Tree-structured approximations by expectation propagation

Thomas Minka and Yuan Qi

Overview

- Each factor is approximated by a tree
- More accurate than loopy belief propagation, for small extra cost
- Analogous to structured mean-field (but cheaper, more accurate)
- Behaves differently than clustering (GBP)

EP vs mean-field

- Both approximate complex distribution (p) with simpler distribution (q)
- Mean-field minimizes `exclusive' KL-divergence: $\min_q KL(q \| p)$
- EP minimizes `inclusive' KL-divergence:
$$\min_q KL(p \| q)$$
- Inclusive gives more accurate expectations

Related work

- Structured mean-field
  – (Ghahramani & Jordan, 1997) (Wiegerinck, 2000)
- Tree-structured upper bounds
  – (Wainwright et al, 2002)
- Tree-based scheduling for BP
  – (Wainwright et al, 2001)
- Tree-structured assumed-density filtering
  – (Frey et al, 2000)
- Expectation propagation
  – (Minka, 2001)

EP in a nutshell

- Approximate a function by a simpler one:

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

- Where each $\tilde{f}_a(\mathbf{x})$ lives in tractable family
- Factors $f_a(\mathbf{x})$ can be conditional distributions in a Bayesian network, or potentials in Markov network

EP algorithm

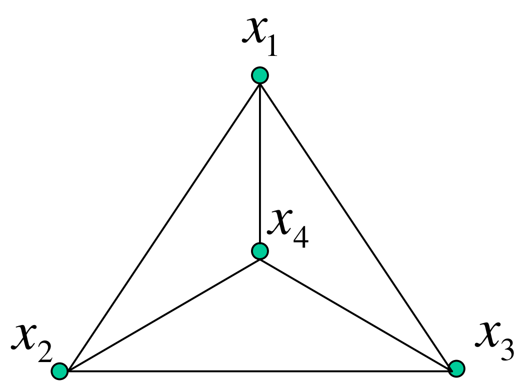
- Iterate the fixed-point equations:

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

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

- Coordinated local approximations

Boltzmann machines

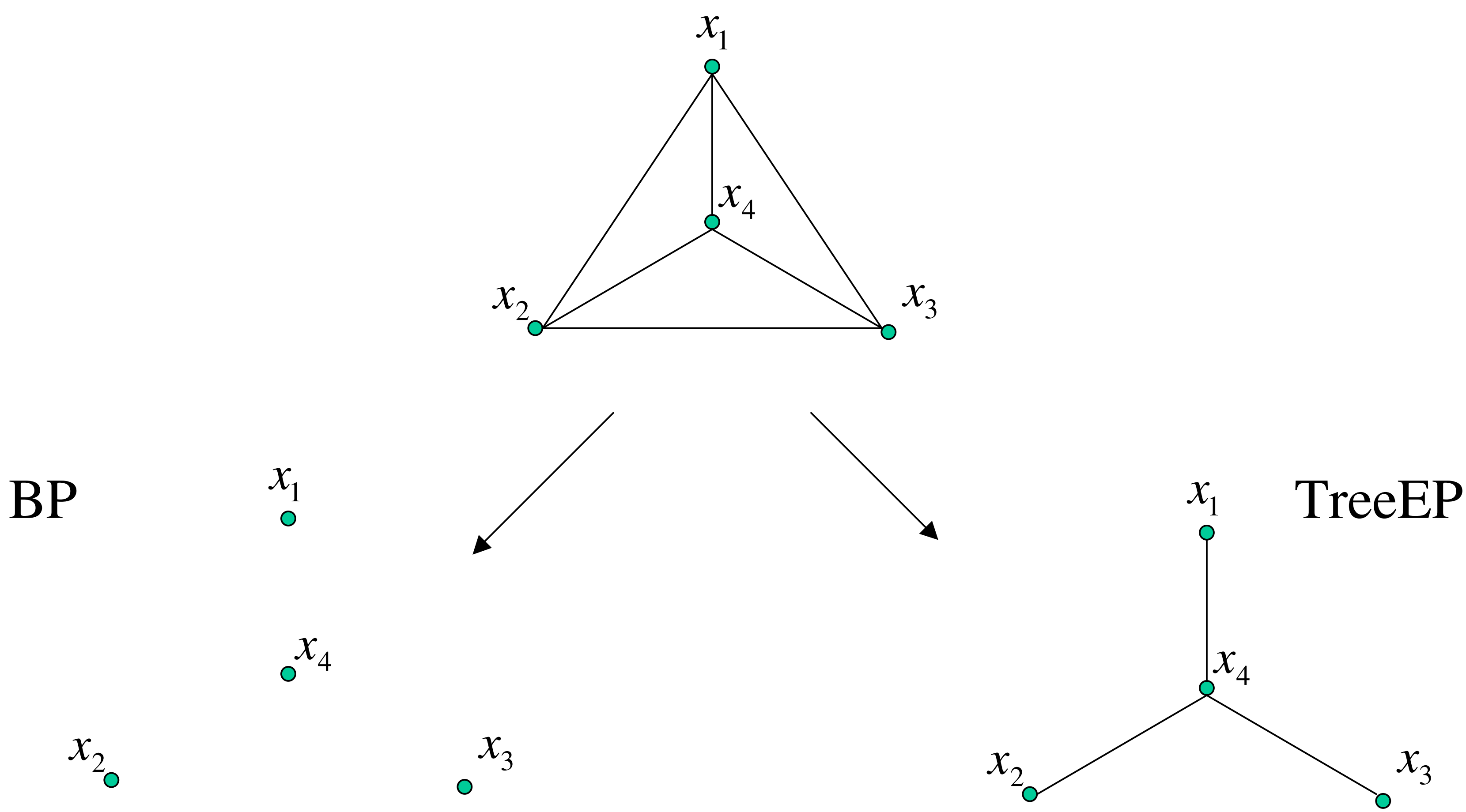


Joint distribution is product of pair potentials:

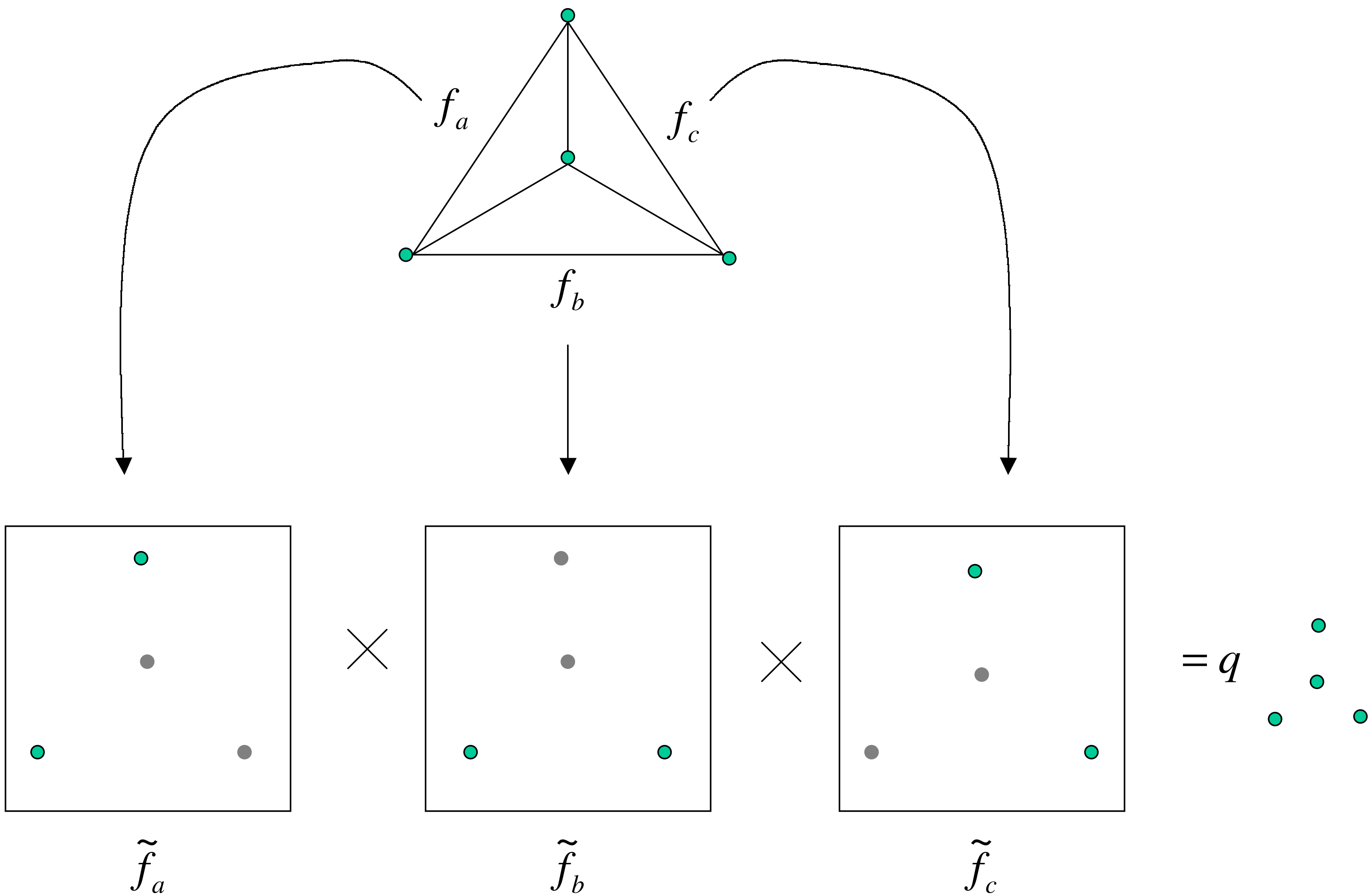
$$p(\mathbf{x}) = \prod_a f_a(\mathbf{x}) \longrightarrow q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x})$$

Want to approximate by a simpler distribution

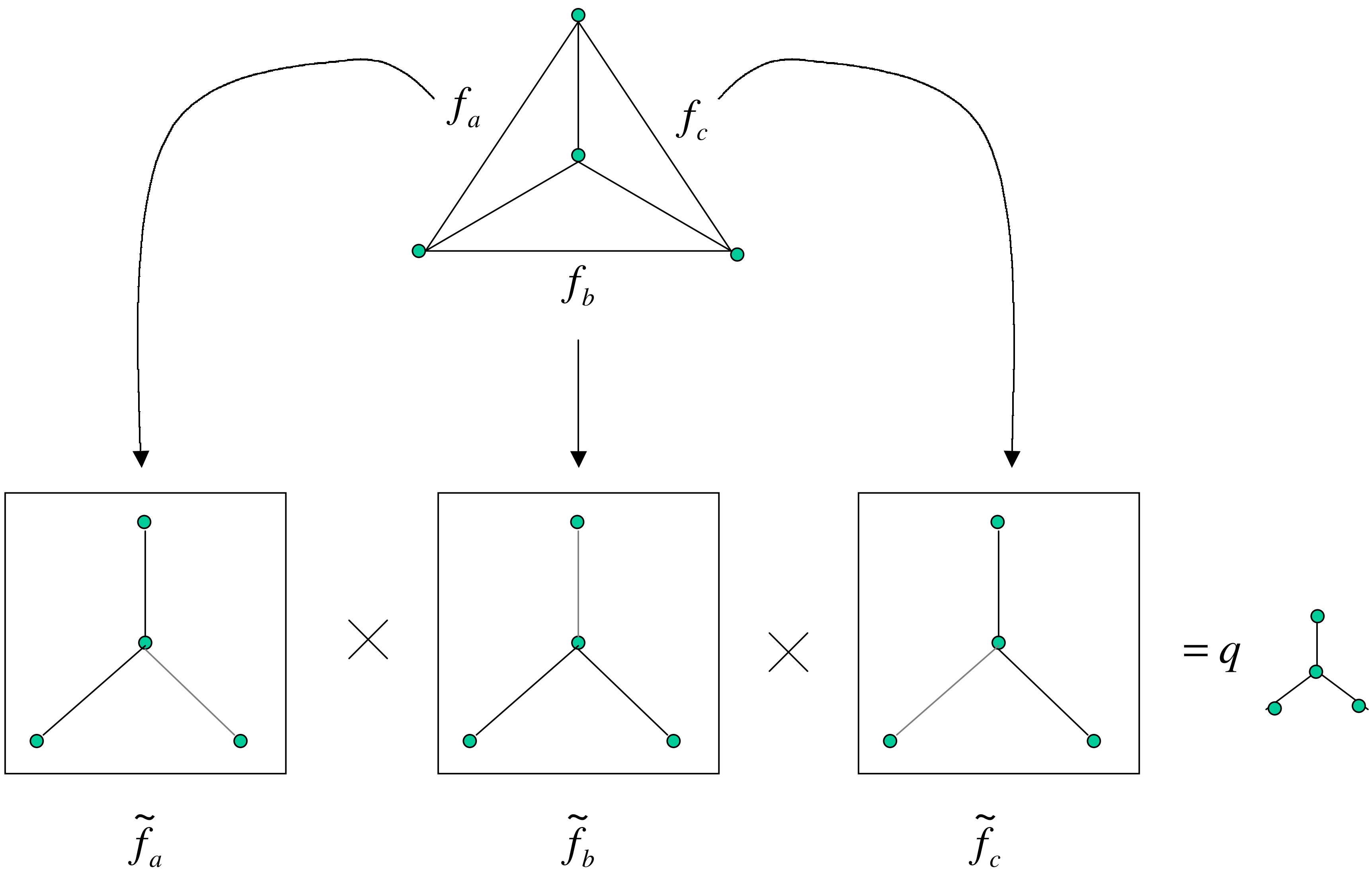
Approximations



BP



TreeEP



Approximations

- BP (= factorized EP)

$$q(x) = \prod_i q(x_i)$$

- TreeEP

$$q(x) = \frac{\prod_{(j,k)\in T} q(x_j, x_k)}{\prod_{s\in S} q(x_s)}$$

Approximating an edge by a tree

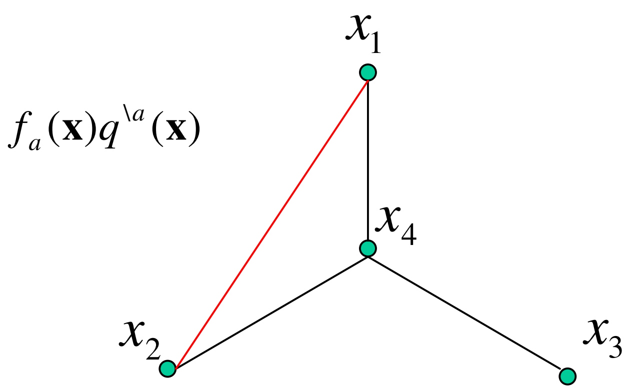
Each potential in p is projected onto the tree-structure of q

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

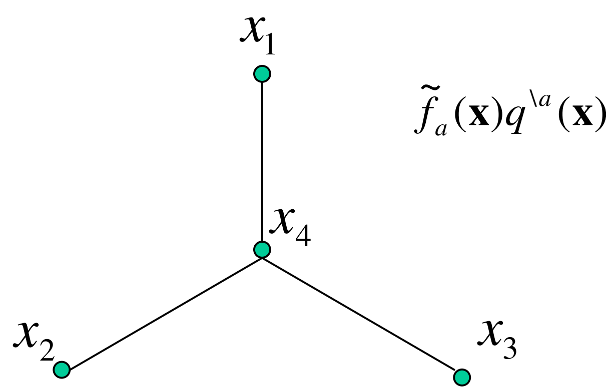
Correlations are not lost, but projected onto the tree

Fixed-point equations

- Match single and pairwise marginals of



and



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

Full algorithm

- Loop off-tree edges a
- Deletion: divide $q(\mathbf{x}) / \tilde{f}_a(\mathbf{x})$ to get $q^{\backslash a}(\mathbf{x})$

$$q^{\backslash a}(x_j, x_k) = \frac{q(x_j, x_k)}{\tilde{f}_a(x_j, x_k)} \qquad (j,k) \in T$$

- Incorporate evidence: exact inference on $f_a(\mathbf{x}) q^{\backslash a}(\mathbf{x})$ to get $q(\mathbf{x})$
- Update: divide $q(\mathbf{x}) / q^{\backslash a}(\mathbf{x})$ to get $\tilde{f}_a(\mathbf{x})$

$$\tilde{f}_a(x_j, x_k) = \frac{q(x_j, x_k)}{q^{\backslash a}(x_j, x_k)} \qquad (j,k) \in T$$

Choosing structure

- Spanning tree with maximum pairwise information (Chow & Liu)

$$I(x_j, x_k) = \sum_{x_j, x_k} p(x_j, x_k) \log \frac{p(x_j, x_k)}{p(x_j)\, p(x_k)}$$

- Pairwise marginals estimated by

$$p(x_j, x_k) \approx \prod_a f_a(x_j, x_k)$$

Experiments

- All algorithms implemented in Matlab using Bayes Net Toolbox
- Floating-point operations (FLOPS) counted via Lightspeed toolbox
- 5% rule: stop when error on all following iterations is within 5% of final error

Other algorithms

- TreeVB (Wiegerinck, 2000) with same tree structure as TreeEP, same junction tree optimizations
- BP used GBP code with no clusters (can also use TreeEP code with empty tree)
  - Probably not the most efficient implementation
  - Used largest step size that gave convergence on each network

Random potentials

- Single-node potentials:

$$f_a(x_j) = \begin{bmatrix} \exp(\theta_j) & \exp(-\theta_j) \end{bmatrix} \qquad \theta_j \sim N(0,1)$$

- Pairwise potentials:

$$f_a(x_j, x_k) = \begin{bmatrix} \exp(w_{jk}) & \exp(-w_{jk}) \\ \exp(-w_{jk}) & \exp(w_{jk}) \end{bmatrix} \qquad w_{jk} \sim N(0, J^2)$$

Generalized Belief Propagation

- A family of algorithms, depending on what clusters you choose
- For grids, clusters were 4-node loops
- For complete graphs, clusters were all 3-node loops
  - Probably not the best choice
- Used parent-child algorithm, with 0.5 damping, from Yedidia et al (2002)

Results

- TreeEP more accurate than BP, faster than TreeVB and GBP
- GBP with right clusters is best on grids
  - But extra edges can ruin its performance
- GBP with ` wrong' clusters can be worse than BP

Open questions

- What networks are best suited to TreeEP?
  - Probably not grids, complete graphs
  - Small tree-width?
- What is best way to choose structure?
  - Needed for TreeEP, TreeVB, GBP