Expectation Propagation in Practice

Tom Minka
CMU Statistics
Joint work with Yuan Qi and John Lafferty

Outline

- EP algorithm
- Examples:
  - Tracking a dynamic system
  - Signal detection in fading channels
  - Document modeling
  - Boltzmann machines

Extensions to EP

- Alternatives to moment-matching
- Factors raised to powers
- Skipping factors

EP in a nutshell

- Approximate a function by a simpler one:

$$p(\mathbf{x}) = \prod_a f_a(\mathbf{x}) \quad \longrightarrow \quad q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x})$$

- Where each $\tilde{f}_a(\mathbf{x})$ lives in a parametric, exponential family (e.g. Gaussian)
- Factors $f_a(\mathbf{x})$ can be conditional distributions in a Bayesian network

EP algorithm

- Iterate the fixed-point equations:

$$\tilde{f}_a(\mathbf{x}) = \arg\min D(f_a(\mathbf{x})q^{\backslash a}(\mathbf{x}) \,\|\, \tilde{f}_a(\mathbf{x})q^{\backslash a}(\mathbf{x}))$$

where $q^{\backslash a}(\mathbf{x}) = \prod_{b \neq a} \tilde{f}_b(\mathbf{x})$

- $q^{\backslash a}(\mathbf{x})$ specifies where the approximation needs to be good
- Coordinated local approximations

(Loopy) Belief propagation

- Specialize to factorized approximations:

$$\tilde{f}_a(\mathbf{x}) = \prod_i \tilde{f}_{ai}(x_i)$$ "messages"

- Minimize KL-divergence = match marginals of $f_a(\mathbf{x})q^{\backslash a}(\mathbf{x})$ (partially factorized) and $\tilde{f}_a(\mathbf{x})q^{\backslash a}(\mathbf{x})$ (fully factorized)
  - "send messages"

EP versus BP

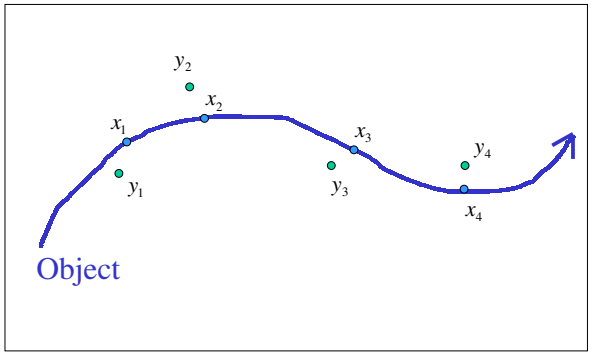
- EP approximation can be in a restricted family, e.g. Gaussian
- EP approximation does not have to be factorized
- EP applies to many more problems
  - e.g. mixture of discrete/continuous variables

EP versus Monte Carlo

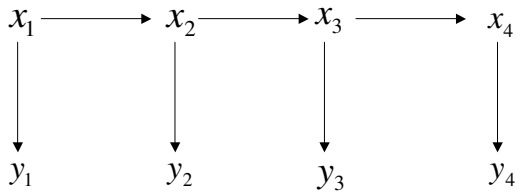
- Monte Carlo is general but expensive
  - A sledgehammer
- EP exploits underlying simplicity of the problem (if it exists)
- Monte Carlo is still needed for complex problems (e.g. large isolated peaks)
- Trick is to know what problem you have

Example: Tracking

Guess the position of an object given noisy measurements



Bayesian network



e.g. $x_t = x_{t-1} + v_t$ (random walk)

$y_t = x_t + \text{noise}$

want distribution of x's given y's

Terminology

- Filtering: posterior for last state only
- Smoothing: posterior for middle states
- On-line: old data is discarded (fixed memory)
- Off-line: old data is re-used (unbounded memory)

Kalman filtering / Belief propagation

- Prediction:

$$p(x_t \mid y_{<t}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{<t}) dx_{t-1}$$

- Measurement:

$$p(x_t \mid y_{<t}, y_t) \propto p(y_t \mid x_t) p(x_t \mid y_{<t})$$

- Smoothing:

$$p(x_t \mid y_{\le t}, y_{>t}) \propto p(x_t \mid y_{\le t}) \int p(x_{t+1} \mid x_t) p(x_{t+1} \mid y_{\le t+1}, y_{>t+1}) dx_{t+1}$$

Approximation

$$p(\mathbf{x},\mathbf{y}) = p(x_1)\,p(y_1 \mid x_1)\prod_{t>1} p(x_t \mid x_{t-1})\,p(y_t \mid x_t)$$

$$q(\mathbf{x}) = p(x_1)\tilde{o}_1(x_1)\prod_{t>1} \tilde{p}_{t-1\to t}(x_t)\,\tilde{p}_{t\to t-1}(x_{t-1})\,\tilde{o}_t(x_t)$$

Factorized and Gaussian in x

Approximation

$$q(x_t) = \tilde{p}_{t-1\to t}(x_t)\,\tilde{o}(x_t)\,\tilde{p}_{t+1\to t}(x_t)$$

= (forward msg)(observation)(backward msg)

EP equations are exactly the prediction, measurement, and smoothing equations for the Kalman filter
- but only preserve first and second moments

Consider case of linear dynamics…

EP in dynamic systems

- Loop t = 1, …, T (filtering)
  - Prediction step
  - Approximate measurement step
- Loop t = T, …, 1 (smoothing)
  - Smoothing step
  - Divide out the approximate measurement
  - Re-approximate the measurement
- Loop t = 1, …, T (re-filtering)
  - Prediction and measurement using previous approx

Generalization

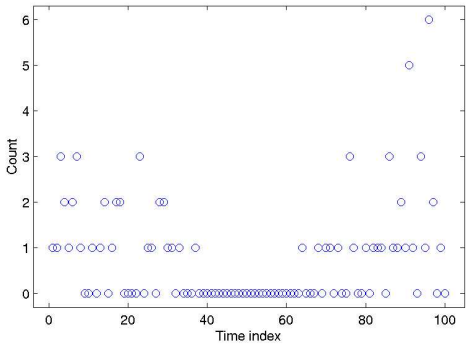
- Instead of matching moments, can use any method for approximate filtering
- E.g. Extended Kalman filter, statistical linearization, unscented filter, etc.
- All can be interpreted as finding linear/Gaussian approx to original terms

Interpreting EP

- After more information is available, re-approximate individual terms for better results
- Optimal filtering is no longer on-line

Example: Poisson tracking

- y_t is an integer valued Poisson variate with mean $\exp(x_t)$



Poisson tracking model

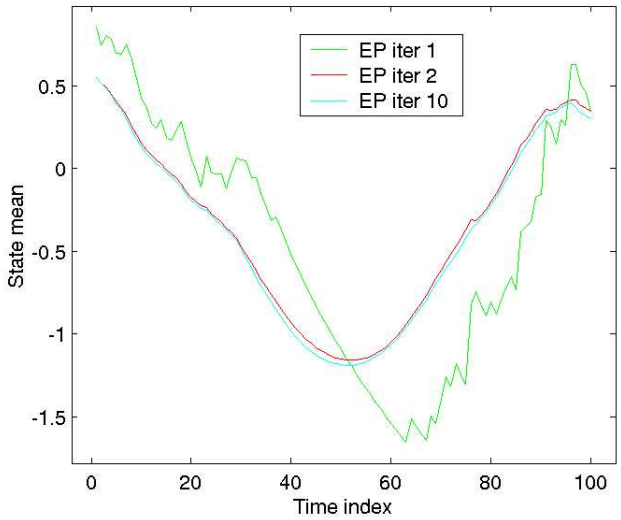
$$p(x_1) \sim N(0,100)$$

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

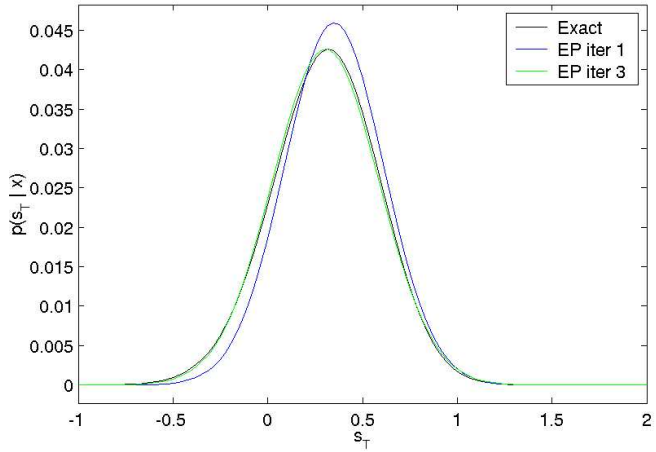
$$p(y_t \mid x_t) = \exp(y_t x_t - e^{x_t}) / y_t!$$

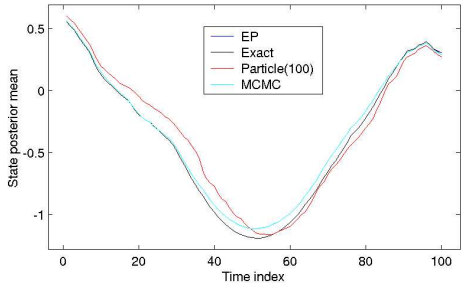
Approximate measurement step

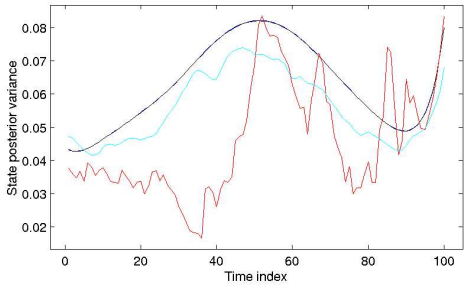
- $p(y_t \mid x_t)p(x_t \mid y_{<t})$ is not Gaussian
- Moments of x not analytic
- Two approaches:
  - Gauss-Hermite quadrature for moments
  - Statistical linearization instead of moment-matching
- Both work well

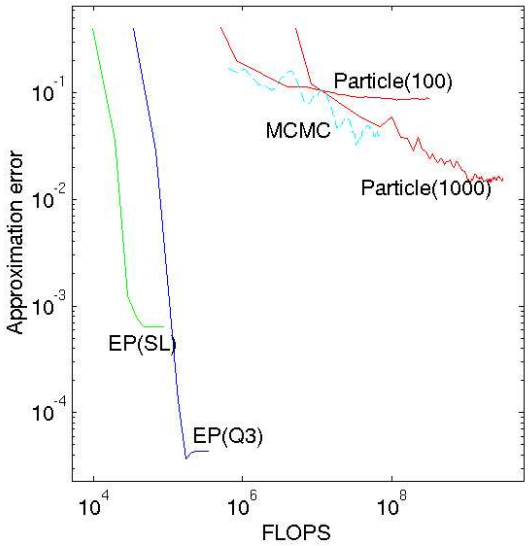


Posterior for the last state



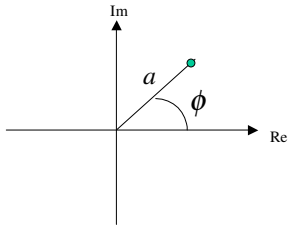






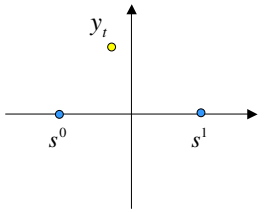
EP for signal detection

- Wireless communication problem
- Transmitted signal = $a\sin(\omega t+\phi)$
- (a,ϕ) vary to encode each symbol
- In complex numbers: $ae^{i\phi}$



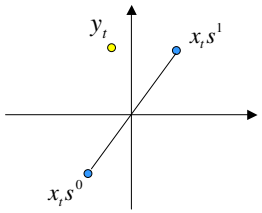
Binary symbols, Gaussian noise

- Symbols are 1 and –1 (in complex plane)
- Received signal = $a\sin(\omega t+\phi)+\text{noise}$
- Recovered $\hat{a}e^{\hat{\phi}} = ae^{\phi}+\text{noise} = y_t$
- Optimal detection is easy



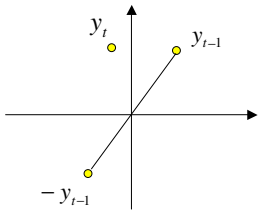
Fading channel

- Channel systematically changes amplitude and phase:
$$y_t = x_t s+\text{noise}$$
- x_t changes over time

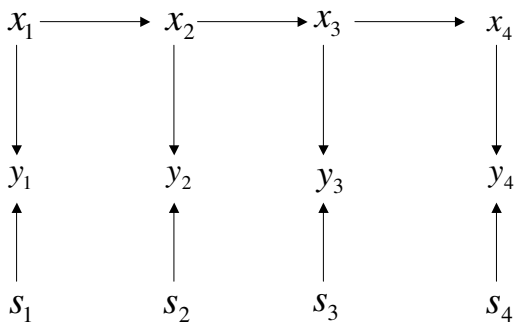


Differential detection

- Use last measurement to estimate state
- Binary symbols only
- No smoothing of state = noisy



Bayesian network



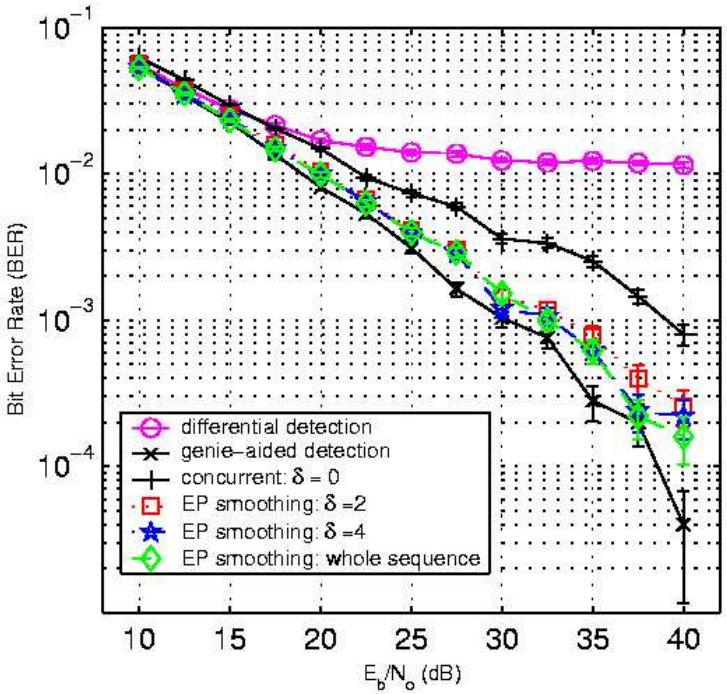
Symbols can also be correlated (e.g. error-correcting code)

Dynamics are learned from training data (all 1's)

On-line implementation

- Iterate over the last δ measurements
- Previous measurements act as prior

- Results comparable to particle filtering, but much faster



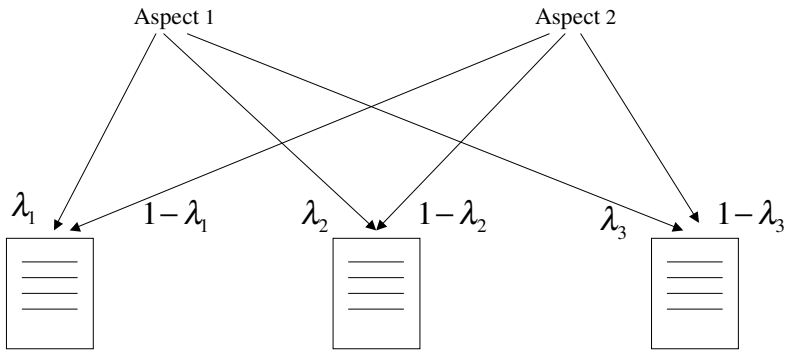
Document modeling

- Want to classify documents by semantic content
- Word order generally found to be irrelevant
  - Word *choice* is what matters
- Model each document as a bag of words
  - Reduces to modeling correlations between word probabilities

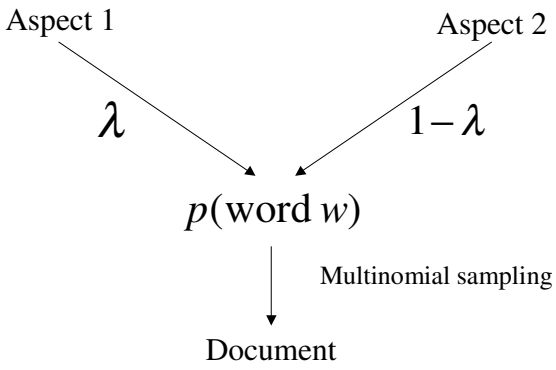
Generative aspect model

(Hofmann 1999; Blei, Ng, & Jordan 2001)

Each document mixes aspects in different proportions



Generative aspect model



$p(\lambda) \sim Dirichlet(\alpha_1,...,\alpha_J)$

Two tasks

Inference:

- Given aspects and document i, what is (posterior for) λ_i?

Learning:

- Given some documents, what are (maximum likelihood) aspects?

Approximation

- Likelihood is composed of terms of form

$$t_w(\lambda)^{n_w} = p(w)^{n_w} = (\sum_a \lambda_a p(w \mid a))^{n_w}$$

- Want Dirichlet approximation:

$$\tilde{t}_w(\lambda) = \prod_a \lambda_a^{\beta_{wa}}$$

EP with powers

- These terms seem too complicated for EP
- Can match moments if $n_w = 1$, but not for large n_w
- Solution: match moments of one occurrance at a time
  - Redefine what are the "terms"

EP with powers

- Moment match:

$$t_w(\lambda) q^{\setminus w}(\lambda) \leftrightarrow \tilde{t}_w(\lambda) q^{\setminus w}(\lambda)$$

- Context function: all but one occurrence

$$q^{\setminus w}(\lambda) = \tilde{t}_w(\lambda)^{n_w - 1} \prod_{w' \neq w} \tilde{t}_{w'}(\lambda)^{n_{w'}}$$

- Fixed point equations for β

EP with skipping

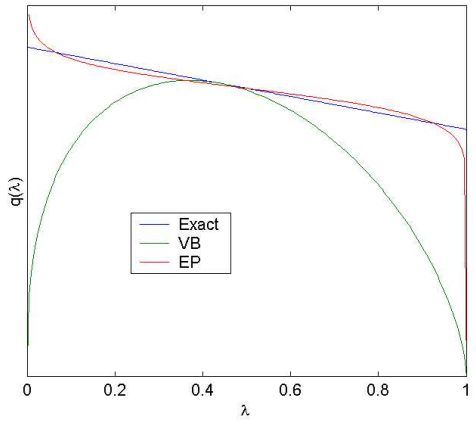
- Context fcn might not be a proper density
- Solution: "skip" this term
  - (keep old approximation)
- In later iterations, context becomes proper

Another problem

- Minimizing KL-divergence of Dirichlet is expensive
  - Requires iteration
- Match (mean,variance) instead
  - Closed-form

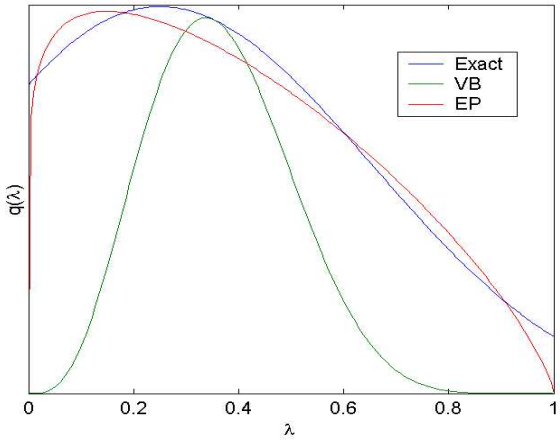
One term

$$t_w(\lambda) = (\lambda)0.4 + (1-\lambda)0.3$$



VB = Variational Bayes (Blei et al)

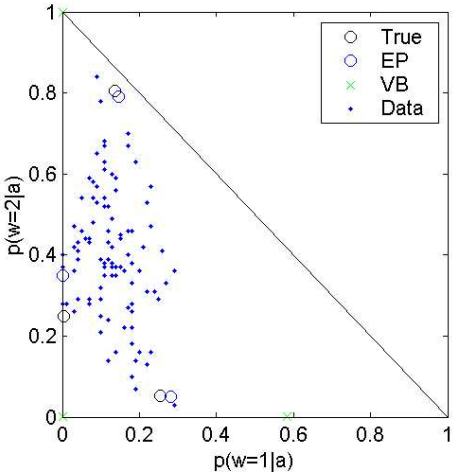
Ten word document



General behavior

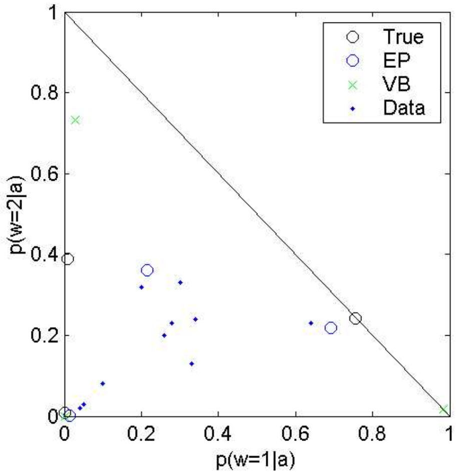
- For long documents, VB recovers correct mean, but not correct variance of λ
- Disastrous for learning
  - No Occam factor
- Gets *worse* with more documents
  - No asymptotic salvation
- EP gets correct variance, learns properly

Learning in probability simplex



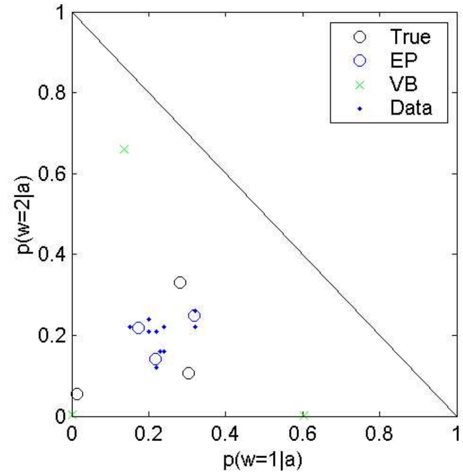
100 docs,
Length 10

Learning in probability simplex



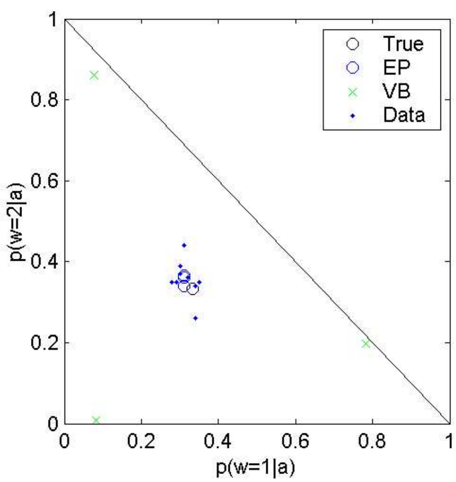
10 docs,
Length 10

Learning in probability simplex



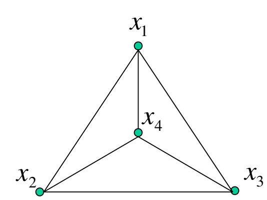
10 docs,
Length 10

Learning in probability simplex



10 docs,
Length 10

Boltzmann machines

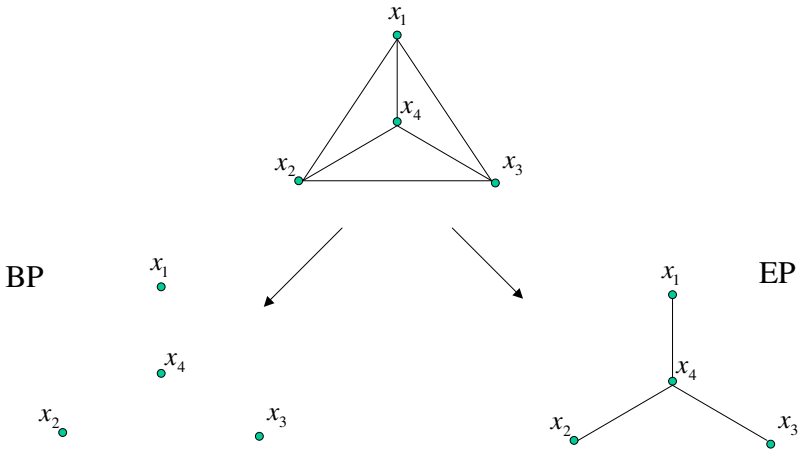


Joint distribution is product of pair potentials:

$$p(\mathbf{x}) = \prod_a f_a(\mathbf{x}) \quad \longrightarrow \quad q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x})$$

Want to approximate by a simpler distribution

Approximations



Approximating an edge by a tree

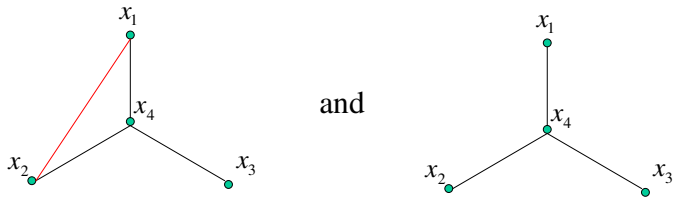
Each potential in p is projected onto the tree-structure of q

$$f_a(x_1, x_2) \approx \frac{\tilde{f}_a^{14}(x_1, x_4)\tilde{f}_a^{24}(x_2, x_4)\tilde{f}_a^{34}(x_3, x_4)}{\tilde{f}_a^{4}(x_4)^2}$$

Correlations are not lost, but projected onto the tree

Fixed-point equations

- Match single and pairwise marginals of



- Reduces to exact inference on single loops
  - Use cutset conditioning

5-node complete graphs, 10 trials

Method	FLOPS	Error
Exact	500	0
TreeEP	3,000	0.032
BP/double-loop	200,000	0.186
GBP	360,000	0.211

8x8 grids, 10 trials

Method	FLOPS	Error
Exact	30,000	0
TreeEP	300,000	0.149
BP/double-loop	15,500,000	0.358
GBP	17,500,000	0.003

TreeEP versus BP

- TreeEP always more accurate than BP, often faster
- GBP slower than BP, not always more accurate
- TreeEP converges more often than BP and GBP

Conclusions

- EP algorithms exceed state-of-art in several domains
- Many more opportunities out there
- EP is sensitive to choice of approximation
  - does not give guidance in choosing it (e.g. tree structure) – error bound?
- Exponential family constraint can be limiting – mixtures?

End