Message passing and expectation propagation

Christoph Dehner

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

29 May 2017

Motivation

Graphical models: Bayesian networks and Markov random fields

Motivation

Graphical models: Bayesian networks and Markov random fields

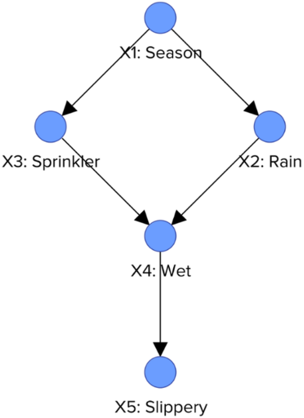
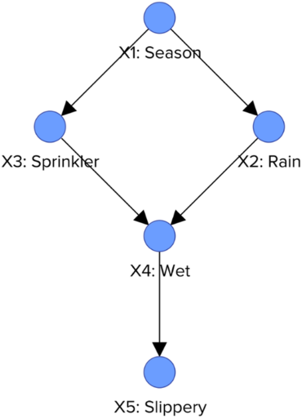


Image sources:
http://www.bayesia.com/hs-fs/hubfs/Bayesian_Networks/Fig_2_2_Sprinkler.png?t=1495201256220
&width=450&height=626&name=Fig_2_2_Sprinkler.png

Motivation

Graphical models: Bayesian networks and Markov random fields



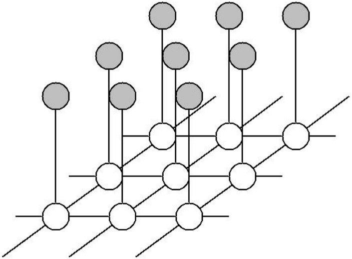
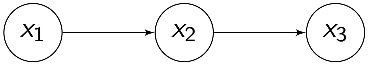


Image sources:
http://www.bayesia.com/hs-fs/hubfs/Bayesian_Networks/Fig_2_2_Sprinkler.png?t=1495201256220
&width=450&height=626&name=Fig_2_2_Sprinkler.png
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0809/ORCHARD/mrf.jpg

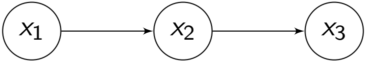
Factor graph

Bayesian network: $p(X) = p(x_1)p(x_2|x_1)p(x_3|x_1)$

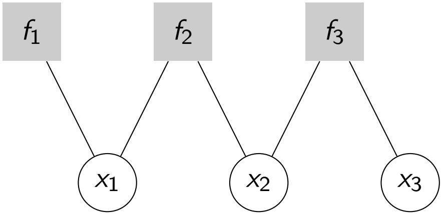


Factor graph

Bayesian network: $p(X) = p(x_1)p(x_2|x_1)p(x_3|x_1)$



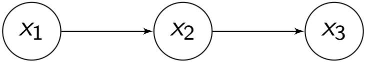
Corresponding factor graph:



Message passing

Message passing: Idea

Bayesian network: $p(X) = p(x_1)p(x_2|x_1)p(x_3|x_1)$



Message passing: Idea

Bayesian network: $p(X) = p(x_1)p(x_2|x_1)p(x_3|x_1)$

$$x_1 \longrightarrow x_2 \longrightarrow x_3$$

Marginalization naive:

$$p(x_2) = \sum_{X \setminus x_2} p(X)$$

Message passing: Idea

Bayesian network: $p(X) = p(x_1)p(x_2|x_1)p(x_3|x_1)$

$$x_1 \longrightarrow x_2 \longrightarrow x_3$$

Marginalization naive:

$$p(x_2) = \sum_{X \setminus x_2} p(X)$$

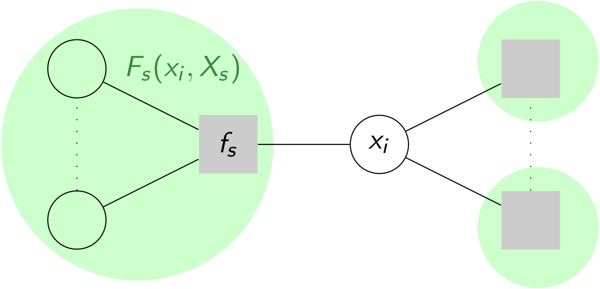
Marginalization advanced:

$$\begin{aligned} p(x_2) &= \sum_{x_1} \sum_{x_3} p(x_1)p(x_2|x_1)p(x_3|x_2) \\ &= \underbrace{\left[\sum_{x_1} p(x_1)p(x_2|x_1) \right]}_{\mu_{x_1 \to x_2}} \cdot \underbrace{\left[\sum_{x_3} p(x_3|x_2) \right]}_{\mu_{x_3 \to x_2}} \end{aligned}$$

Sum-product algorithm 1

General marginalization:

$$p(x_i) = \sum_{X \setminus x_i} \Bigg[\prod_{s \in ne(x_i)} F_s(x_i, X_s) \Bigg] = \prod_{s \in ne(x_i)} \underbrace{\Bigg[\sum_{X_s} F_s(x_i, X_s) \Bigg]}_{\mu_{f_s \to x_i}(x_i)}$$



Sum-product algorithm 2

Local messages through factor graph:

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

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

Sum-product algorithm 2

Local messages through factor graph:

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

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

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

Sum-product algorithm 2

Local messages through factor graph:

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

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

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

Propagate messages through the graph!

Max-sum algorithm

Maximum aposteriori estimate:

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

Max-sum algorithm

Maximum aposteriori estimate:

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

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

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

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

Max-sum algorithm

Maximum aposteriori estimate:

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

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

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

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

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

Message passing: Discussion

- Linear complexity in the number of involved variables

Message passing: Discussion

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

Message passing: Discussion

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

Message passing: Discussion

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

Expecatation propagation

Expectation propagation: Methodology

Approximate posterior distribution:

$$p(\theta|\mathbf{D}) = \frac{1}{p(\mathbf{D})} p(\theta, \mathbf{D}) = \frac{1}{p(\mathbf{D})} \prod_i f_i(\theta, \mathbf{D})$$

Expectation propagation: Methodology

Approximate posterior distribution:

$$p(\theta|\mathbf{D}) = \frac{1}{p(\mathbf{D})} p(\theta, \mathbf{D}) = \frac{1}{p(\mathbf{D})} \prod_i f_i(\theta, \mathbf{D})$$

Approximating function from exponential family:

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

Expectation propagation: Methodology

Approximate posterior distribution:

$$p(\theta|\mathbf{D}) = \frac{1}{p(\mathbf{D})} p(\theta, \mathbf{D}) = \frac{1}{p(\mathbf{D})} \prod_i f_i(\theta, \mathbf{D})$$

Approximating function from exponential family:

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

Minimize KL divergence $KL(p||q)$: Moment matching!

Expectation propagation: Algorithm

Approximating function:

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

Expectation propagation: Algorithm

Approximating function:

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

Iteratively refine one factor at a time:

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

Expectation propagation: Discussion

- Can be applied to general distributions

Expectation propagation: Discussion

- Can be applied to general distributions
- Expectation propagation is not guaranteed to converge

Expectation propagation: Discussion

- Can be applied to general distributions
- Expectation propagation is not guaranteed to converge
- If EP converges, it often outperforms VI

Expectation propagation: Discussion

- Can be applied to general distributions
- Expectation propagation is not guaranteed to converge
- If EP converges, it often outperforms VI
- sum-product algorithm as 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 $KL(q||p) = \int q \ln \frac{q}{p}$

EP vs. VI

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

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

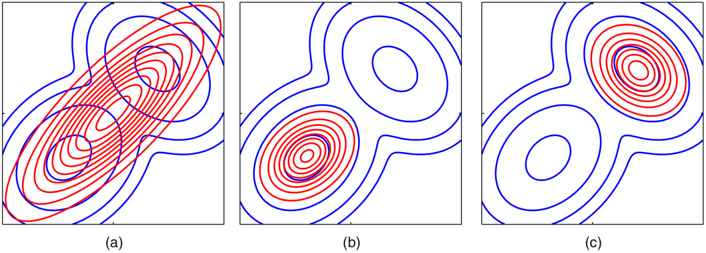


Figure taken from Bishop: Pattern Recognition and Machine Learning.

Thanks for your attention!
Questions?