# Natural Language Processing CSCI 4152/6509 — Lecture 18 Sum-Product (Message-passing) Algorithms for BN Inference

Instructors: Vlado Keselj
Time and date: 16:05 – 17:25, 2-Nov-2023
Location: Rowe 1011

# Previous Lecture

- **HMM as Bayesian Network**
- Bayesian Network definition
- Burglar-earthquake example
- Computational tasks
- BN inference using brute force
- Complexity of general inference in BNs
- Sum-product algorithms (started)

# Computation Problems Solved by Message Passing

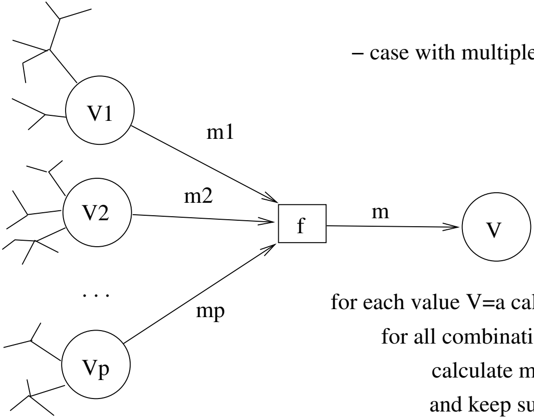
- Applicable to all inference problems
- Two main types of computation:
  - **Summation** of resulting overall products where variables take different domain values
  - **Maximization:** Finding variable values for which the resulting overall product is maximized
- Two main situations:
  - Factor node passing a message to variable node
  - Variable node passing a message to factor node

# Four Cases of Message Computation

- Actually, we can distinguish 4 cases of message computation:

1. Factor node with multiple neighbours to variable node
2. Factor leaf node to variable node
3. Variable node with multiple neighbours to factor node
4. Variable leaf node to factor node

# Factor Node with Multiple Neighbours Passing a Message to Variable Node




– case with multiple neighbours

for each value V=a calculate m(a):

for all combinations of V1 .. Vp

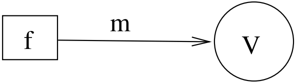
calculate m1*m2*..mp*f

and keep sum or max

m(a) is resulting sum or max

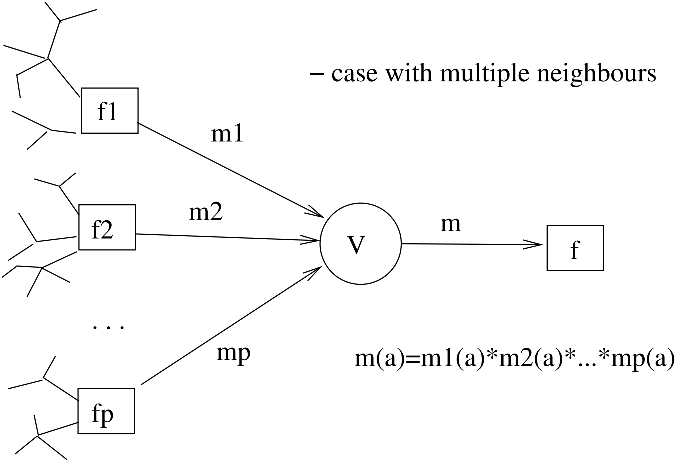
# Factor Node with No Other Neighbours Passing a Message to Variable Node

– case with no other neighbours



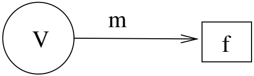

for each value V=a : m(a) = f(a)

# Variable Node with Multiple Neighbours Passing a Message to Factor Node



# Variable Node with No Other Neighbours Passing a Message to Factor Node

– case with no other neighbours




for each value a of V: m(a) = 1

# Solving Inference Tasks

- Distinguish the following cases of inference tasks:

1. Marginalization with one variable
2. Marginalization in general
3. Conditioning with one variable
4. Conditioning in general
5. Completion

# Marginalization with One Variable

- $\mathrm{P}(V_i = x_i) = ?$
- Apply general message passing rules with summation
- At the end

$$\mathrm{P}(V_i = x_i) \;=\; M_{f_1 \to V_i}(x_i) \cdots M_{f_p \to V_i}(x_i)$$

- Running time: $O(nm^{p+1})$

# Marginalization in General

- Consider calculating $\mathrm{P}(V_1 = x_1, ..., V_k = x_k)$.
- The variables $V_1, ..., V_k$ are called *evidence variables* and the instantiated values $x_1, \ldots, x_k$ are called *observed evidence*.
- An evidence-variable to function message is computed in the same way as before if $x = x_j$ (i.e., it is equal to observed evidence), otherwise it is 0.
- Final computation is done in any evidence node $V_j$:

$$\mathrm{P}(V_1 = x_1, ..., V_k = x_k) \;=\; M_{f_1 \to V_j}(x_j) \cdots M_{f_p \to V_j}(x_j)$$

# Conditioning with One Variable

Let us assume that we need to calculate the following conditional probability: $\mathrm{P}(V_{k+1}=y_{k+1}|V_1=x_1,\ldots,V_k=x_k)$. We can use the same message passing algorithm as above, treating $V_1,\ldots,V_k$ as *evidence variables*, except that

- once all of the messages have been passed, then the final conditional probability can be determined by

$$\begin{aligned}\mathrm{P}(V_{k+1}=y_{k+1}|V_1=x_1,\ldots,V_k=x_k)\\ = \frac{M_{f_1\to V_{k+1}}(y_{k+1})\cdots M_{f_p\to V_{k+1}}(y_{k+1})}{Z}\end{aligned}$$

  where $Z$ is a normalization constant over choices of $V_{k+1}$; that is,

$$Z = \sum_y M_{f_1\to V_{k+1}}(y)\cdots M_{f_p\to V_{k+1}}(y)$$

# Conditioning in General

To compute arbitrary conditional probability $\mathrm{P}(\mathbf{V}_\alpha = \mathbf{y}_\alpha | \mathbf{V}_\beta = \mathbf{x}_\beta)$, where $\alpha$ and $\beta$ are two disjoint sets of indices from $\{1, \ldots, n\}$, we can use formula:

$$\mathrm{P}(\mathbf{V}_\alpha = \mathbf{y}_\alpha | \mathbf{V}_\beta = \mathbf{x}_\beta) = \frac{\mathrm{P}(\mathbf{V}_\alpha = \mathbf{y}_\alpha, \mathbf{V}_\beta = \mathbf{x}_\beta)}{\mathrm{P}(\mathbf{V}_\beta = \mathbf{x}_\beta)},$$

where we know how to calculate marginal probabilities $\mathrm{P}(\mathbf{V}_\alpha = \mathbf{y}_\alpha, \mathbf{V}_\beta = \mathbf{x}_\beta)$ and $\mathrm{P}(\mathbf{V}_\beta = \mathbf{x}_\beta)$ using the message-passing algorithm.

# Completion

Completion with one variable: use conditioning on one variable; otherwise

$$y^*_{k+1}, ..., y^*_n = \underset{y_{k+1},...,y_n}{\arg\max} \mathrm{P}(V_{k+1}=y_{k+1}, ..., V_n=y_n | V_1=x_1, ..., V_k=x_k)$$

use the same message passing algorithm as the algorithm for calculating marginal probability $\mathrm{P}(V_1 = x_1, \ldots, V_k = x_k)$, except:

$$M_{f\to V}(x) = \max_{x_1,...,x_p} f(x, x_1, ..., x_p) M_{V_1\to f}(x_1) \cdots M_{V_p\to f}(x_p)$$

At the end

$$y^*_{k+j} = \underset{y_{k+j}}{\arg\max} \quad M_{f_1\to V_{k+j}}(y_{k+j}) \cdots M_{f_p\to V_{k+j}}(y_{k+j})$$

Variables must be assigned consistently (check by "hard-wiring")