Notes on FutureMapping 2: Gaussian Belief Propagation for Spatial AI

Rom Parnichkun

April 14, 2022

1 Tutorial on Belief Propagation

Marginal distribution is found by taking the joint distribution, and summing over all of the other variables:

$$p(x) = \sum_{\mathbf{x} \setminus x} p(\mathbf{x}). \tag{1}$$

The total joint probability can be written as the product of all factors in a graph: [1]

$$p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s), \tag{2}$$

where for some measurement z_s,

$$f_s(\mathbf{x}_s) = p(z_s \mid \mathbf{x}_s) \tag{3}$$

Setting F_s as the product of all factors associated with f_s (a factor that is directly connnected to x) we get:

$$p(\mathbf{x}) = \prod_{s \in n(x)} F_s(x, \mathbf{X}_s), \tag{4}$$

where $n(x)$ is the set of factor nodes that are neighbors of x and $\mathbf{X}_s$ is the vector of all variables in the subtree conntected to x via f_s. Note that with equation 4 we reduce the problem of finding the joint probability to a local one $s \in n(x)$.

Next, combining equations 1 and 4 we get:

$$p(x) = \sum_{\mathbf{x} \setminus x} \Big[\prod_{s \in n(x)} F_s(x, \mathbf{X}_s) \Big]. \tag{5}$$

We can reorder the sum and product to obtain:

$$p(x) = \prod_{s \in n(x)} \Big[\sum_{\mathbf{X}_s} F_s(x, \mathbf{X}_s) \Big]. \tag{6}$$

Intuitively, this reordering is simply changing the marginalisation to be over a subtree, then multiplying over all the marginals to get the final marginal. It effectively decouples the marginalisation process to a subtree (from the summation over $\mathbf{x} \setminus x$ to $\mathbf{X}_s$) allowing us to define a "message" from f_s as shown below. [2]

$$\mu_{f_s \to x} = \sum_{\mathbf{X}_s} F_s(x, \mathbf{X}_s). \tag{7}$$

Substituting Equation 7 to 6 results in

$$\boxed{p(x) = \prod_{s \in n(x)} \Big[\mu_{f_s \to x} \Big].} \tag{8}$$

[1]Note that $p(\mathbf{x})$ here may be unnormalized.
[2]A message is like a local marginalisation.

Next, we break down $F_s(x, \mathbf{X}_s)$ as

$$\begin{aligned} F_s(x, \mathbf{X}_s) &= f_s(x, x_1, \ldots, x_M) \\ &\quad \times \left[G_1(x_1, \mathbf{X}_{S_1}), \ldots, G_M(x_M, \mathbf{X}_{S_M}) \right] \\ &= f_s(x, x_1, \ldots, x_M) \prod_{m \in n(f_s)} G_m(x_m, \mathbf{X}_{S_m}), \end{aligned} \tag{9}$$

in which the factor f_s is a function of all nodes conntected to it, and $G_m(x_m, X_{S_m})$ is defined as follows.

$$G_m(x_m, \mathbf{X}_{S_m}) = \prod_{l \in n(x_m) \setminus f_s} F_l(x_m, \mathbf{X}_{m_l}). \tag{10}$$

Equation 10 shows the recursive nature of computing F_s ($F \to G \to F \ldots$), therefore we can reorganize and substitute equation 9 to equation 7 to reflect this nature.

$$\begin{aligned} \mu_{f_s \to x}(x) &= \sum_{\mathbf{X}_s} \Big[f_s(x, x_1, \ldots, x_M) \prod_{m \in n(f_s)} G_m(x_m, \mathbf{X}_{S_m}) \Big] \\ &= \sum_{x_1, \ldots, x_M} \sum_{\mathbf{X}_{S_1}, \ldots, \mathbf{X}_{S_M}} \Big[f_s(x, x_1, \ldots, x_M) \prod_{m \in n(f_s)} G_m(x_m, \mathbf{X}_{S_m}) \Big] \\ &= \sum_{x_1, \ldots, x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in n(f_s)} \sum_{\mathbf{X}_{S_1}, \ldots, \mathbf{X}_{S_M}} G_m(x_m, \mathbf{X}_{S_m}) \end{aligned} \tag{11}$$

Similar to Equation 7, G_m can be thought of as a message coming from the variable, hence we define the message $\mu_{x_m \to f_s}(x_m)$ as follows:

$$\mu_{x_m \to f_s}(x_m) = \sum_{\mathbf{X}_{S_m}} G_m(x_m, \mathbf{X}_{S_m}) \tag{12}$$

Substituting $\mu_{x_m \to f_s}$ to equation 11 we get

$$\boxed{\begin{aligned} \mu_{f_s \to x}(x) = \sum_{x_1, \ldots, x_M} & f_s(x, x_1, \ldots, x_M) \\ & \times \prod_{m \in n(f_s)} \mu_{x_m \to f_s}(x_m). \end{aligned}} \tag{13}$$

Next, we substitute Equation 10 to 12 to get

$$\mu_{x_m \to f_s}(x_m) = \sum_{\mathbf{X}_{S_m}} \prod_{l \in n(x_m) \setminus f_s} F_l(x_m, \mathbf{X}_{m_l}), \tag{14}$$

and swap the order of the sum and product to obtain:

$$\mu_{x_m \to f_s}(x_m) = \prod_{l \in n(x_m) \setminus f_s} \sum_{\mathbf{X}_{m_l}} F_l(x_m, \mathbf{X}_{m_l}). \tag{15}$$

Finally we derive the message from variables that completes the recursion

$$\boxed{\mu_{x_m \to f_s}(x_m) = \prod_{l \in n(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m).} \tag{16}$$

In order to find the marginal distribution for x, we start from all of the leaf nodes of the factor graph relative to x, and pass messages inwards torwards x. To initialise the leaf nodes, a variable leaf node sends a message $\mu_{x \to f}(x) = 1$ to its only connected factor and a leaf factor sends $\mu_{f \to x}(x) = f(x)$.

To find the marginal for *all* variables, we send the messages outwards from the root back to the leaves.

2 Gaussian Belief Propagation

If the relationship between variables is linear, representing the probabilities of the variables with a multivariate Gaussian distribution enables the marginals to also be Gaussian distributions.

2.1 Factor Definition

A Gaussian factor can be written as follows.

$$f_s(\mathbf{x}_s) = Ke^{-\frac{1}{2}[(\mathbf{z}_s - \mathbf{h}_s(\mathbf{x}_s))^T \Lambda_s (\mathbf{z}_s - \mathbf{h}_s(\mathbf{x}_s))]}. \tag{17}$$

This expression represents the probability of obtaining measurement vector $\mathbf{z}_s$ from the sensor based on the other variables $\mathbf{x}_s$. $\mathbf{h}_s$ is the function which describes the relationship between the variables on the expected measurement $\mathbf{z}_s$. Matrix $\mathbf{\Lambda}_s$ is the precision of the measurement.

With the information form (canonical form), the Gaussian distrubution can be represented as follows.[3]

$$f_s(\mathbf{x}_s) = Ke^{[-\frac{1}{2}\mathbf{x}_s^T \mathbf{\Lambda}_s^{'} \mathbf{x}_s + \boldsymbol{\eta}_s^T \mathbf{x}_s]}, \tag{18}$$

where

$$\boldsymbol{\eta}_s = \mathbf{\Lambda}_s \boldsymbol{\mu}_s. \tag{19}$$

2.2 State Representation

The probability distribution over state variables also have a Gaussian form as follows.

$$p_m(\mathbf{x}_m) = Ke^{[-\frac{1}{2}\mathbf{x}_m^T \mathbf{\Lambda}_m \mathbf{x}_m + \eta_m^T \mathbf{x}_m]}. \tag{20}$$

2.3 Linearising Factors

We can linearise any factor to the equation below

$$f_s(\mathbf{x}_s) = Ke^{[-\frac{1}{2}\mathbf{x}_s^T \mathbf{\Lambda}_s^{'} \mathbf{x}_s + \boldsymbol{\eta}_s^T \mathbf{x}_s]} \tag{21}$$

$$\boldsymbol{\eta}_s = \mathbf{J}_s^T \mathbf{\Lambda}_s [\mathbf{J}_s \mathbf{x}_0 + \mathbf{z}_s - \mathbf{h}_s(\mathbf{x}_0)] \tag{22}$$

$$\mathbf{\Lambda}_s^{'} = \mathbf{J}_s^T \mathbf{\Lambda}_s \mathbf{J}_s. \tag{23}$$

where $\mathbf{J}_s$ is the Jacobian $\frac{\partial \mathbf{h}_s}{\partial \mathbf{x}_s}|_{\mathbf{x}_s = \mathbf{x}_0}$, $\mathbf{x}_0$ is the point of linearisation.

[3]For more information about the information form read Exactly Sparse Delayed-State Filters.

2.4 Message Passing at a Variable Node

For GBP, each incoming message $\mu_{f_l \to \mathbf{x}_m}(\mathbf{x}_m)$ is represented by an information vector $\boldsymbol{\eta}_{ml}$ and a precision matrix $\mathbf{\Lambda}_{ml}$. The outgoing message $\mu_{\mathbf{x}_m \to f_s}(\mathbf{x}_m)$ is computed by simply adding

$$\boldsymbol{\eta}_{ms} = \sum_{l \in n(\mathbf{x}_m) \setminus f_s} \boldsymbol{\eta}_{ml} \tag{24}$$

$$\mathbf{\Lambda}_{ms} = \sum_{l \in n(\mathbf{x}_m) \setminus f_s} \mathbf{\Lambda}_{ml}. \tag{25}$$

This is equivalent to multiplying the distributions.