

Notes on FutureMapping 2: Gaussian Belief Propagation for Spatial AI

Rom Parnichkun

April 12, 2022

1 Tutorial on Belief Propagation

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

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

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

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

where for some measurement z_s ,

$$f_s(\mathbf{x}_s) = p(z_s | \mathbf{x}_s) \quad (3)$$

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

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

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 connected 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:

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

We can reorder the sum and product to obtain:

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

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

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

Substituting Equation 7 to 6 results in

$$p(x) = \prod_{s \in n(x)} [\mu_{f_s \rightarrow x}].$$

¹Note that $p(\mathbf{x})$ here may be unnormalized.

²A message is like a local marginalisation.

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

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

in which the factor f_s is a function of all nodes connected to it, and $G_m(x_m, \mathbf{X}_{S_m})$ is defined as follows.

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

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

$$\begin{aligned} \mu_{f_s \rightarrow x}(x) &= \sum_{\mathbf{X}_s} \left[f_s(x, x_1, \dots, x_M) \prod_{m \in n(f_s)} G_m(x_m, \mathbf{X}_{S_m}) \right] \\ &= \sum_{x_1, \dots, x_M} \sum_{\mathbf{X}_{S_1}, \dots, \mathbf{X}_{S_M}} \left[f_s(x, x_1, \dots, x_M) \prod_{m \in n(f_s)} G_m(x_m, \mathbf{X}_{S_m}) \right] \\ &= \sum_{x_1, \dots, x_M} f_s(x, x_1, \dots, x_M) \prod_{m \in n(f_s)} \sum_{\mathbf{X}_{S_1}, \dots, \mathbf{X}_{S_M}} G_m(x_m, \mathbf{X}_{S_m}) \end{aligned} \quad (11)$$

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

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

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

$$\begin{aligned} \mu_{f_s \rightarrow x}(x) &= \sum_{x_1, \dots, x_M} f_s(x, x_1, \dots, x_M) \\ &\times \prod_{m \in n(f_s)} \mu_{x_m \rightarrow f_s}(x_m). \end{aligned} \quad (13)$$

Next, we substitute Equation 10 to 12 to get

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

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

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

Finally we derive the message from variables that completes the recursion

$$\mu_{x_m \rightarrow f_s}(x_m) = \prod_{l \in n(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m). \quad (16)$$

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 towards x . To initialise the leaf nodes, a variable leaf node sends a message $\mu_{x \rightarrow f}(x) = 1$ to its only connected factor and a leaf factor sends $\mu_{f \rightarrow x}(x) = f(x)$.

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.

$$f_s(\mathbf{x}_s) = K e^{-\frac{1}{2}[(\mathbf{z}_s - \mathbf{h}_s(\mathbf{x}_s))^T \Lambda_s (\mathbf{z}_s - \mathbf{h}_s(\mathbf{x}_s))]}, \quad (17)$$

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 Λ_s is the precision of the measurement.

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

$$f_s(\mathbf{x}_s) = K e^{[-\frac{1}{2}\mathbf{x}_s^T \Lambda_s' \mathbf{x}_s + \boldsymbol{\eta}_s^T \mathbf{x}_s]}, \quad (18)$$

where

$$\boldsymbol{\eta}_s = \Lambda_s \boldsymbol{\mu}_s. \quad (19)$$

2.2 State Representation

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

$$p_m(\mathbf{x}_m) = K e^{[-\frac{1}{2}\mathbf{x}_m^T \Lambda_m \mathbf{x}_m + \boldsymbol{\eta}_m^T \mathbf{x}_m]}. \quad (20)$$

2.3 Linearising Factors

We can linearise any factor to the equation below

$$f_s(\mathbf{x}_s) = K e^{[-\frac{1}{2}\mathbf{x}_s^T \Lambda_s' \mathbf{x}_s + \boldsymbol{\eta}_s^T \mathbf{x}_s]} \quad (21)$$

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

$$\Lambda_s' = \mathbf{J}_s^T \Lambda_s \mathbf{J}_s. \quad (23)$$

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

³For 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 $\mu_{f_l \rightarrow \mathbf{x}_m}(\mathbf{x}_m)$ is represented by an information vector $\boldsymbol{\eta}_{ml}$ and a precision matrix Λ_{ml} . The outgoing message $\mu_{\mathbf{x}_m \rightarrow f_s}(\mathbf{x}_m)$ is computed by simply adding

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

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

This is equivalent to multiplying the distributions.

2.5 Message Passing at a Factor Node