171

Cost vs. accuracy



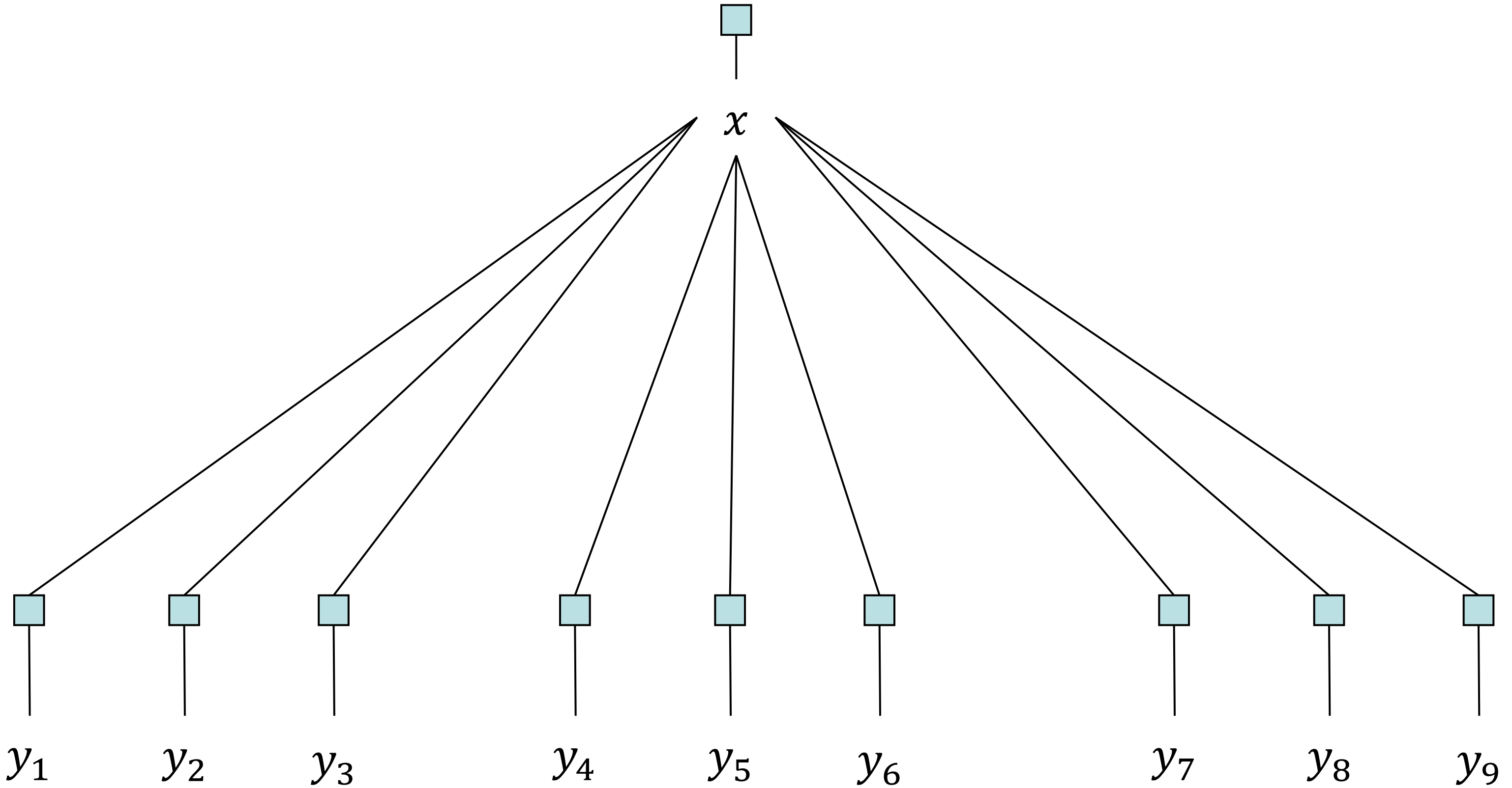
20 points



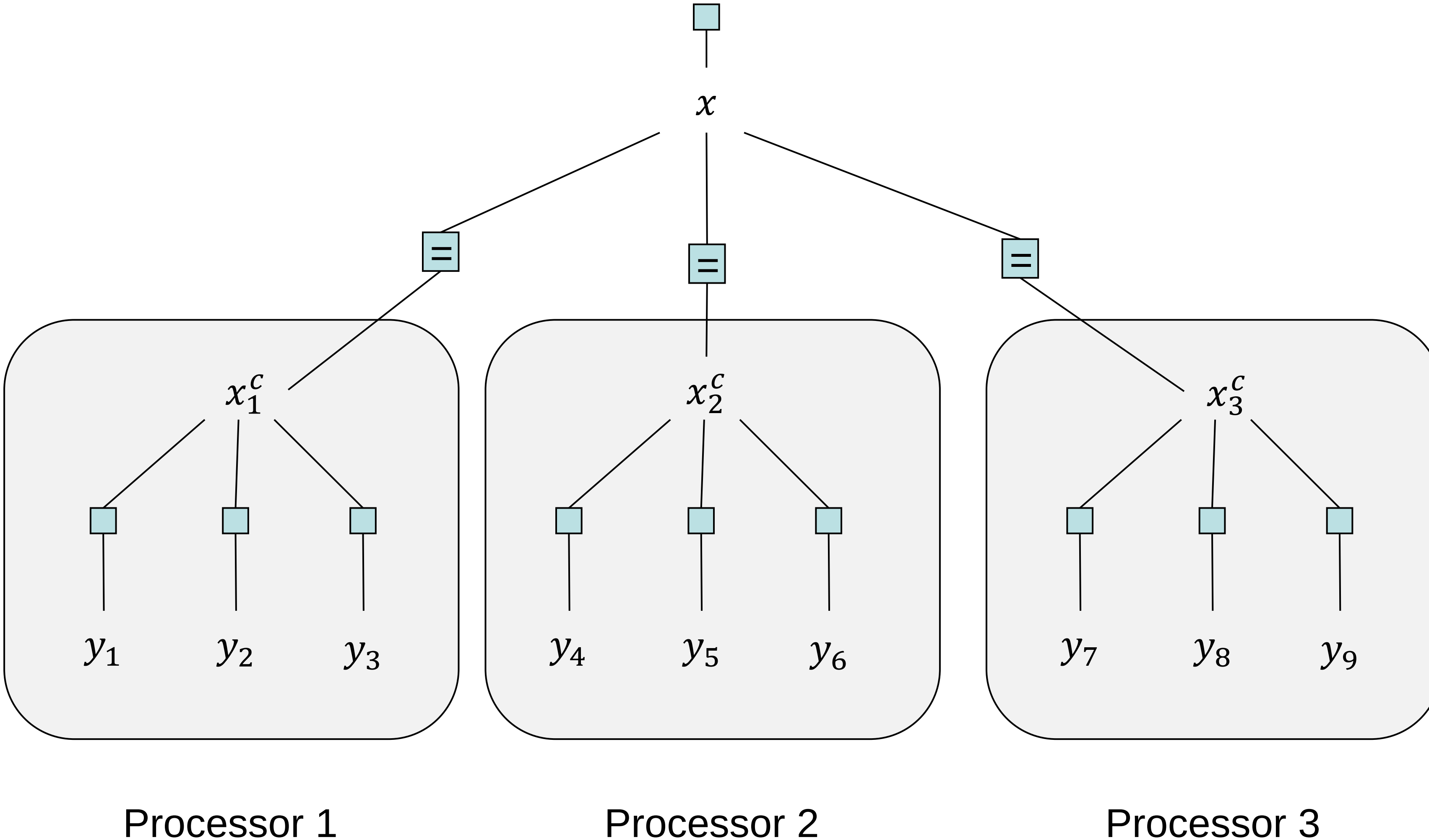
200 points

Deterministic methods improve with more data (posterior is more Gaussian)
Sampling methods do not

Parallel processing



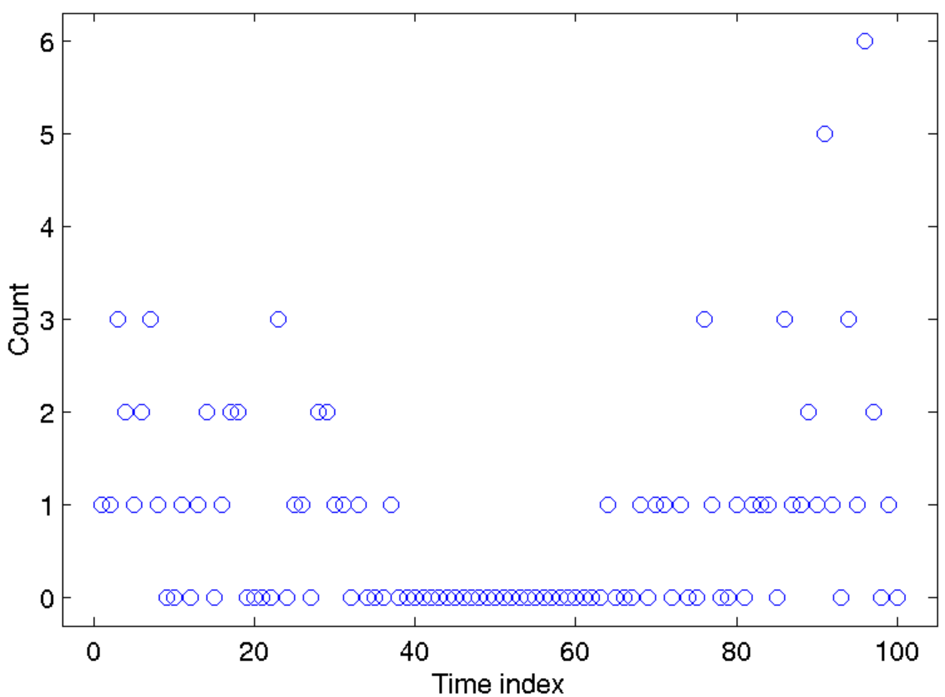
Parallel processing



Time series problems

Example: Poisson tracking

- y_t is a Poisson-distributed integer with mean $\exp(x_t)$



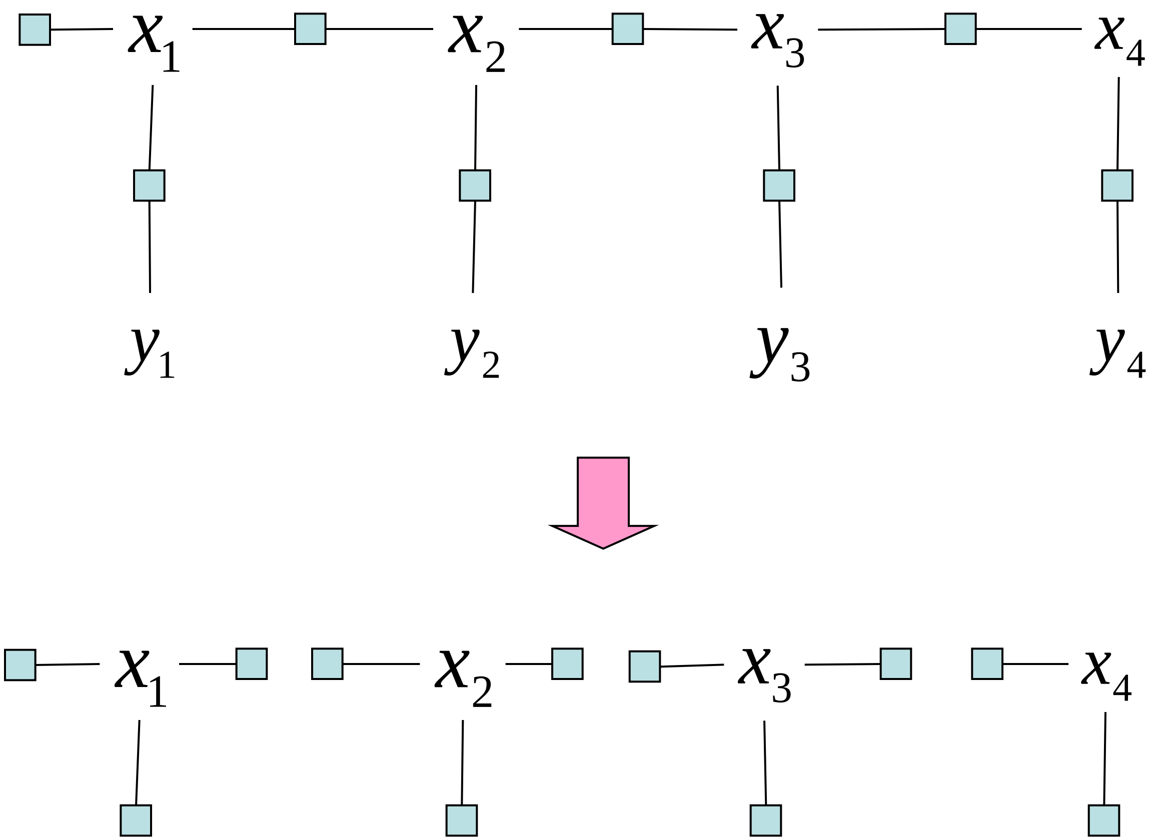
Poisson tracking model

$$p(x_1) \sim N(0{,}100)$$

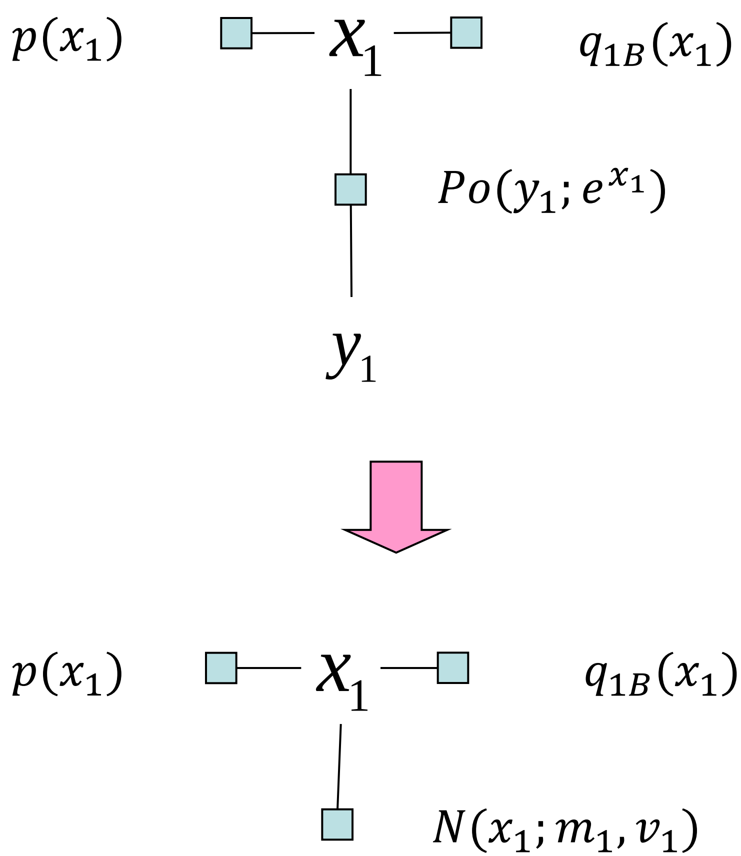
$$p(x_t \mid x_{t-1}) \sim N(x_{t-1}, 0.01)$$

$$p(y_t \mid x_t) = \exp(y_t x_t - e^{x_t}) / y_t!$$

Factor graph

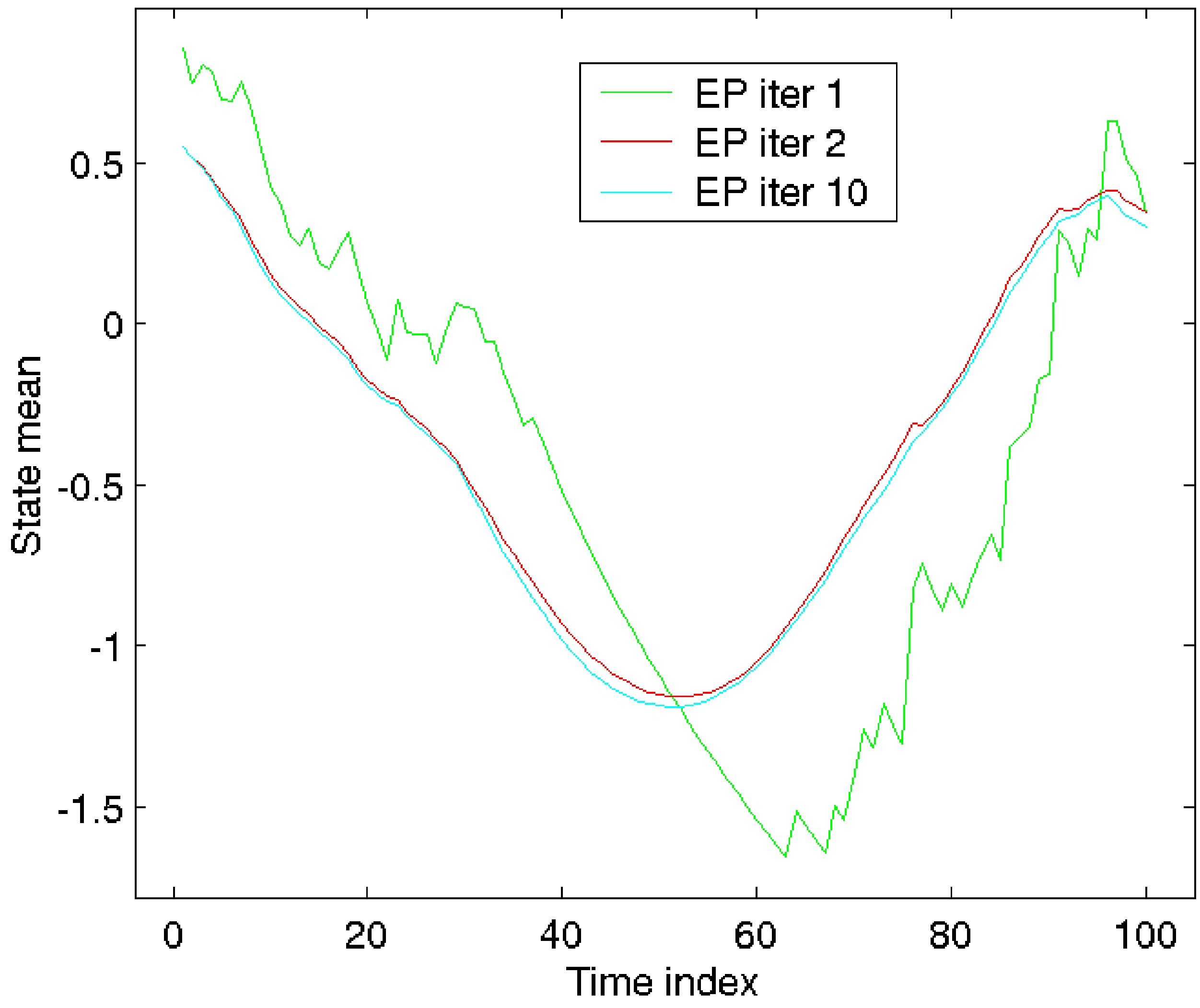


Approximating a measurement factor

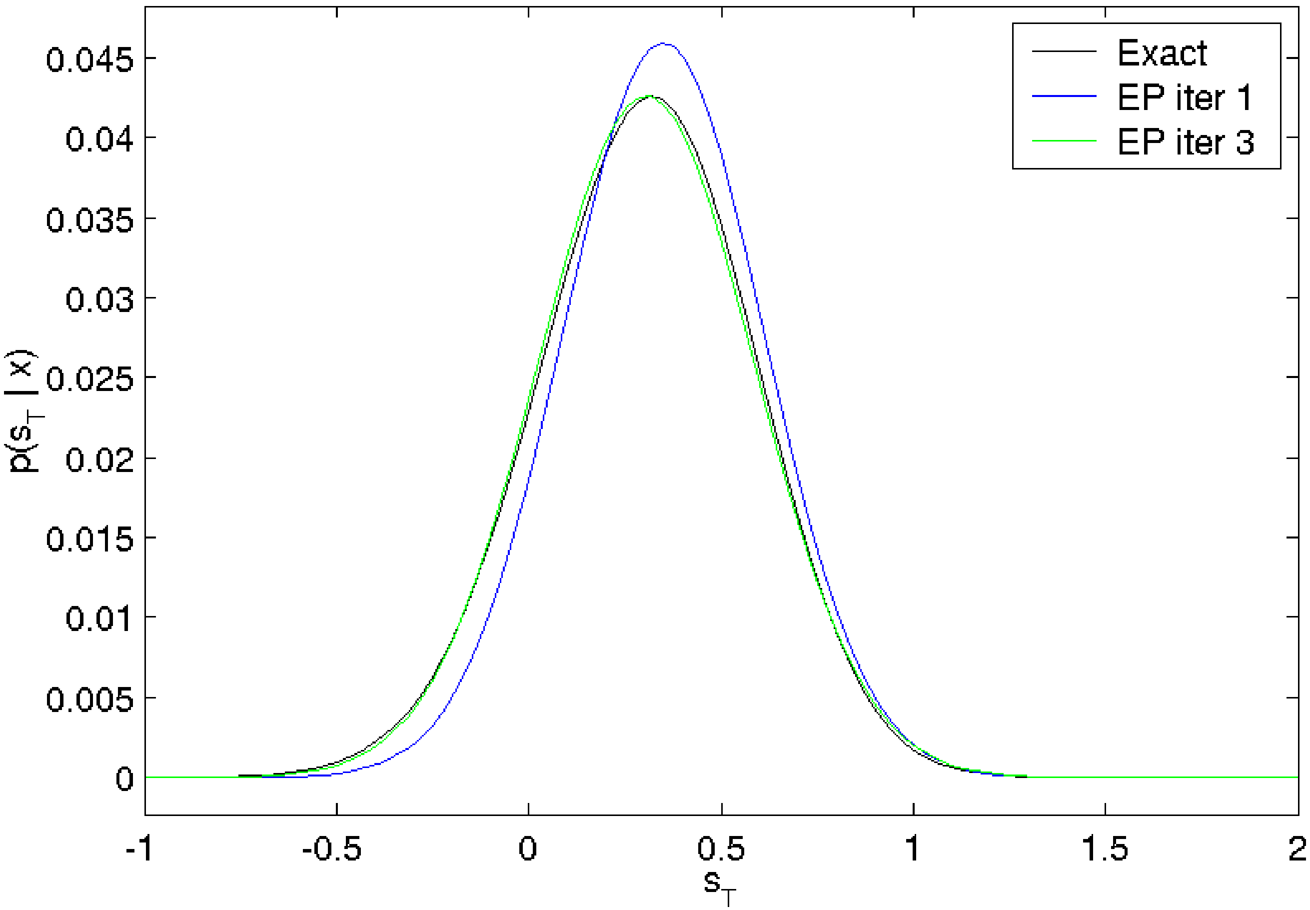


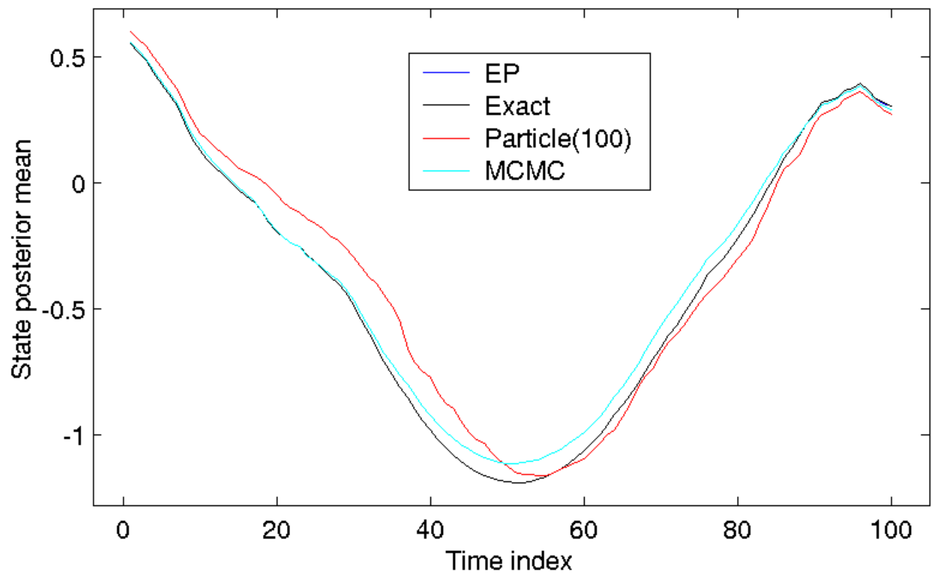
$$N(x_1; m_1, v_1) = \frac{\mathrm{proj}[Po(y_1; e^{x_1}) q_{1B}(x_1) p(x_1)]}{q_{1B}(x_1) p(x_1)}$$

Implementation: 1D quadrature

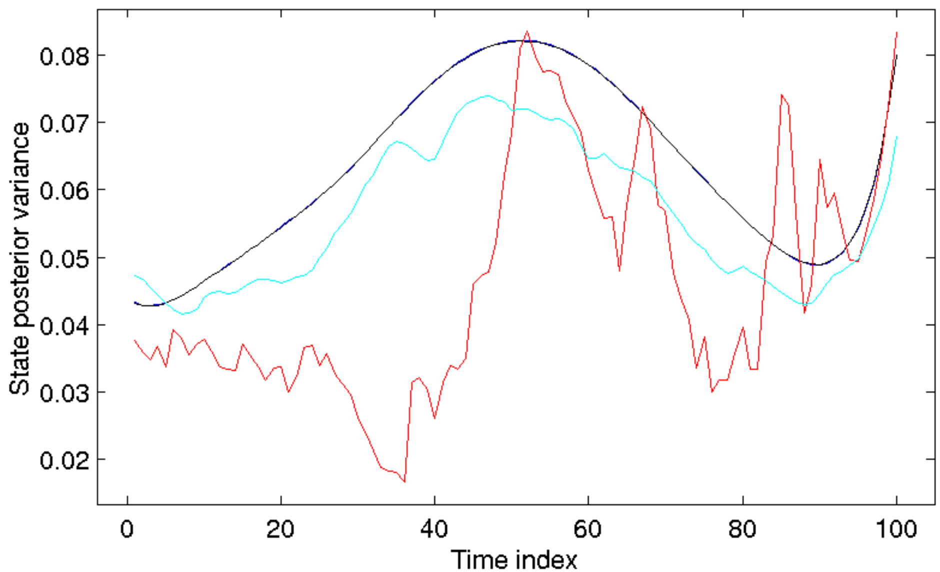

EP iter 1
EP iter 2
EP iter 10
0.5
0
-0.5
-1
-1.5
State mean
0
20
40
60
80
100
Time index

Posterior for the last state

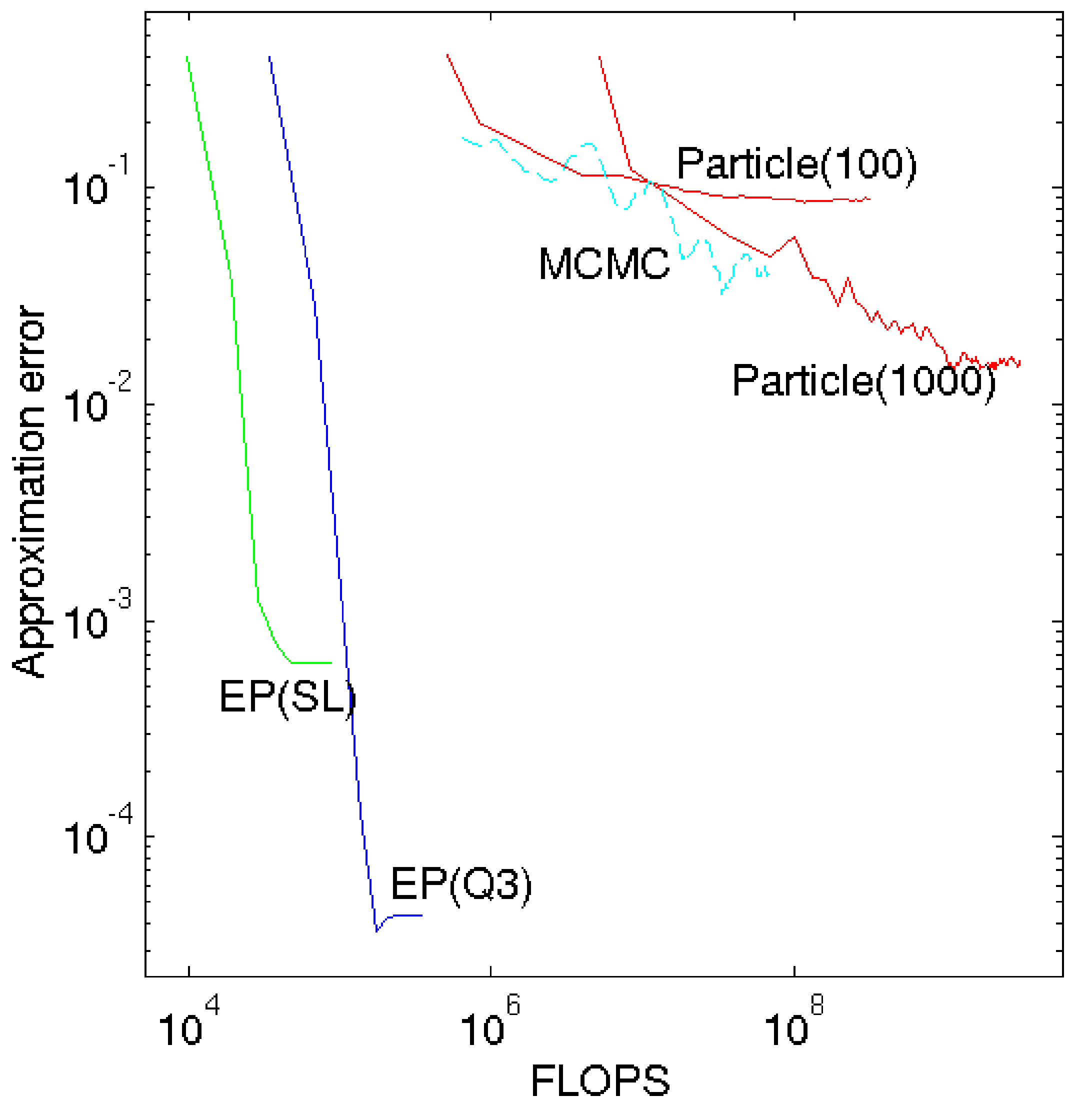




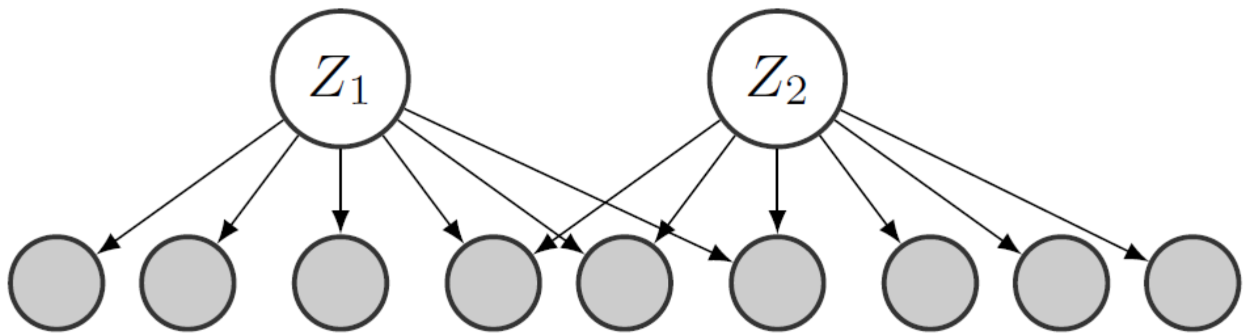
EP
Exact
Particle(100)
MCMC
State posterior mean
0.5
0
-0.5
-1
0
20
40
60
80
100
Time index



State posterior variance
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0
20
40
60
80
100
Time index


Approximation error
10^-1
10^-2
10^-3
10^-4
10^4
10^6
10^8
FLOPS
Particle(100)
MCMC
Particle(1000)
EP(SL)
EP(Q3)

Learning graph structure



Goals:

1. Learn latent variables (Z) that explain observed data
2. Learn sparse connectivity between Z and observed variables

"Structural Expectation Propagation", Lazic et al, AISTATS 2013

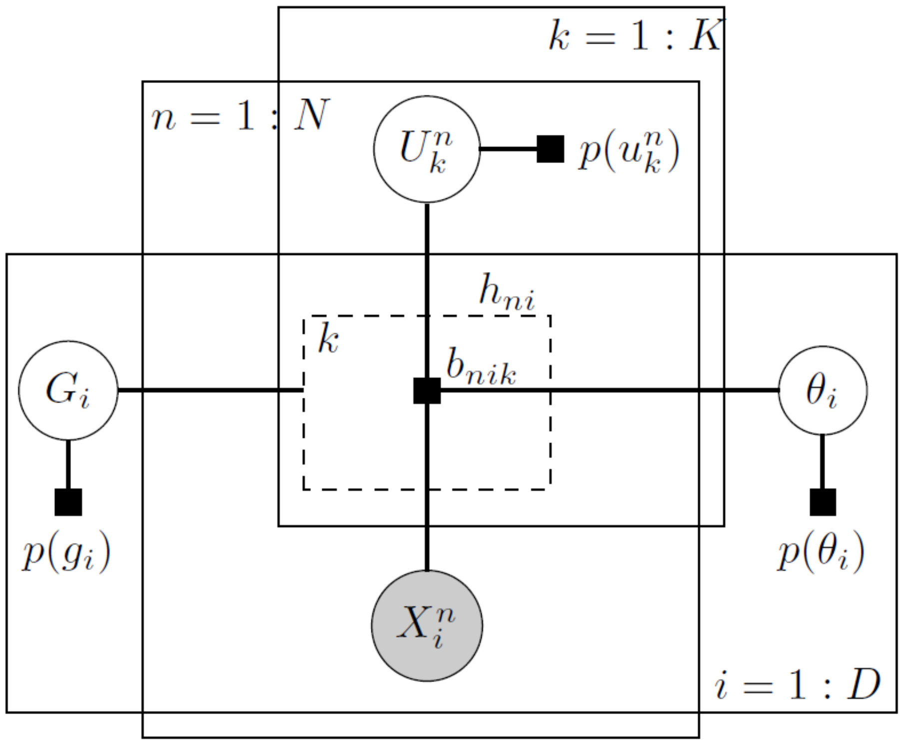
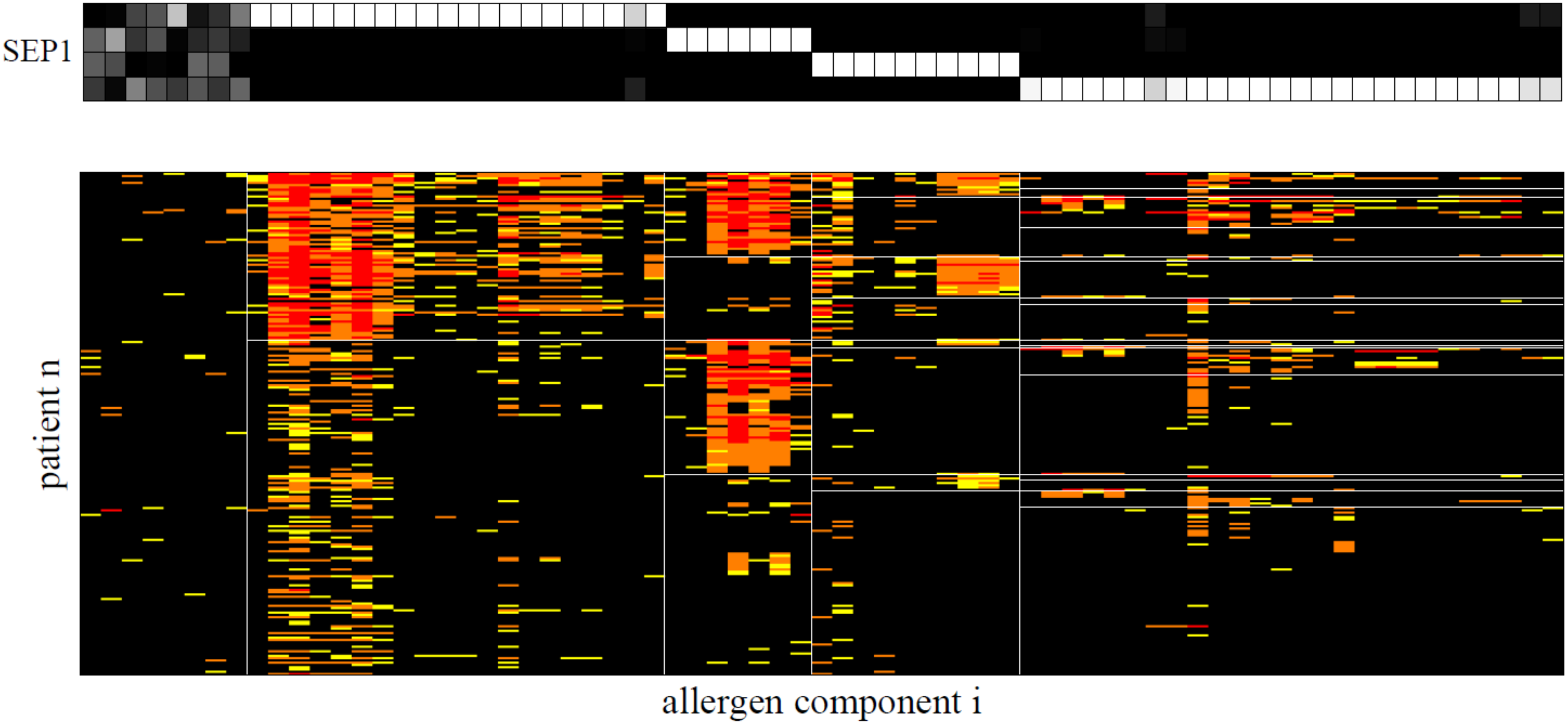


Figure 2: Gated factor graph representing a bipartite network in which each observed variable X_i has a single latent parent U_k, and the parent is indexed by G_i.

Lazic et al, AISTATS 2013

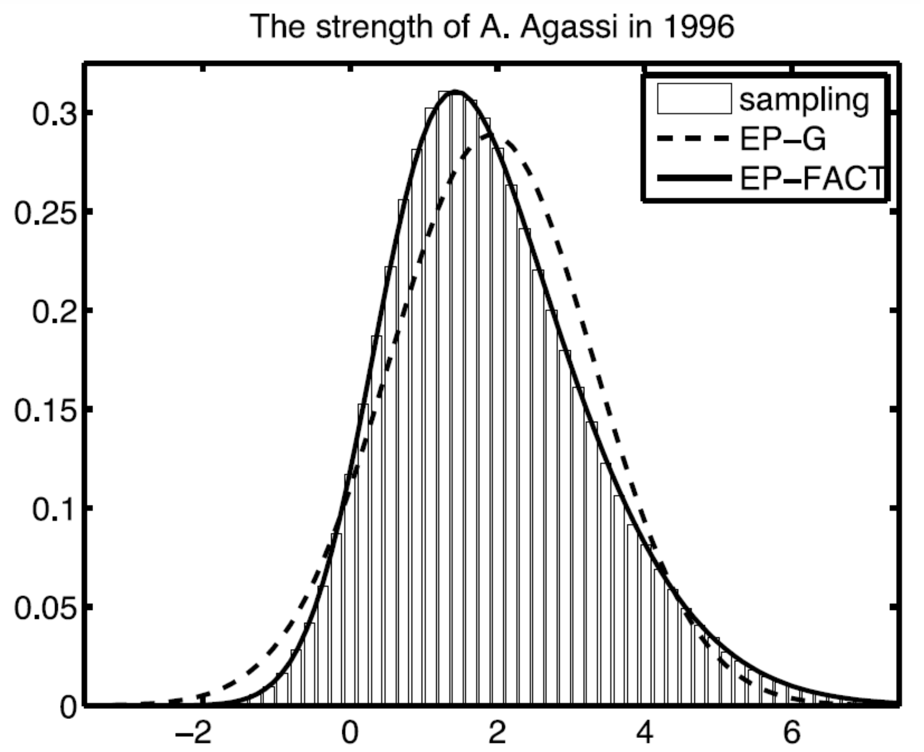
Results



Lazic et al, AISTATS 2013

Enhancing EP

Improving accuracy by conditioning



- Run EP multiple times with different values of a variable, then interpolate the results
- Can be done efficiently by exploiting previous factor approximations

Cseke and Heskes, JMLR 2011

Learning to initialize message passing

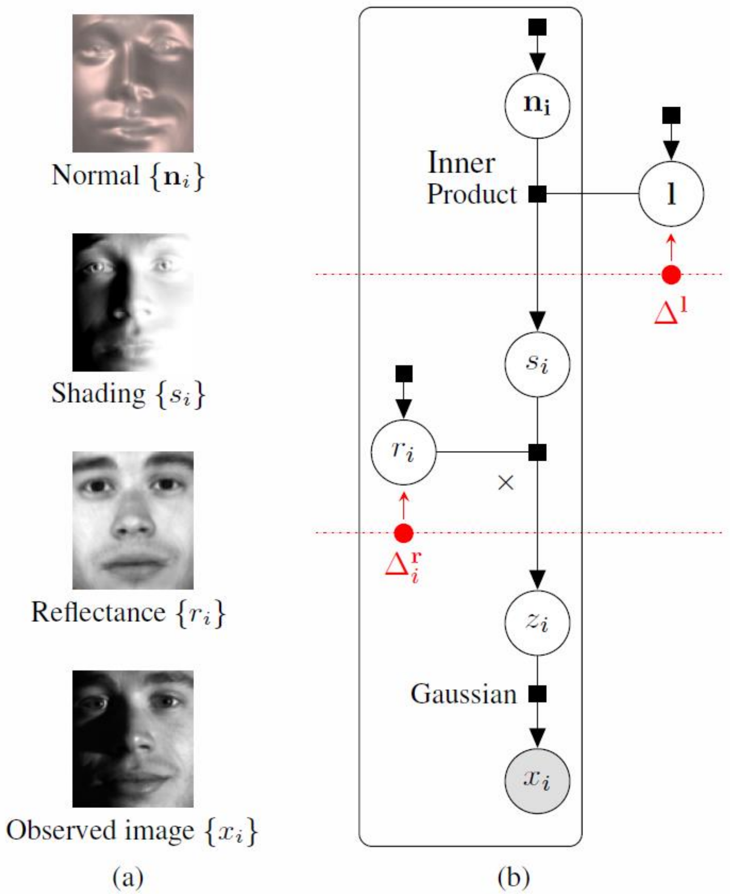
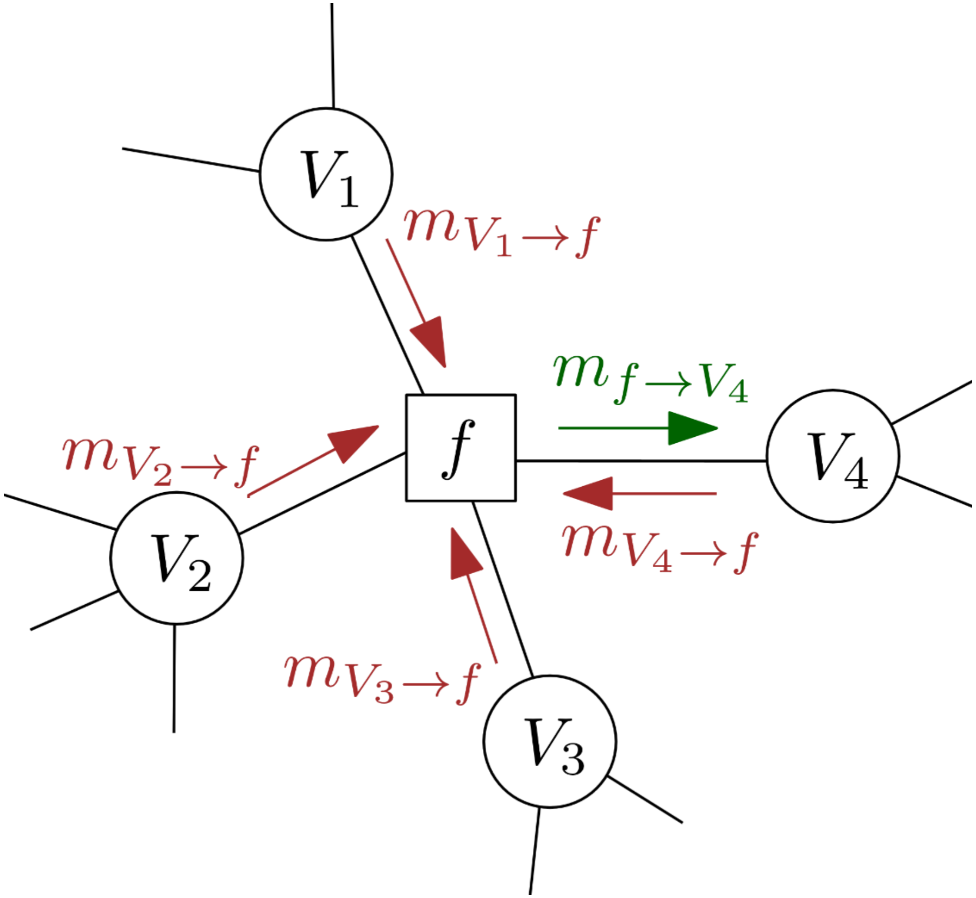


Figure 6: **The face problem.** (a) We observe an image and wish to infer the corresponding reflectance map and normal map (visualized here as 3D shape). (b) A graphical model for this problem. Symmetry priors not shown.

1. Sample from model
2. Train a regressor to predict r from x
3. Train a regressor to predict L from s
4. Given new image, do one upward sweep using the regressors
5. Run message-passing from this starting point

Jampani et al, AISTATS 2015

Learning to pass messages



1. Start with a slow implementation of EP
2. Collect (input,output) pairs of messages at a factor
3. Train a regressor to predict the output message
4. For inputs where the regressor is confident, use its output instead

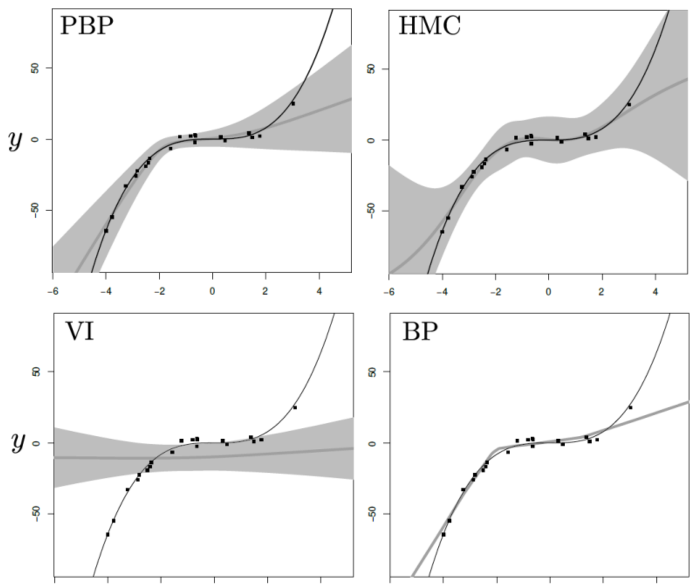
Eslami et al, NIPS 2014
Jitkrittum et al, UAI 2015

Benefits of learned messages

- Amortizes cost of divergence minimization
- Large subgraphs, e.g. cycles, can be processed in one step
- Can use unusual divergence measures
- Can pass non-Gaussian (non-Exp Family) messages

Probabilistic Backpropagation

- EP with fast approximate divergence minimization
- Allows Bayesian learning of multilayer neural nets



Hernandez-Lobato and Adams, ICML 2015

Further reading

- Divergence measures and message passing
  http://research.microsoft.com/~minka/papers/message-passing/
- EP bibliography
  http://research.microsoft.com/~minka/papers/ep/roadmap.html
- Infer.NET software
  http://research.microsoft.com/infernet