Message passing and EM

Ranjitha P

March 19, 2011

1. Graphical Models
   - Inference
   - Variable Elimination
2. Message Passing

Introduction to Graphical models

- Probabilistic graphical models are diagrammatic representation of joint probability distributions.
- Node: random variable. Links: probabilistic relationships between these variables
- Bayesian networks (directed graphical models) and Markov random fields (Undirected graphical models).
- Often convenient to convert both directed and undirected graphs into a different representation called a factor graph.

- Compute the density $\mathcal{P}(X_F/X_E)$

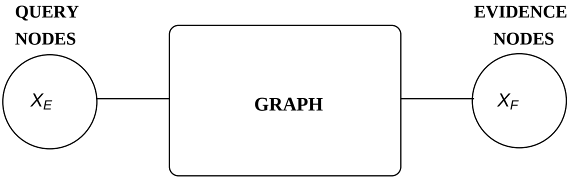

Figure: Inference

- Calculate conditional density of a single query node x_F given an arbitrary set of evidence nodes x_E.
- Computation involved: Marginalization over set of random variables.
- Marginalization takes advantage of the factorization.
- Variable elimination algorithm gives a computationally efficient method for calculating the marginals.

$$\mathcal{P}(x_F, x_E) = \sum_{x_R} \mathcal{P}(x_F, x_E, x_R) \quad \text{also} \quad \mathcal{P}(x_E) = \sum_{x_F} \mathcal{P}(x_F, x_E) \tag{1}$$

Questions to be asked: Given any order of marginalization, does the complexity change? Answer: YES!
Eg: Consider a chain:

$$\mathcal{P}(x_1, x_2, x_3) = \mathcal{P}(x_1)\mathcal{P}(x_2/x_1)\mathcal{P}(x_3/x_2) \tag{2}$$

Elimination ordering $I_1 = [1, 2, 3]$ and $I_2 = [2, 3, 1]$

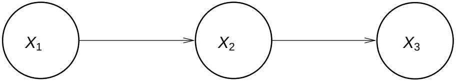

Figure: Chain Structure GM

Algorithm

- Initialize : $\mathcal{G}$, Ordering $\mathcal{I}$ and initialize active list.
- Evidence
- Update : Place suitable $m_i(x_{S_i})$ on the active list.
- Normalize

Motivation

- What if the problem involves computing mutliple conditional densities?
- What if the problem involves computing single node marginals?
- Is it computationally efficient to run VE multiple times?
- Many intermediate factors are same for different conditional densities: Possibility of re-use of factors.
- Optimal order, do not address - simplify : Consider TREES!!

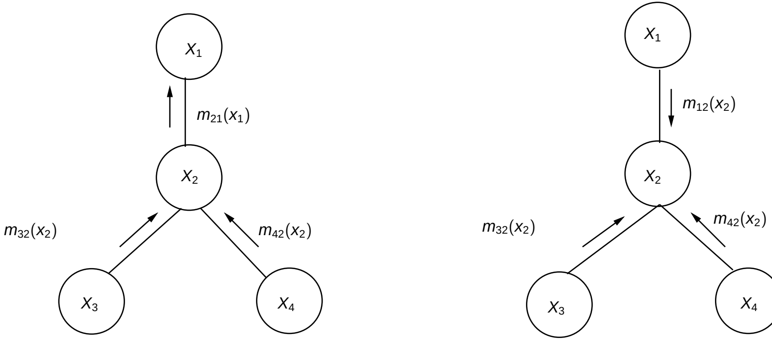

Figure: Message Passing

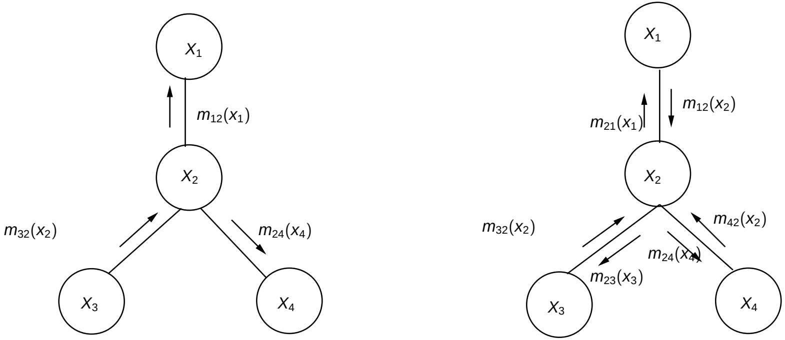

Figure: Message Passing

How does MP work?

Compute messages:

$$m_{ji}(x_i) = \sum_{x_j} \left(\Psi^E(x_j)\Psi(x_i, x_j) \prod_{k \in \mathcal{N}(j) \backslash i} m_{kj}(x_j) \right) \quad (3)$$

The marginal can be written as

$$\mathcal{P}(x_F / x_E) \propto \Psi^E(x_f) \prod_{e \in \mathcal{N}(f)} m_{ef}(x_f) \quad (4)$$

Protocol: A node can send message to a neighboring node only if it has received messages from all its neighbors (other than the target node).

Algorithm

- Recursive implementation possible.
- Collect - Distribute - Compute Marginal

COLLECT(i,j)	DISTRIBUTE(i,j)
for$k \in \mathcal{N}(j)\backslash i$	Send Mesage(j,i)
COLLECT(j,k)	for$k \in \mathcal{N}(j)\backslash i$
Send Mesage(j,i)	DISTRIBUTE(j,k)

- $$\text{Compute Marginal}(i) : \mathcal{P}(x_i) \propto \Psi^E(x_i) \prod_{j \in \mathcal{N}(i)} m_{ji}(x_i)$$

EM - explained using Message Passing in Factor Graphs

- ML estimation problem: $\mathcal{P}(y/s) = \prod_j \mathcal{P}(y_j/s_k, k \in \mathcal{K}(j))$, Y_j depends on $\{s_k : k \in \mathcal{K}(j)\}$.
- Hidden variables X_{jk} associated with each edge, $\mathcal{J}(k) = \{j : k \in \mathcal{K}(j)\}$
- An iterative algorithm is message passing on a graph if computation at a given node at a given iteration use only results of previous computation at that node and information communicated from other connected nodes.

Graph Learning problem

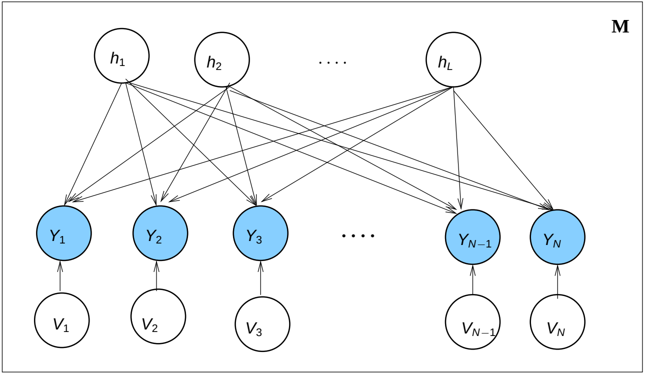

Figure: BN of a SISO-OFDM system in Time Domain