Factor Graphs

Seungjin Choi

Department of Computer Science and Engineering
Pohang University of Science and Technology
77 Cheongam-ro, Nam-gu, Pohang 37673, Korea
seungjin@postech.ac.kr
Tanner Graphs

Parity-check codes: Send $k$ information bits in a block of $N$ bits ($N > k$). (Gallager codes and Turbo codes)

Tanner graph for a parity check code with $N = 6$, $k = 3$:

Parity check 1: $\#1 + \#2 + \#4 = \text{even}$.

Parity check 2: $\#1 + \#5 + \#3 = \text{even}$.

Parity check 3: $\#2 + \#3 + \#6 = \text{even}$. 

Parity check 1:

Parity check 2:

Parity check 3:
Codewords that satisfy these parity constraints are (first 3 bits are information):

000000, 001011, 010101, 011110, 100110, 101101, 110011, 111000.

Noisy channel causes bit-flips:

010101 \rightarrow \text{channel} \rightarrow 011101 \rightarrow \text{decoding} \rightarrow 010101

Decoding is a probabilistic inference problem, inferring the $N$ bits of the transmitted codeword $x_i$ when a sequence of $N$ bits $y_i$ is received.
Assume
\[ p(y|x) = \prod_i p(y_i|x_i). \]

Then, we have
\[
p(x, y) = p(x)p(y|x) \\
= \frac{1}{Z} \psi_{124}(x_1, x_2, x_4)\psi_{135}(x_1, x_3, x_5)\psi_{236}(x_2, x_3, x_6) \prod_{i=1}^{6} p(y_i|x_i). \]

In general,
\[
p(x, y) = \frac{1}{Z} \prod_{a=1}^{N-k} \psi_a(x_a) \prod_{i=1}^{N} p(y_i|x_i). \]
A decoding algorithm for parity check codes that minimizes the number of bits that are decoded incorrectly is to compute the marginal probabilities $p(x_i)$ for $\forall i$ and then threshold each bit to its most probable value.

Taking the most probable value of each bit independently is the strategy that is generated to minimize the number of incorrectly decoded bits even though it may not yield a valid codeword (Gallager 1068).

Factor graph is an extension of Tanner graph with each square (parity check node) representing any function.
Factorization vs Conditional Independence

- Graphical model representations (directed/undirected) aim at characterizing probability distributions in terms of conditional independence statements.
- Factor graphs, an alternative graphical representation of probability distributions, aim at capturing factorizations.
- Conditional independence and factorization are not exactly the same concepts.
- Example: The complete undirected graph with the clique potential $\psi(x_1, x_2, x_3)$

\[
\psi(x_1, x_2, x_3) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_1, x_3) \quad \text{(middle)}
\]

\[
\psi(x_1, x_2, x_3) = f(x_1, x_2, x_3) \quad \text{(right)}
\]
Factor Graphs

Definition  A factor graph is a bipartite graph, $G(\mathcal{V}, \mathcal{F}, \mathcal{E})$, where vertices $\mathcal{V}$ index the variables, the vertices $\mathcal{F}$ index the factors, and edges $\mathcal{E}$ are connected between $\mathcal{V}$ and $\mathcal{F}$.

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 $f$, i.e., $x_i \in x_{C_s}$.

Expresses the structure of the factorization:

$$f(x_1, x_2, \ldots, x_n) \triangleq \prod_{s=1}^{S} f_s(x_{C_s}).$$

Example  $\psi(x_1, x_2, x_3, x_4, x_5) = f_a(x_1, x_3)f_b(x_3, x_4)f_c(x_2, x_4, x_5)f_d(x_1, x_3)$
Directed Graph $\rightarrow$ Factor Graph

- All directed and undirected models can be trivially converted into factor graphs by introducing a factor for each parent-conditional distribution or each clique potential.

- However, given that we do not assume that the subsets $C_s$ correspond to cliques of an underlying graph, we do not need to moralize in the factor graph formalism. This is consistent with the fact that the factor graph does not attempt to represent conditional independencies.
The Sum-Product Algorithm: Review

1. Choose a root node (arbitrarily or as first query node).
2. If $j$ is an evidence node, $\psi^E(x_j) = \delta(x_j, \bar{x}_j)$, else $\psi^E(x_j) = 1$.
3. Pass messages from leaves up to root and then back down using

$$m_{ji}(x_i) = \sum_{x_j} \left[ \psi^E(x_i) \psi(x_i, x_j) \prod_{k \in N(j) \setminus i} m_{kj}(x_j) \right].$$

4. Given messages, compute the marginal of $x_f$ using

$$p(x_f | \bar{x}_E) \propto \psi^E(x_f) \prod_{k \in N(f)} m_{kf}(x_f)$$
Message Passing

- **Factor Tree**: Factor graphs that are trees, i.e., if the undirected graph obtained by ignoring the distinction between variable nodes and factor nodes is an undirected tree.

- Messages that flow from variable nodes to factor nodes
  \[ \nu_{ib}(x_i) = \prod_{a \in \mathcal{N}(i) \setminus b} \mu_{ai}(x_i) \]

- Messages that flow from factor nodes to variable nodes (\( x_b = (x_i, x_j, x_k) \))
  \[ \mu_{bi}(x_i) = \sum_{x_b \setminus x_i} \left( f_b(x_b) \prod_{j \in \mathcal{N}(b) \setminus i} \nu_{jb}(x_j) \right) \]
**BP Equations**

- **Belief** $b_i(x_i)$ at variable node $i$ is the BP approximation to the exact marginal distribution $p_i(x_i)$:

  $$b_i(x_i) \propto \prod_{a \in \mathcal{N}(i)} \mu_{ai}(x_i).$$

- **Joint belief** $b_a(x_a)$ at nodes adjacent to the factor node $f_a$, is given by

  $$b_a(x_a) \propto f_a(x_a) \prod_{i \in \mathcal{N}(a)} \nu_{ia}(x_i) \quad \propto \quad f_a(x_a) \prod_{i \in \mathcal{N}(a)} \prod_{c \in \mathcal{N}(i) \setminus a} \mu_{ci}(x_i).$$

- One can directly derive message passing rules from BP equations, along with the marginalization condition,

  $$b_i(x_i) = \sum_{x_a \setminus x_i} b_a(x_a).$$
The SUM-PRODUCT Algorithm for Factor Trees
Example (a)-(c)

(a) $m_2(x_2) = 2$

(b) $v_1d(x_1) = v_3e(x_3) = X_1$

(c) $m_2(x_2) = m_2(x_2) = 2$

$fa \rightarrow fd \rightarrow X_2 \rightarrow fe \rightarrow X_3$
Example (d)-(f)

\[ v_3(x_3) = 2 \]

\[ f_a f_b f_c \]

\[ X_1 \]

\[ f_d \]

\[ X_2 \]

\[ f_e \]

\[ X_3 \]

\[ v_2(d(x_2)) = \]

\[ v_2(e(x_2)) = \]

\[ v_2(b(x_2)) = 2 \]

\[ m_{d1}(x_1) = 2 \]

\[ m_{e3}(x_3) = 2 \]

\[ v_1(a(x_1)) = 2 \]

\[ v_3(c(x_3)) = 2 \]
"m messages" are equal to "μ messages".

\[
\mu_{si} = \sum_{x_i, x_j} \left( f_s(x_{N(s)}) \prod_{j \in N(s) \setminus i} \nu_{js}(x_j) \right) = \sum_{x_j} \psi(x_i, x_j) \nu_{js}(x_j)
\]

\[
= \sum_{x_j} \psi(x_i, x_j) \prod_{j \in N(j) \setminus \{s\}} \mu_{tj}(x_j) = \sum_{x_j} \left( \psi^E(x_j) \psi(x_i, x_j) \prod_{j \in N'(j) \setminus s} \mu_{tj}(x_j) \right)
\]
The **Sum-Product** algorithm for factor trees applies directly to the graph in (c), because the factor graph is a tree, while the original graph is not.

If the variables in an undirected graphical model can be clustered into non-overlapping cliques, and the parameterization of each clique is a general, non-factorized potential, then the corresponding factor graph is a tree, and the **Sum-Product** applies directly.
**Polytrees**

A **polytree** is a directed graph that reduces to an undirected tree if we convert each directed edge to an undirected edge.