Tutorial on Factor Graph and Belief Propagation

References:

Factor Graphs and the Sum-Product Algorithm
by Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger,
IT Trans. Feb. 2001

Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms
by Jonathan S. Yedidia, Wiliam T. Freeman, and Yair Weiss

Marginal Functions

$$g_i(x_i) = \sum_{x_i} g(x_1, \ldots, x_n)$$

Example:
posteriori probability distribution $p(\mathbf{x} \mid \mathbf{y})$:

$$g(x_1, \ldots, x_n) = p(x_1, \ldots, x_n|\mathbf{y})$$

likelihood function

$$g(x_1, \ldots, x_n) = p(\mathbf{y} \mid x_1, \ldots, x_n)$$

Marginalize: estimate individual symbols.

Factor Graph

Consider the class of functions

$$g(x_1, \ldots, x_n) = \prod_{j \in J} f_j(X_j)$$

Example:

$$g(x_1, \ldots, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5)$$

Example: Linear Codes

Example: Linear Codes, Tanner Graph

Parity check matrix

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Code C is the set of $\mathbf{x}$ such that $H\mathbf{x} = 0$, it's characteristic function is:

$$\begin{aligned} \chi_C(x_1, \ldots, x_6) &\triangleq 1_{[(x_1,\ldots,x_6)\in C]} \\ &= \prod_j 1_{[j^{th} \text{ parity check is satisfied}]} \end{aligned}$$

Linear Codes, Noisy Observations

posteriori distribution

$$p(\mathbf{x} \mid \mathbf{y}) \stackrel{\triangle}{=} g(\mathbf{x}) \propto f(\mathbf{y} \mid \mathbf{x}) p(\mathbf{x})$$

Assuming a memoryless channel, and equally likely codewords

$$g(\mathbf{x}) = \frac{1}{C} \chi_C(\mathbf{x}) \prod_{i=1}^{n} f(y_i \mid x_i)$$

Example: Kalman Filter

$$\begin{aligned} x_{j+1} &= A_j x_j + B_j u_j \\ y_j &= C_j x_j + D_j w_j \end{aligned}$$

Consider

$$\begin{aligned} g(x_1, \ldots, x_n) &= f(x_1, \ldots, x_n \mid y_1, \ldots, y_n) \\ &\propto \prod_{j=1}^{n} f(x_j \mid x_{j-1}) f(y_j \mid x_j) \end{aligned}$$

Marginalize:

$$g_n(x_n) = f(x_n \mid y_1, \ldots, y_n)$$

gives the MMSE of x_n and the estimation error distribution.

Example: Kalman Filter

Factor Graph for Kalman Filtering:

Factor Graph and Expression Tree

Idea: sum over one variable at a time.

Example:

$$g(x_1, \ldots, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5)$$

Factor Graph and Expression Tree

- Each factor node sums over all the variables that are its children, since they are not related to any other function

$$m_{a \to i} = \sum_{\mathbf{x}_a / x_i} f_a(\mathbf{x}_a) \prod_{j \in N(a)/i} n_{j \to a}(x_j)$$

- Each variable node combines all its children factors

$$n_{i \to a}(x_i) = \prod_{b \in N(i)/a} m_{b \to i}(x_i)$$

Belief Propagation

At a variable node, belief

$$b_i(x_i) \propto \prod_{a \in N(i)} m_{a \to i}(x_i)$$

Approximation to the exact marginal distribution $p_i(x_i)$.

At a variable node, belief

$$b_a(\mathbf{x}_a) \propto f_a(\mathbf{x}_a) \prod_{i \in N(a)} n_{i \to a}(x_i)$$

Approximation of the joint distribution of $\mathbf{x}_a$.
Marginalize:

$$b_i(x_i) = \sum_{\mathbf{x}_a / x_i} b_a(\mathbf{x}_a)$$

Example: parity check node

$$\sum_{x_1, x_2} f_a(x_3 = 0, x_1, x_2) p(x_1, x_2) = P(x_1 + x_2 = 0)$$

Message transmitted in and out a variable node i is an approximate of the marginal distribution of x_i.

Example: Kalman Filtering

$$\begin{aligned} x_{j+1} &= A_j x_j + B_j u_j \\ y_j &= C_j x_j + D_j w_j \end{aligned}$$

Goal: forward estimation

$$\begin{aligned} P_{n|n}(x_n) &= f(x_n \mid y_1, \ldots, y_n) \\ &= \int_{\sim\{x_n\}} f(x_1, \ldots, x_n \mid y_1, \ldots, y_n) d(\sim \{x_n\}) \end{aligned}$$

Example: Kalman Filtering

Update rule:

$$P_{j|j}(x_j) = P_{j|j-1}(x_j) f(y_j|x_j) \propto \mathcal{N}(\hat{m}_{j|j}, \sigma^2_{j|j})$$

estimate of x_j based on observations $y_1, \ldots, y_j$.

$$P_{j+1|j}(x_{j+1}) = \int P_{j|j}(x_j)\mathcal{N}(A_j x_j, B_j^2) dx_j \propto \mathcal{N}(\hat{m}_{j+1|j}, \sigma^2_{j+1|j})$$

estimate of x_{j+1} based on observations $y_1, \ldots, y_j$.

Computing All Marginal Functions

Each node sends message on an edge only when it has received messages from all other edges.

Example: Kalman Filter (again)

Factor Graphs with Cycles

Iterative Processing

- LDPC codes
- RA codes
- ...

Message-passing Schedules

- flooding
- serial

Free Energy

For a factor graph probability distribution

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{a=1}^{M} f_a(\mathbf{x}_a)$$

define the *energy* of a state as

$$E(\mathbf{x}) = -\sum_{i=1}^{M} log f_a(\mathbf{x}_a)$$

Helmholtz free energy

$$F_{helmholtz} = -log Z$$

Given $E(\mathbf{x})$, can integral to compute Z and $p(\mathbf{x})$, but it is hard.

Free Energy

Gibbs free energy: for another distribution $b(\mathbf{x})$,

$$\begin{aligned} F(b) &= U(b) - H(b) \\ &= \sum b(\mathbf{x}) E(\mathbf{x}) + \sum b(\mathbf{x}) \log b(\mathbf{x}) \\ &= F_{helmholtz} + D(b||p) \end{aligned}$$

Minimize Gibbs free energy: compute Z and recover p: still hard.

Restrict $b(\mathbf{x}) = \prod_i b_i(x_i)$, minimize *mean field free energy* to have an approximation of $p(\mathbf{x})$.

Region-based Free Energy Approximation

Region energy

$$E_R(\mathbf{x}_R) = -\sum_{a \in F_R} \log f_a(\mathbf{x}_a)$$

Region belief $b_R(\mathbf{x}_R)$ is an approximation of $p_R(\mathbf{x}_R)$.

Minimize

$$F_R(b_R) = U_R(b_R) - H_R(b_R)$$

gives an approximation of the original distribution.

Problem: overlap and over counting.
Solution: introduce counting number.

Bethe Method

Bethe method: a special method of defining the regions and the counting numbers.

Fixed points of the BP algorithm correspond to stationary points of the Bethe approximation of the free energy.

- BP always have a fixed point
- Uniqueness of the stationary points is studied in physics.
- BP does not decrease Bethe free energy at each iteration.
- Generalized BP.