Approximate Bayesian Estimation: Lecture 1
Graphical Models and Sum-Product Algorithm

Lan Hua

Northwestern Polytechnical University
Key Laboratory of Information Fusion Technology

E-mail: lanhua@nwpu.edu.cn
Tel: 15399035213

October 18, 2019

Lecture Outline

1. Background
2. Bayesian Networks
3. Markov Random Fields
4. Inference in Graphical Models
5. Example of Inference for Target Tracking

Background

Probabilistic graphical models

- provide a simple way to visualize the structure of a probabilistic model
- insights into the property of model, i.e., conditional independence
- complex computations can be expressed by graphical manipulations

- useful in many statistical and computational fields:
  - machine learning, artificial intelligence
  - computational biology, bioinformatics
  - statistical signal/image processing
  - statistical physics
  - communication and information theory

Background

Probabilistic Graphical Model

- node: represents a random variable or a group of random variables
- link: express probabilistic relationship between these variables

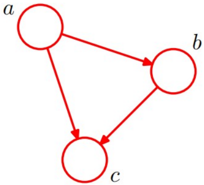
- directed graphical models (Bayesian network): links have a particular directionality $\rightarrow$ express causal relationships between random variables
- undirected graphical models (Markov random fields): no directionality $\rightarrow$ express soft constraints between random variables
- both directed and undirected graphs can be converted into a *factor graph* for solving inference problems.

A graph $G = (V, E)$ comprises *nodes* (*vertices*) connected by *links* (*edges*).

Bayesian Networks

Consider an arbitrary joint distribution $p(a, b, c)$ over variables a, b and c. By application of the product rule of probability, we have

$$p(a, b, c) = p(c|a, b)p(b|a)p(a) \tag{1}$$



- introduce a node for each variables a, b, c
- associate each node with conditional distribution
- add directed links on which node is conditioned

If there is a link going from a node a to a node b, then we say that node a is the *parent* of node b, and node b is the *child* of node a.

Bayesian Networks

Example 1

Draw the probability graphical model of joint distribution $p(x1, \ldots, x_7)$ which has the following factorization

$$\begin{aligned} p(x_1, \ldots, x_7) =& p(x_1)p(x_2)p(x_3) \\ & \times p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5) \end{aligned} \tag{2}$$

Bayesian Networks

Relationship between a given directed and the distribution: The joint distribution defined by a graph is given by the product, over all of the nodes of the graph, of a conditional distribution for each node conditioned on the variables corresponding to the parents of that node in the graph.

For a graph with K nodes, the joint distribution is given by

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k | \mathrm{pa}_k) \tag{3}$$

where pa_k denotes the set of parents of x_k, and $\mathbf{x} = \{x_1, \ldots, x_K\}$.

Bayesian Networks

Example 2. Linear-Gaussian models Consider an arbitrary directed acyclic graph over D variables in which node i represents a single continuous random variable x_i having a Gaussian distribution. The mean of this distribution is taken to be a linear combination of the states of its parent nodes pa_i of node i

$$p(x_i|\mathrm{pa}_k) = \mathcal{N}\left(x_i|\sum_{j\in\mathrm{pa}_i} w_{ij}x_j + b_i, v_i\right) \tag{4}$$

where w_{ij} and b_i are parameters governing the mean, and v_i is the variance of the conditional distribution for x_i.

Bayesian Networks

According to Eq.(3), the log of joint distribution is

$$\begin{aligned} \ln p(\mathbf{x}) &= \sum_{i=1}^{D} \ln p(x_i|\mathrm{pa}_i) \\ &= -\sum_{i=1}^{D} \frac{1}{2v_i}\Big(x_i - \textstyle\sum_{j\in\mathrm{pa}_i} w_{ij}x_j - b_i\Big)^2 + \mathrm{const} \end{aligned} \tag{5}$$

Note that Eq.(5) has a quadratic function form of the components of $\mathbf{x}$, and hence the joint distribution $p(\mathbf{x})$ is a multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

Determine μ and Σ

Each variable x_i has a Gaussian distribution of form Eq.(4), i.e.,

$$x_i = \sum\nolimits_{j \in \mathrm{pa}_i} w_{ij} x_j + b_i + \sqrt{v_i}\epsilon_i \tag{6}$$

with $\epsilon_i \sin \mathcal{N}(0,1)$ being standard normal distribution. Taking the expectation of Eq.(6), we have

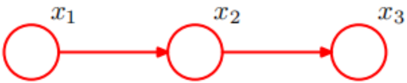
$$\mathbb{E}[x_i] = \sum\nolimits_{j \in \mathrm{pa}_i} w_{ij} \mathbb{E}[x_j] + b_i \tag{7}$$

and

$$\begin{aligned} \mathrm{cov}[x_i, x_j] =& \mathbb{E}[(x_i - \mathbb{E}[x_i])(x_j - \mathbb{E}[x_j])] \\ =& \mathbb{E}\left[(x_i - \mathbb{E}[x_i])\left(\sum\nolimits_{k \in \mathrm{pa}_j} w_{jk}(x_k - \mathbb{E}[x_k]) + \sqrt{v_j}\epsilon_j\right)\right] \\ =& \sum\nolimits_{k \in \mathrm{pa}_j} w_{jk} \mathrm{cov}[x_i, x_k] + I_{ij} v_j \end{aligned} \tag{8}$$

The mean $\mathbb{E}[\mathbf{x}]$ and covariance $\mathrm{cov}[\mathbf{x}]$ can be obtain by starting at the lowest numbered node and working recursively through the graph.

Bayesian Networks



Consider the above graph, which has a link missing between variables x_1 and x_3. Using the recursion relations Eqs.(7)-(8), we have

$$\mu =(b_1, b_1 + w_{21} b_1, b_3 + w_{32} b_2 + w_{32} w_{21} b_1) \tag{9}$$

$$\Sigma = \begin{pmatrix} v_1 & w_{21} v_1 & w_{32} w_{21} v_1 \\ w_{21} v_1 & v_2 + w_{21}^2 v_1 & w_3 2(v_2 + w_{21}^2 v_1) \\ w_{32} w_{21} v_1 & w_{32}(v_2 + w_{21}^2 v_1) & v_3 + w_{32}^2 (v_2 + w_{21}^2 v_1) \end{pmatrix} \tag{10}$$

Markov Random Fields

Conditional Independence: Consider three variables a, b and c, and suppose that the conditional distribution of a, given b and c, does not depend on the value of b, i.e.,

$$p(a|b,c) = p(a|c) \tag{11}$$

or in the form of joint distribution

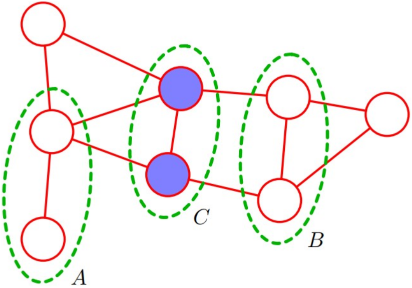
$$p(a,b|c) = p(a|b,c)p(b|c) = p(a|c)p(b|c) \tag{12}$$

We say that a is conditionally independent of b given c.

Conditional independence properties play an important role in simplifying both the structure of a model and the computations needed to perform inference and learning under that model.

Markov Random Fields

Suppose that in an undirected graph we have three sets of nodes, denoted A, B and C. To test the conditional independence of A and B given C, we consider all possible paths that connect nodes in set A to nodes in set B. If all such paths pass through one or more nodes in set C, then the conditional independence property holds.

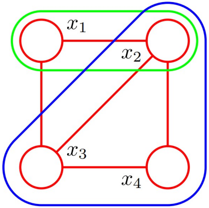


Markov Random Fields

Factorization properties:

Definition

Clique is defined as a subset of the nodes in a graph such that there exists a link between all pairs of nodes in the subset. In other words, the set of nodes in a clique is full connected.



Five cliques of two nodes and two maximal cliques

- $\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}, \{x_4, x_2\}$, and $\{x_1, x_3\}$
- $\{x_1, x_2, x_3\}$ and $\{x_2, x_3, x_4\}$
- Is $\{x_1, x_2, x_3, x_4\}$ a clique?

Markov Random Fields

Let us denote a clique by C and the set of variables in that clique by X_C. Then the joint distribution is written as a product of *potential functions* $\phi_C(X_C)$ over the maximal cliques of the graph

$$p(x) = \frac{1}{Z} \prod_C \phi_C(X_C) \tag{13}$$

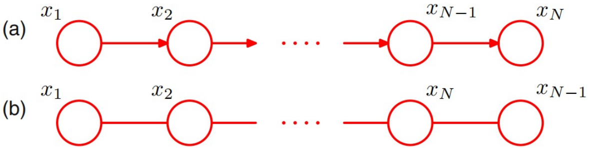
where the normalization constant Z ensures that the distribution $p(x)$ is correctly normalized, i.e.,

$$Z = \sum_x \prod_C \phi_C(X_C) \tag{14}$$

The presence of this normalization constant Z is one of the major limitations of undirected graphs.

Markov Random Fields

Relation to directed graphs:



The joint distribution for the directed graph

$$p(x) = p(x_1)p(x_2|x_1)\cdots(x_N|x_{N-1}) \tag{15}$$

In the undirected graph, the joint distribution is in the form

$$p(x) = \frac{1}{Z}\phi_{1,2}(x_1, x_2)\phi_{2,3}(x_2, x_3)\cdots\phi_{N-1,N}(x_{N-1}, x_N) \tag{16}$$

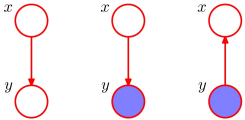
with

$$\phi_{1,2}(x_1, x_2) = p(x_1)p(x_2|x_1), \phi_{2,3}(x_2, x_3) = p(x_3|x_2), \cdots \tag{17}$$

Inference in Graphical Models

Graphical interpretation of Bayes' theorem

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{18}$$



(a) (b) (c)

- (a) joint distribution $p(x, y) = p(x)p(y|x)$.
- (b) the value of y is observed, indicated by shaded node.
- (c) infer the posterior PDF over x by using prior distribution over x and observation y.

Inference on a chain

The joint distribution for this graph (Fig.16-(b)) takes the form

$$p(x) = \frac{1}{Z}\phi_{1,2}(x_1, x_2)\phi_{2,3}(x_2, x_3)\cdots\phi_{N-1,N}(x_{N-1}, x_N) \qquad (19)$$

Assume that N nodes represent discrete variables each having K states, each $\phi_{N-1,N}(x_{N-1}, x_N)$ comprises an $K \times K$ table, and so the joint distribution has $(N-1)K^2$ parameters.

The marginal distribution $p(x_n)$ for a specific node x_n is

$$p(x_n) = \sum_{x_1}\cdots\sum_{x_{n-1}}\sum_{x_{n+1}}\cdots\sum_{x_N} p(x) \qquad (20)$$

The computation ($O(K^N)$) scales exponentially with the chain length N.

Inference in Graphical Models

Efficient algorithm by exploiting the conditional independence properties of the graphical model.

Consider the summation over x_N, the potential $\phi_{N-1,N}(x_{N-1}, x_N)$ only depends on x_N, and so we can perform

$$\sum_{x_N} \phi_{N-1,N}(x_{N-1}, x_N) \tag{21}$$

first to give a function of x_{N-1}, referred as $f(x_{N-1})$, and then use $f(x_{N-1})$ together with the potential $\phi_{N-2,N-1}(x_{N-2}, x_{N-1})$ to perform the summation over x_{N-1}. Similarly, the summation over x_1 involves only the potential $\phi_{1,2}(x_1, x_2)$, and so on.

Inference in Graphical Models

The desired marginal can be expressed in the form

$$\begin{aligned} p(x_n) =& \frac{1}{Z} \underbrace{\left[\sum_{x_{n-1}} \phi_{n-1,N}(x_{n-1}, x_n) \cdots \left[\sum_{x_{n-1}} \phi_{1,2}(x_1, x_2) \right] \right]}_{\mu_\alpha(x_n)} \\ & \times \underbrace{\left[\sum_{x_{n+1}} \phi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \phi_{N-1,N}(x_{N-1}, x_N) \right] \right]}_{\mu_\beta(x_n)} \end{aligned} \tag{22}$$

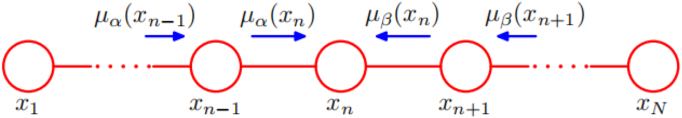
The computation cost is $O(NK^2)$.

Key concept: $ab + bc = a(b + c)$. Three operations reduce to two operations.

Messages passing around on the graph

$$p(x_n) = \frac{1}{Z}\mu_\alpha(x_n)\mu_\beta(x_n) \tag{23}$$

where $\mu_\alpha(x_n)$ represents message passed forwards from node x_{n-1} to x_n, and $\mu_\beta(x_n)$ message passed backwards to node x_n from node x_{n+1}.



Inference in Graphical Models

The messages $\mu_\alpha(x_n)$ and $\mu_\beta(x_n)$ can be evaluated recursively as

$$\begin{aligned}\mu_\alpha(x_n) &= \sum_{x_{n-1}} \phi_{n-1,n}(x_{n-1}, x_n) \left[\sum_{x_{n-2}} \cdots \right] \\ &= \sum_{x_{n-1}} \phi_{n-1,n}(x_{n-1}, x_n) \mu_\alpha(x_{n-1})\end{aligned} \tag{24}$$

$$\begin{aligned}\mu_\beta(x_n) &= \sum_{x_{n+1}} \phi_{n+1,n}(x_{n+1}, x_n) \left[\sum_{x_{n+2}} \cdots \right] \\ &= \sum_{x_{n+1}} \phi_{n+1,n}(x_{n+1}, x_n) \mu_\beta(x_{n+1})\end{aligned} \tag{25}$$

with starting node x_1 and x_N, respectively.

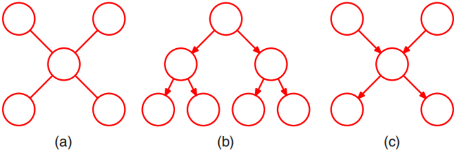
Inference in Graphical Models

Definition

In the case of an undirected graph, a *tree* is defined as a graph in which there is one, and only one, path between any pair of nodes. (No loops)

For directed graphs, a *tree* is defined such that there is single node, called the *root*, which has no parents, and all other nodes have one parent.

If there are nodes in a directed graph that have more than one parent, but still only one path between any two nodes, the graph is called *polytree*.



Inference in Graphical Models

The joint distribution is given as

$$p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s) \tag{26}$$

where $\mathbf{x}_s$ denotes a subset of the variables. Each factor f_s is a function of a corresponding set of variables $\mathbf{x}_s$.

Definition

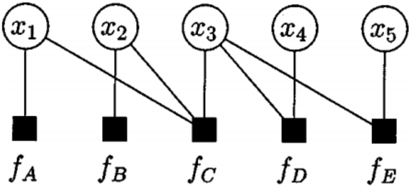
A *factor graph* is a bipartite graph that expresses the structure of the factorization (26). A factor graph has a *variable node* for each variable x_i, a *factor node* for each local function f_s, and an edge-connecting variable node x_i to factor node f_s if and only if x_i is an argument of f_s.

Inference in Graphical Models

A simple factor graph: Let $f(x_1, x_2, x_3, x_4, x_5)$ be a function of five variables, and suppose that f can be expressed as a product

$$f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5) \tag{27}$$

of five factors, so that $s = \{A, B, C, D, E\}$, $\mathbf{x}_A = \{x_1\}$, $\mathbf{x}_B = \{x_2\}$, $\mathbf{x}_C = \{x_3, x_4, x_5\}$, $\mathbf{x}_D = \{x_3, x_4\}$, and $\mathbf{x}_E = \{x_3, x_5\}$.



Inference in Graphical Models

Sum-Product Algorithm: operates by computing various sums and products, evaluating local marginal over nodes or subsets of nodes. Consider the problem of finding the marginal $p(x)$ for variable node x

$$p(x) = \sum_{\mathbf{x}\backslash x} p(\mathbf{x}) \tag{28}$$

where $\mathbf{x}\backslash x$ denotes the set of variables in $\mathbf{x}$ with variable x omitted.

The idea is to substitute $p(\mathbf{x})$ using the factor graph expression (26) and then interchange sums and products to obtain an efficient algorithm.

Inference in Graphical Models

Factor graph is tree structure that allows us to partition the factors in the joint distribution into groups, with one group associated with each of the factor nodes that is a neighbour of the variable node x.

The joint distribution can be written as a product form

$$p(\mathbf{x}) = \prod_{s \in \mathrm{ne}(x)} F_s(x, X_s) \tag{29}$$

where

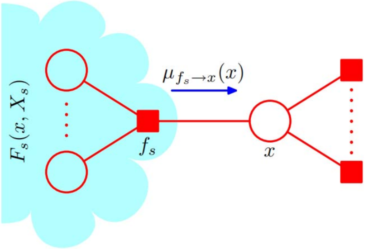
- $\mathrm{ne}(x)$ dentes the set of factor nodes that are neighbours of x
- X_s denotes the set of all variables in the subtree connected to the variable node x via the factor node f_s
- $F_s(x, X_s)$ represents the product of all the factors in the group associated with factor f_s

Inference in Graphical Models

Substituting (29) into (28) and interchanging the sums and products, yield

$$p(x) = \prod_{s \in \mathrm{ne}(x)} \left[\sum_{X_s} F_s(x, X_s) \right] = \prod_{s \in \mathrm{ne}(x)} \mu_{f_s \to x}(x) \tag{30}$$

where $\mu_{f_s \to x}(x)$ can be viewed as *messages* from the factor nodes f_s to the variable node x.



Inference in Graphical Models

Each factor $F_s(x, X_s)$ is described by a factor (sub-)graph, which can be factorized as

$$F_s(x, X_s) = f_s(x, x_1, \dots, x_M) G_1(x_1, X_{s1}) \dots G_M(x_M, X_{sM}) \tag{31}$$

where $x_1, \dots, x_M$ are the neighbours variable nodes of x associated with factor f_s. $G_1(x_1, X_{s1}), \dots, G_M(x_1, X_{sM})$ are the factor that connect with the neighbours variable nodes.

Inference in Graphical Models

Substituting (31) into (30), we obtain

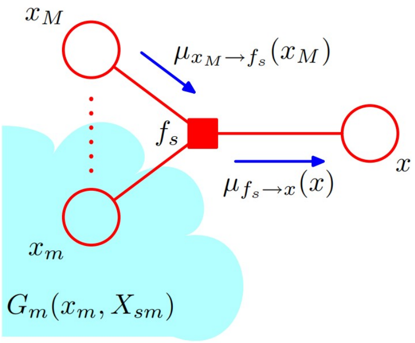
$$\mu_{f_s \to x}(x) = \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \setminus x} \left[\sum_{X_{xm}} G_m(x_m, X_{sm}) \right] \tag{32}$$

$$= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \setminus x} \mu_{x_m \to f_s(x_m)}$$

where $\mu_{x_m \to f_s(x_m)}$ is the message from variable nodes x_m to factor nodes f_s

To evaluate the message sent by a factor node to a variable node along the link connecting them, take the product of the incoming messages along all other links coming into the factor node, multiply by the factor associated with that node, and then marginalize over all of the variables associated with the incoming messages.

Inference in Graphical Models



Two kinds of messages

- $\mu_{f_s \to x}(x)$: from factor node to variable node (incoming message)
- $\mu_{x_m \to f_s(x_m)}$: from variable node to factor node (outing message)

Inference in Graphical Models

The term $G_m(x_m, X_{sM})$ associated with node x_m is given by a product of terms $F_l(x_m, X_{ml})$ each associated with one of the factor nodes f_l that is linked to x_m (excluding node f_s), so that

$$G_m(x_m, X_{sM}) = \prod_{l \in \text{ne}(x_m) \setminus f_s} F_l(x_m, X_{ml}) \tag{33}$$

Substituting (33) into (32), we have

$$\mu_{x_m \to f_s}(x_m) = \prod_{l \in \text{ne}(x_m) \setminus f_s} \left[\sum_{X_{ml}} F_l(x_m, X_{ml}) \right] = \prod_{l \in \text{ne}(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m) \tag{34}$$

To evaluate the message sent by a variable node to an adjacent factor node, take the product of incoming messages along all of the other links.

Inference in Graphical Models

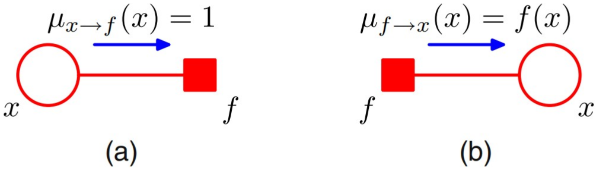
Initialization:

If a leaf node is a variable node, then

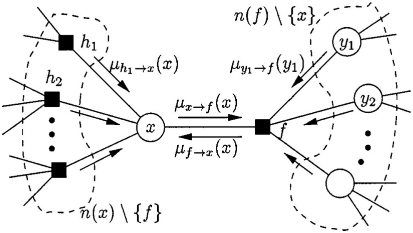
$$\mu_{x\to f}(x) = 1 \tag{35}$$

and if the leaf node is a factor node, then

$$\mu_{f\to x}(x) = f(x) \tag{36}$$



Inference in Graphical Models



The Sum-Product Update Rule:

variable node to factor node:

$$\mu_{x \to f}(x) = \prod_{h \in \mathrm{ne}(x) \setminus f} \mu_{h \to x}(x) \tag{37}$$

factor node to variable node:

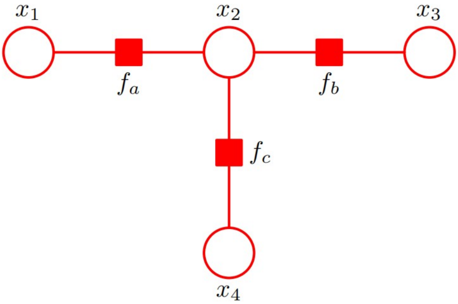
$$\mu_{f \to x}(x) = \sum_{\mathrm{ne}(f) \setminus x} \left[f(X) \prod_{m \in \mathrm{ne}(f) \setminus x} \mu_{m \to f}(m) \right] \tag{38}$$

Inference in Graphical Models

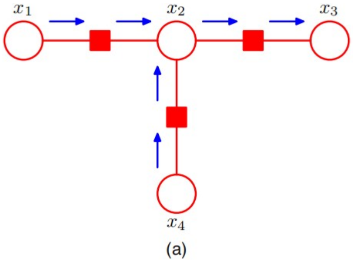
Example of the sum-product algorithm to the graph

The unnormalized joint distribution is given by

$$\tilde{p}(x) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4) \tag{39}$$



Inference in Graphical Models



Inference in Graphical Models

Choose node x_3 as the root, and x_1 and x_4 as the leaf nodes. Starting with the leaf nodes, we have

$$\mu_{x_1 \to f_a}(x_1) = 1 \tag{40}$$

$$\mu_{f_a \to x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2) \tag{41}$$

$$\mu_{x_4 \to f_c}(x_4) = 1 \tag{42}$$

$$\mu_{f_c \to x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4) \tag{43}$$

$$\mu_{x_2 \to f_b}(x_2) = \mu_{f_a \to x_2}(x_2) \mu_{f_c \to x_2}(x_2) \tag{44}$$

$$\mu_{f_b \to x_3}(x_3) = \sum_{x_2} f_b(x_2, x_3) \mu_{x_2 \to f_b}(x_2) \tag{45}$$

Inference in Graphical Models

Propagation messages from the root node out to the leaf nodes

$$\mu_{x_3 \to f_b}(x_3) = 1 \tag{46}$$

$$\mu_{f_b \to x_2}(x_2) = \sum_{x_3} f_b(x_2, x_3) \tag{47}$$

$$\mu_{x_2 \to f_a}(x_2) = \mu_{f_b \to x_2}(x_2)\mu_{f_c \to x_2}(x_2) \tag{48}$$

$$\mu_{x_2 \to f_c}(x_2) = \mu_{f_b \to x_2}(x_2)\mu_{f_a \to x_2}(x_2) \tag{49}$$

$$\mu_{f_a \to x_1}(x_1) = \sum_{x_2} f_a(x_1, x_2)\mu_{x_2 \to f_a}(x_2) \tag{50}$$

$$\mu_{f_c \to x_4}(x_4) = \sum_{x_2} f_c(x_2, x_4)\mu_{x_2 \to f_c}(x_2) \tag{51}$$

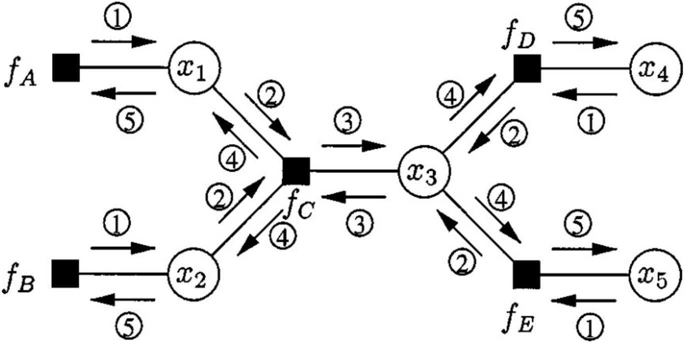
Inference in Graphical Models

The marginal $p(x_2)$ is given by

$$\begin{aligned}\tilde{p}(x_2) =& \mu_{f_a \to x_2}(x_2) \mu_{f_b \to x_2}(x_2) \mu_{f_c \to x_2}(x_2) \\ =& \sum_{x_1} f_a(x_1, x_2) \sum_{x_3} f_b(x_2, x_4) \sum_{x_4} f_c(x_2, x_4) \\ =& \sum_{x_1, x_3, x_4} \tilde{p}(x)\end{aligned} \tag{52}$$

Inference in Graphical Models

Test



References

Christopher M.Bishop, Pattern Recognition and Machine Learning, *Spring* , 2006.

Frank R. Kschischang, *et al.*, Factor Graphs and the Sum-Product Algorithm, *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

The End