

Message passing and expectation propagation

Christoph Dehner

Technische Universität München
Department of Informatics
Data Mining and Analytics
kdd.in.tum.de

29 May 2017

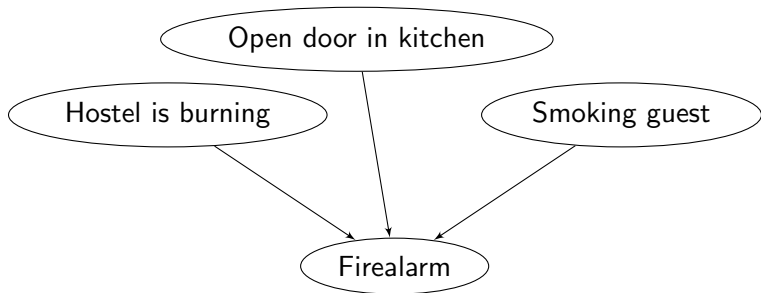
Graphical models: Bayesian networks and Markov random fields

$$p(X) = \prod_s f_s(X_s)$$

Inference in graphical models:

- Marginalization
- Maximum a posteriori estimation
- Posterior approximation

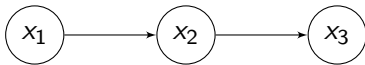
Motivation for marginalization



Factor graph

More simple and formally:

$$p(X) = p(x_1)p(x_2|x_1)p(x_3|x_1)$$



Factor graph

More simple and formally:

$$p(X) = p(x_1)p(x_2|x_1)p(x_3|x_1)$$



A corresponding factor graph: $f_1 = p(x_1)$, $f_2 = p(x_2|x_1)$,
 $f_3 = p(x_3|x_2)$

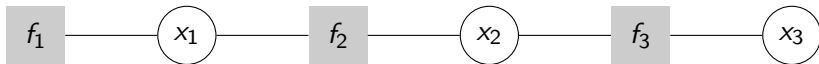


$$p(X) = \prod_s f_s(X_s)$$

Message passing

Message passing: Idea

$$p(X) = \prod_s f_s(X_s)$$



Message passing: Idea

$$p(X) = \prod_s f_s(X_s)$$

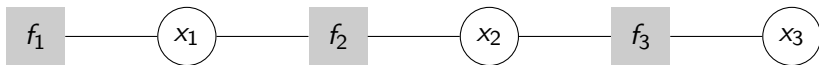


Marginalization naive:

$$p(x_2) = \sum_{x_1} \sum_{x_3} p(X) \in \mathcal{O}(k^n)$$

Message passing: Idea

$$p(X) = \prod_s f_s(X_s)$$



Marginalization naive:

$$p(x_2) = \sum_{x_1} \sum_{x_3} p(X) \in \mathcal{O}(k^n)$$

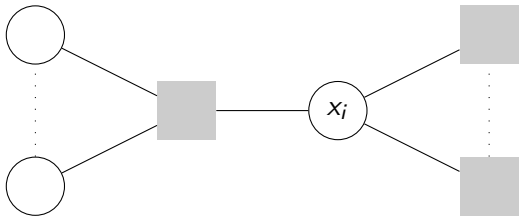
Marginalization advanced:

$$\begin{aligned} p(x_2) &= \sum_{x_1} \sum_{x_3} p(x_1) p(x_2|x_1) p(x_3|x_2) \\ &= \underbrace{\left[\sum_{x_1} p(x_1) p(x_2|x_1) \right]}_{\mu_{f_2 \rightarrow x_2}} \cdot \underbrace{\left[\sum_{x_3} p(x_3|x_2) \right]}_{\mu_{f_3 \rightarrow x_2}} \in \mathcal{O}(n \cdot k^2) \end{aligned}$$

Sum-product algorithm 1

General marginalization:

$$p(x_i) = \prod_{s \in ne(x_i)} \mu_{f_s \rightarrow x_i}(x_i)$$

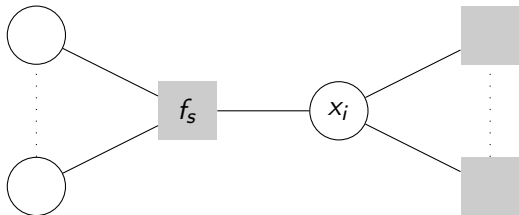


Sum-product algorithm 2

Local messages through factor graph:

$$\mu_{f_s \rightarrow x_i}(x_i) = \sum_{X_s \setminus x_i} f_s(x_i, X_s) \prod_{m \in ne(f_s) \setminus x_i} \mu_{x_m \rightarrow f_s}(x_m)$$

$$\mu_{x_m \rightarrow f_s}(x_m) = \prod_{l \in ne(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)$$

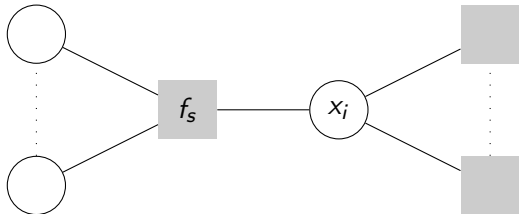


Sum-product algorithm 2

Local messages through factor graph:

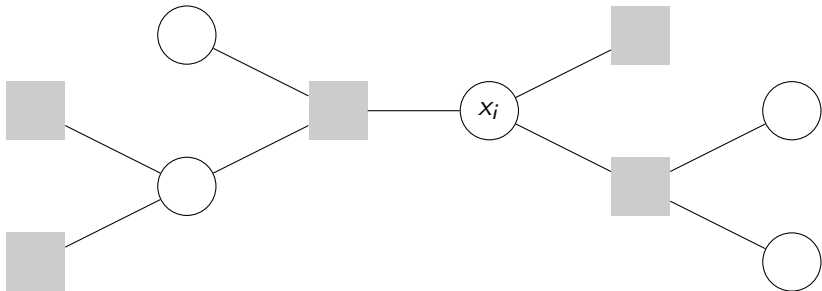
$$\mu_{f_s \rightarrow x_i}(x_i) = \sum_{X_s \setminus x_i} f_s(x_i, X_s) \prod_{m \in ne(f_s) \setminus x_i} \mu_{x_m \rightarrow f_s}(x_m)$$

$$\mu_{x_m \rightarrow f_s}(x_m) = \prod_{l \in ne(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)$$

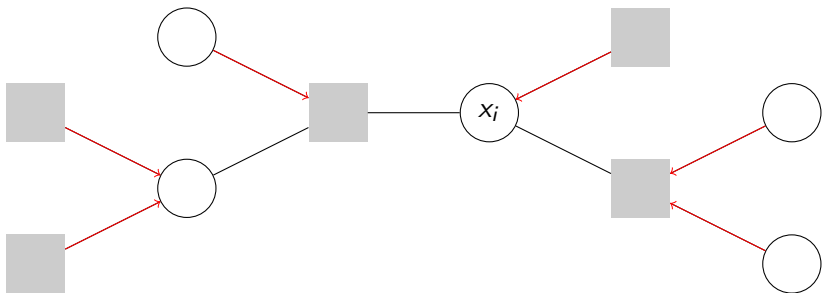


Initialization at leaf nodes: $\mu_{x \rightarrow f}(x) = 1$, $\mu_{f \rightarrow x}(x) = f(x)$

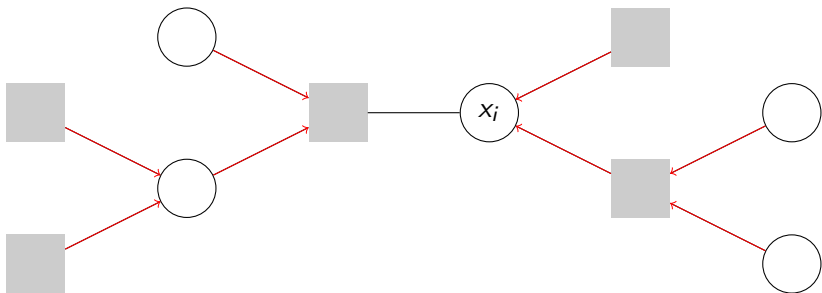
Message passing in action



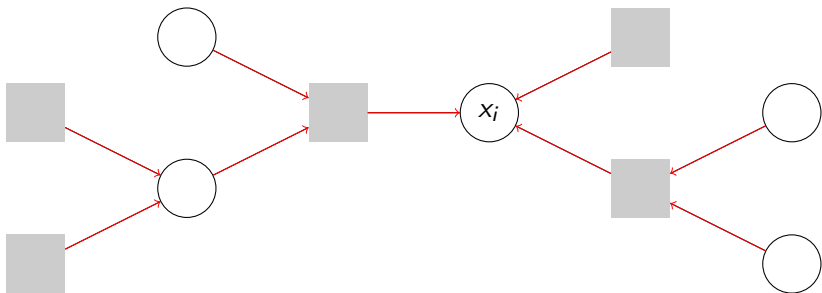
Message passing in action



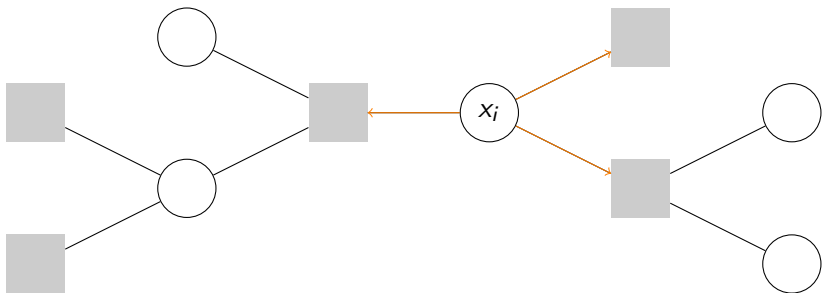
Message passing in action



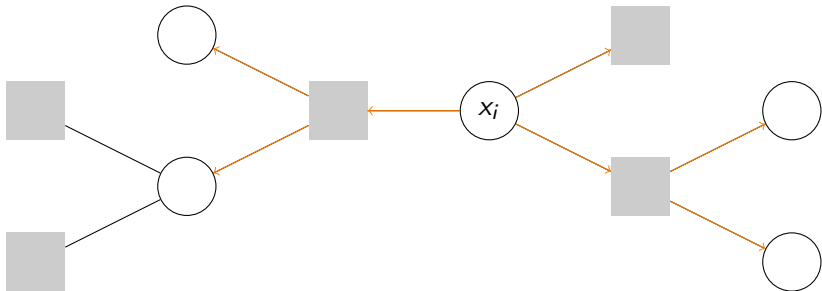
Message passing in action



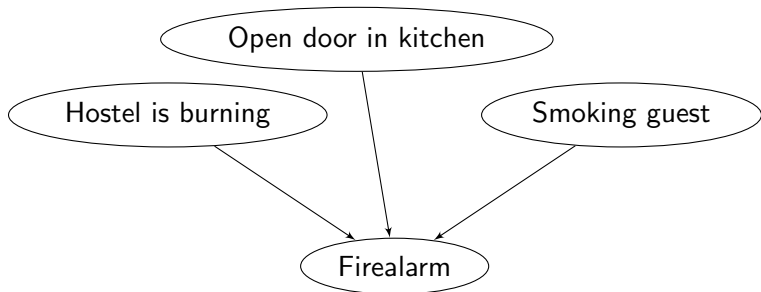
Message passing in action



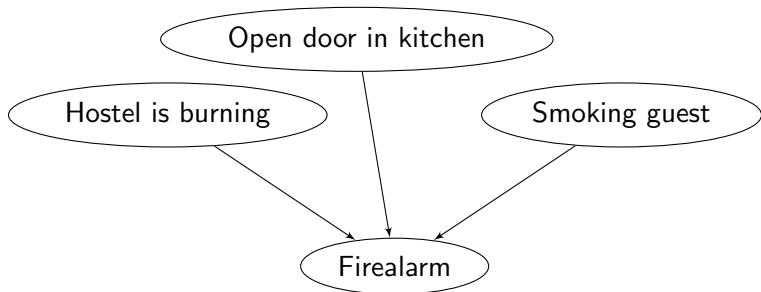
Message passing in action



Maximum a posteriori estimation



Maximum a posteriori estimation



$$X^* = \arg \max_X p(X)$$

Maximum a posteriori estimation:

$$X^* = \arg \max_X p(X) = \arg \max_X \ln(p(X))$$

Max-sum algorithm

Maximum a posteriori estimation:

$$X^* = \arg \max_X p(X) = \arg \max_X \ln(p(X))$$

Infer $p(X^*)$ by adapting sum-product algorithm:

$$\mu_{f_s \rightarrow x_i}(x_i) = \max_{X_s \setminus x_i} \left[f_s(x_i, X_s) \sum_{m \in ne(f_s) \setminus x_i} \mu_{x_m \rightarrow f_s}(x_m) \right]$$

$$\mu_{x_m \rightarrow f_s}(x_m) = \sum_{l \in ne(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)$$

Max-sum algorithm

Maximum a posteriori estimation:

$$X^* = \arg \max_X p(X) = \arg \max_X \ln(p(X))$$

Infer $p(X^*)$ by adapting sum-product algorithm:

$$\mu_{f_s \rightarrow x_i}(x_i) = \max_{X_s \setminus x_i} \left[f_s(x_i, X_s) \sum_{m \in ne(f_s) \setminus x_i} \mu_{x_m \rightarrow f_s}(x_m) \right]$$

$$\mu_{x_m \rightarrow f_s}(x_m) = \sum_{l \in ne(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)$$

Initialization at leaf nodes: $\mu_{x \rightarrow f}(x) = 0$, $\mu_{f \rightarrow x}(x) = \ln f(x)$

Max-sum algorithm

Maximum a posteriori estimation:

$$X^* = \arg \max_X p(X) = \arg \max_X \ln(p(X))$$

Infer $p(X^*)$ by adapting sum-product algorithm:

$$\mu_{f_s \rightarrow x_i}(x_i) = \max_{X_s \setminus x_i} \left[f_s(x_i, X_s) \sum_{m \in ne(f_s) \setminus x_i} \mu_{x_m \rightarrow f_s}(x_m) \right]$$

$$\mu_{x_m \rightarrow f_s}(x_m) = \sum_{l \in ne(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)$$

Initialization at leaf nodes: $\mu_{x \rightarrow f}(x) = 0$, $\mu_{f \rightarrow x}(x) = \ln f(x)$

Keep track of maximal X values!

- Linear complexity in the number of involved variables

- Linear complexity in the number of involved variables
- Only exact for trees

- Linear complexity in the number of involved variables
- Only exact for trees
- In general graphs: Loopy belief propagation

- Linear complexity in the number of involved variables
- Only exact for trees
- In general graphs: Loopy belief propagation
- Linearized message passing

Expectation propagation

Intractable true posterior distribution:

$$p(X|\mathcal{D}) = \frac{1}{Z} \prod_i f_i(X)$$

Intractable true posterior distribution:

$$p(X|\mathcal{D}) = \frac{1}{Z} \prod_i f_i(X)$$

Approximating function from exponential family:

$$q(X) = \frac{1}{Z} \prod_i \tilde{f}_i(X)$$

Intractable true posterior distribution:

$$p(X|\mathcal{D}) = \frac{1}{Z} \prod_i f_i(X)$$

Approximating function from exponential family:

$$q(X) = \frac{1}{Z} \prod_i \tilde{f}_i(X)$$

Minimize KL divergence:

$$KL(p||q) = \int_X p(X) \ln \frac{p(X)}{q(X)}$$

Intractable true posterior distribution:

$$p(X|\mathcal{D}) = \frac{1}{Z} \prod_i f_i(X)$$

Approximating function from exponential family:

$$q(X) = \frac{1}{Z} \prod_i \tilde{f}_i(X)$$

Minimize KL divergence:

$$KL(p||q) = \int_X p(X) \ln \frac{p(X)}{q(X)}$$

Moment matching!

Expectation propagation: Minimize KL-divergence

Complete function:

$$KL\left(\prod_i f_i(X) \parallel \prod_i \tilde{f}_i(X)\right) \quad \text{intractable}$$

Expectation propagation: Minimize KL-divergence

Complete function:

$$KL\left(\prod_i f_i(X) \parallel \prod_i \tilde{f}_i(X)\right) \quad \text{intractable}$$

All factors independently:

$$KL\left(f_i(X) \parallel \tilde{f}_i(X)\right) \quad \text{simple, non-iterative, inaccurate}$$

Expectation propagation: Minimize KL-divergence

Complete function:

$$KL\left(\prod_i f_i(X) \parallel \prod_i \tilde{f}_i(X)\right) \quad \text{intractable}$$

All factors independently:

$$KL\left(f_i(X) \parallel \tilde{f}_i(X)\right) \quad \text{simple, non-iterative, inaccurate}$$

One factor at a time:

$$KL\left(f_i(X) \cdot \prod_{j \neq i} \tilde{f}_j(X) \parallel \tilde{f}_i(X) \cdot \prod_{j \neq i} \tilde{f}_j(X)\right) \quad \text{simple, iterative, accurate}$$

Expectation propagation: Algorithm

Input: $\forall i : f_i(X)$

Initialization: $\forall i : \tilde{f}_i(X) = 1$

Expectation propagation: Algorithm

Input: $\forall i : f_i(X)$

Initialization: $\forall i : \tilde{f}_i(X) = 1$

Iteratively refine one factor at a time until convergence:

$$\begin{aligned}q^{\setminus j}(\theta) &= \prod_{i \neq j} \tilde{f}_i(\theta) \\q^{new}(\theta) &\propto f_j q^{\setminus j}(\theta) \\\tilde{f}_j &= \frac{1}{Z_j} \frac{q^{new}(\theta)}{q^{\setminus j}(\theta)}\end{aligned}$$

Expectation propagation: Discussion

- Can be applied to any graphical model

Expectation propagation: Discussion

- Can be applied to any graphical model
- Expectation propagation is not guaranteed to converge

Expectation propagation: Discussion

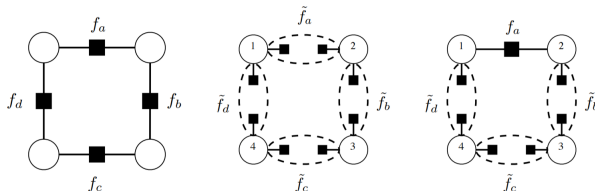
- Can be applied to any graphical model
- Expectation propagation is not guaranteed to converge
- If EP converges, it often outperforms VI

Expectation propagation: Discussion

- Can be applied to any graphical model
- Expectation propagation is not guaranteed to converge
- If EP converges, it often outperforms VI
- Sum-product algorithm special case of EP with fully factorized approximation function

Expectation propagation: Discussion

- Can be applied to any graphical model
- Expectation propagation is not guaranteed to converge
- If EP converges, it often outperforms VI
- Sum-product algorithm special case of EP with fully factorized approximation function



EP vs. VI

Expectation propagation: Minimize $KL(p||q) = \int p \ln \frac{p}{q}$

Variational inference: Minimize $KL(q||p) = \int q \ln \frac{q}{p}$

Expectation propagation: Minimize $KL(p||q) = \int p \ln \frac{p}{q}$

Variational inference: Minimize $KL(q||p) = \int q \ln \frac{q}{p}$

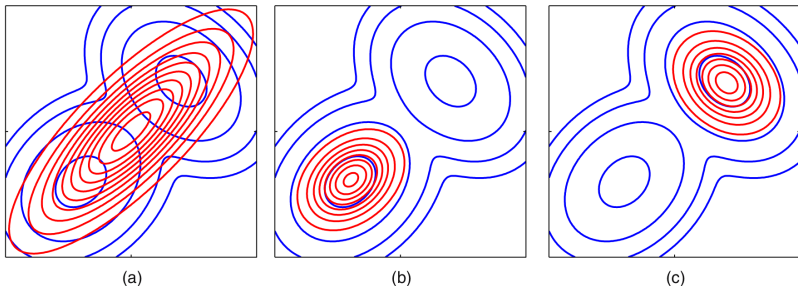
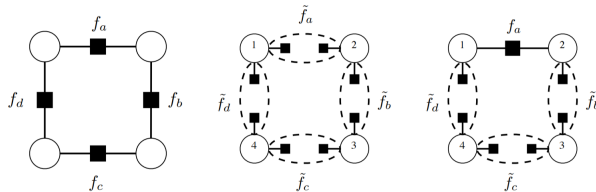


Figure taken from Bishop: Pattern Recognition and Machine Learning.

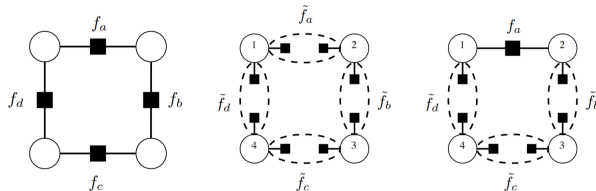
Thanks for your attention!
Questions?

Loopy BP as special case of EP



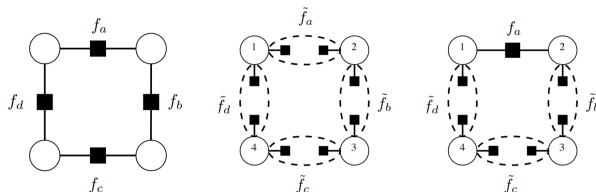
- Fully factorized approximation function

Loopy BP as special case of EP



- Fully factorized approximation function
- Refinement step in EP

Loopy BP as special case of EP



- Fully factorized approximation function
- Refinement step in EP
- If EP converges, it often outperforms VI