12 Gaussian Belief Propagation

Our inference story so far has focused on discrete random variables, starting with the Elimination algorithm for marginalization. For trees, we saw how there is always a good elimination order, which gave rise to the Sum-Product algorithm for trees, also referred to as *belief propagation* (BP). For the problem of determining MAP configurations of the variables, we introduced the MAP Elimination algorithm, which is obtained from its marginalization counterpart by replacing summations with maximizations and keeping track of the maximizing argument of every message. For trees, an efficient implementation of elimination yielded the Max-Product algorithm. We then specialized our algorithms to the case of the simplest class of trees—the hidden Markov model—obtaining the forward-backward algorithm as the specialization of Sum-Product, and the Viterbi algorithm as the specialization of Max-Product.

In the case of continuous-valued distributions, we can of course continue to use the Elimination Algorithm, where now summations are replaced with integrals. Obviously our computational complexity analysis does not carry over from the discrete case, as we need to think about the complexity of integration—which could be intractable for arbitrary continuous distributions. However, as we develop in these notes, for the special case of jointly Gaussian distributions, explicit integration can be entirely avoided by exploiting the algebraic structure of the underlying distributions, leading, once again, to highly efficient algorithms!

We begin with the Gaussian Sum-Product algorithm, or equivalently, *Gaussian belief propagation*.

12.1 Preliminary Example: the Two-Node Case

We begin with a two-node undirected graph for the jointly Gaussian case, observing the message passing structure associated with eliminating one node. The insights here will help us anticipate the structure for the general case of trees.

Our development makes use of a key result from our earlier introduction to Gaussian graphical models, whereby in information form, marginalization is computed via a Schur complement:

Claim 1. *If*

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \sim \mathtt{N}^{-1}\left(\begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{pmatrix}, \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix} \right) \tag{1}$$

then

$$\mathbf{x}_1 \sim \mathtt{N}^{-1}(\mathbf{h}', \mathbf{J}') \tag{2}$$

where

$$\mathbf{h}' = \mathbf{h}_1 - \mathbf{J}_{12}\mathbf{J}_{22}^{-1}\mathbf{h}_2 \quad \textit{and} \quad \mathbf{J}' = \mathbf{J}_{11} - \mathbf{J}_{12}\mathbf{J}_{22}^{-1}\mathbf{J}_{21}. \tag{3}$$

Proceeding, consider the case of two jointly Gaussian random variables $\mathbf{x}_1$ and $\mathbf{x}_2$ distributed according to

$$\begin{aligned}
&p_{\mathbf{x}_1,\mathbf{x}_2}(\mathbf{x}_1,\mathbf{x}_2) \\
&\propto \exp\left\{-\frac{1}{2}\mathbf{x}_1^{\mathrm{T}}\mathbf{J}_{11}\mathbf{x}_1 + \mathbf{h}_1^{\mathrm{T}}\mathbf{x}_1 - \frac{1}{2}\mathbf{x}_2^{\mathrm{T}}\mathbf{J}_{22}\mathbf{x}_2 + \mathbf{h}_2^{\mathrm{T}}\mathbf{x}_2 - \mathbf{x}_1^{\mathrm{T}}\mathbf{J}_{12}\mathbf{x}_2\right\} \\
&= \underbrace{\exp\left\{-\frac{1}{2}\mathbf{x}_1^{\mathrm{T}}\mathbf{J}_{11}\mathbf{x}_1 + \mathbf{h}_1^{\mathrm{T}}\mathbf{x}_1\right\}}_{\triangleq\phi_1(\mathbf{x}_1)} \underbrace{\exp\left\{-\frac{1}{2}\mathbf{x}_2^{\mathrm{T}}\mathbf{J}_{22}\mathbf{x}_2 + \mathbf{h}_2^{\mathrm{T}}\mathbf{x}_2\right\}}_{\triangleq\phi_2(\mathbf{x}_2)} \underbrace{\exp\left\{-\mathbf{x}_1^{\mathrm{T}}\mathbf{J}_{12}\mathbf{x}_2\right\}}_{\triangleq\psi_{12}(\mathbf{x}_1,\mathbf{x}_2)}.
\end{aligned}$$

where we want to compute the marginal $p_{\mathbf{x}_1}$. Writing out the associated integration,[1] while interpreting the quantities involved in terms of the sum-product algorithm messages, we obtain

$$p_{\mathbf{x}_1}(\mathbf{x}_1) = \int \phi_1(\mathbf{x}_1)\,\phi_2(\mathbf{x}_2)\,\psi_{12}(\mathbf{x}_1,\mathbf{x}_2)\,\mathrm{d}\mathbf{x}_2 = \phi_1(\mathbf{x}_1)\underbrace{\int \phi_2(\mathbf{x}_2)\psi_{12}(\mathbf{x}_1,\mathbf{x}_2)\,\mathrm{d}\mathbf{x}_2}_{m_{2\to1}(\mathbf{x}_1)}.$$

However, the message $m_{2\to1}$ can be rewritten as

$$\begin{aligned}
m_{2\to1}(\mathbf{x}_1) &= \int \exp\left\{-\frac{1}{2}\mathbf{x}_2^{\mathrm{T}}\mathbf{J}_{22}\mathbf{x}_2 + \mathbf{h}_2^{\mathrm{T}}\mathbf{x}_2 - \mathbf{x}_1^{\mathrm{T}}\mathbf{J}_{12}\mathbf{x}_2\right\}\mathrm{d}\mathbf{x}_2 \\
&= \int \exp\left\{-\frac{1}{2}\begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{0} & \mathbf{J}_{12}\\ \mathbf{J}_{21} & \mathbf{J}_{22}\end{bmatrix}\begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix} + \begin{pmatrix}\mathbf{0}\\ \mathbf{h}_2\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix}\right\}\mathrm{d}\mathbf{x}_2,
\end{aligned}$$

which, conveniently, we can interpret as a marginalization which we can evaluate using a Schur complement! In particular, applying Claim 1 with $\mathbf{J}_{11} = \mathbf{0}$ and $\mathbf{h}_1 = \mathbf{0}$, we then obtain

$$m_{2\to1}(\mathbf{x}_1) \propto \mathtt{N}^{-1}(\mathbf{x}_1; \mathbf{h}_{2\to1}, \mathbf{J}_{2\to1}),$$

where

$$\mathbf{h}_{2\to1} \triangleq -\mathbf{J}_{12}\mathbf{J}_{22}^{-1}\mathbf{h}_2 \quad \text{and} \quad \mathbf{J}_{2\to1} \triangleq -\mathbf{J}_{12}\mathbf{J}_{22}^{-1}\mathbf{J}_{21}.$$

We remark that the proportionality constant does not matter; indeed, even in the discrete case we had the lattitude to normalize messages at each step. However, what is important is the observation that in this case of jointly Gaussian distributions, the messages are Gaussian. In turn, this has important computational implications. Indeed, since Gaussian distributions are characterized by their mean and convariance parameters, in Gaussian inference we need only propagate these parameters. Phrased differently, the mean and covariance parameters constitute equivalent messages in this Gaussian case. As such, the computation involved in the manipulation of these

[1] This is an integration over $\mathbb{R}^d$ if $\mathbf{x}_2$ is d-dimensional.

messages in the inference process amounts to linear algebra, whose complexity is easy to characterize!

To complete our initial example, note that the marginal distribution for $\mathbf{x}_1$, which is also a Gaussian distribution and thereby characterized by its mean and covariance, is obtained from the message $m_{2\to 1}$ via the following final bit of linear algebra:

$$\begin{aligned} p_{\mathbf{x}_1}(\mathbf{x}_1) &\propto \phi_1(\mathbf{x}_1)\, m_{2\to 1}(\mathbf{x}_1) \\ &\propto \exp\left\{-\frac{1}{2}\mathbf{x}_1^{\mathrm{T}}\mathbf{J}_{11}\mathbf{x}_1 + \mathbf{h}_1^{\mathrm{T}}\mathbf{x}_1\right\}\left\{-\frac{1}{2}\mathbf{x}_1^{\mathrm{T}}\mathbf{J}_{2\to 1}\mathbf{x}_1 + \mathbf{h}_{2\to 1}^{\mathrm{T}}\mathbf{x}_1\right\} \\ &= \exp\left\{-\frac{1}{2}\mathbf{x}_1^{\mathrm{T}}(\mathbf{J}_{11}+\mathbf{J}_{2\to 1})\mathbf{x}_1 + (\mathbf{h}_1+\mathbf{h}_{2\to 1})^{\mathrm{T}}\mathbf{x}_1\right\} \\ &\propto \mathbb{N}^{-1}(\mathbf{x}_1; \mathbf{h}_1+\mathbf{h}_{2\to 1}, \mathbf{J}_{11}+\mathbf{J}_{2\to 1}). \end{aligned}$$

In particular, in marginalizing out $\mathbf{x}_2$, we see that $\mathbf{h}_{2\to 1}$ is used in updating the potential vector of $\mathbf{x}_1$, while $\mathbf{J}_{2\to 1}$ is used in updating the information matrix of $\mathbf{x}_1$.

12.2 Undirected Trees

From the two-node example, we saw that messages are Gaussian in the case of Gaussian inference, and that message computation and marginalization involve linear algebraic computation. Thus, when applying the Sum-Product algorithm to the case of Gaussian distributions, our focus is one of determining the associated linear algebra for its implementation, which we now develop, again exploiting the information form of Gaussian distributions and the role of Schur complements in marginalization.

To start, note that the message sent from $\mathbf{x}_i$ to $\mathbf{x}_j$ can be written as

$$\begin{aligned} & m_{i\to j}(\mathbf{x}_j) \\ &= \int \phi_i(\mathbf{x}_i)\psi_{ij}(\mathbf{x}_i,\mathbf{x}_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(\mathbf{x}_i)\,\mathrm{d}\mathbf{x}_i \\ &= \int \exp\left\{-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{ii}\mathbf{x}_i + \mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i\right\}\exp\left\{-\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{ij}\mathbf{x}_j\right\} \\ &\qquad \cdot \prod_{k\in N(i)\setminus j} \exp\left\{-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{k\to i}\mathbf{x}_i + \mathbf{h}_{k\to i}^{\mathrm{T}}\mathbf{x}_i\right\}\mathrm{d}\mathbf{x}_i \\ &= \int \exp\left\{-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{ii}\mathbf{x}_i + \mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i - \mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{ij}\mathbf{x}_j + \sum_{k\in N(i)\setminus j}\left[-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{k\to i}\mathbf{x}_i + \mathbf{h}_{k\to i}^{\mathrm{T}}\mathbf{x}_i\right]\right\}\mathrm{d}\mathbf{x}_i \\ &= \int \exp\left\{-\frac{1}{2}\begin{pmatrix}\mathbf{x}_i\\ \mathbf{x}_j\end{pmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{J}_{ii}+\sum_{k\in N(i)\setminus j}\mathbf{J}_{k\to i} & \mathbf{J}_{ij}\\ \mathbf{J}_{ji} & \mathbf{0}\end{bmatrix}\begin{pmatrix}\mathbf{x}_i\\ \mathbf{x}_j\end{pmatrix}\right. \\ &\qquad \left. + \begin{pmatrix}\mathbf{h}_i+\sum_{k\in N(i)\setminus j}\mathbf{h}_{k\to i}\\ \mathbf{0}\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}\mathbf{x}_i\\ \mathbf{x}_j\end{pmatrix}\right\}\mathrm{d}\mathbf{x}_i. \end{aligned}$$

Applying Claim 1, we obtain

$$m_{i\to j}(\mathbf{x}_j) \propto \mathtt{N}^{-1}(\mathbf{x}_j; \mathbf{h}_{i\to j}, \mathbf{J}_{i\to j}), \tag{4}$$

where

$$\mathbf{h}_{i\to j} = -\mathbf{J}_{ji}\left(\mathbf{J}_{ii} + \sum_{k\in N(i)\setminus j} \mathbf{J}_{k\to i}\right)^{-1}\left(\mathbf{h}_i + \sum_{k\in N(i)\setminus j} \mathbf{h}_{k\to i}\right), \tag{5}$$

$$\mathbf{J}_{i\to j} = -\mathbf{J}_{ji}\left(\mathbf{J}_{ii} + \sum_{k\in N(i)\setminus j} \mathbf{J}_{k\to i}\right)^{-1}\mathbf{J}_{ij}. \tag{6}$$

In turn, the marginal computation can be expressed as

$$\begin{aligned}
p_{\mathbf{x}_i}(\mathbf{x}_i) &\propto \phi_i(\mathbf{x}_i) \prod_{k\in N(i)} m_{k\to i}(\mathbf{x}_i) \\
&\propto \exp\left\{-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{ii}\mathbf{x}_i + \mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i\right\} \prod_{k\in N(i)} \exp\left\{-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{k\to i}\mathbf{x}_i + \mathbf{h}_{k\to i}^{\mathrm{T}}\mathbf{x}_i\right\} \\
&= \exp\left\{-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{ii}\mathbf{x}_i + \mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i + \sum_{k\in N(i)}\left[-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\mathbf{J}_{k\to i}\mathbf{x}_i + \mathbf{h}_{k\to i}^{\mathrm{T}}\mathbf{x}_i\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\left(\mathbf{J}_{ii} + \sum_{k\in N(i)} \mathbf{J}_{k\to i}\right)\mathbf{x}_i + \left(\mathbf{h}_i + \sum_{k\in N(i)} \mathbf{h}_{k\to i}\right)^{\mathrm{T}}\mathbf{x}_i\right\},
\end{aligned}$$

from which we obtain that

$$\mathbf{x}_i \sim \mathtt{N}^{-1}(\hat{\mathbf{h}}_i, \hat{\mathbf{J}}_i)$$

where

$$\hat{\mathbf{h}}_i = \mathbf{h}_i + \sum_{k\in N(i)} \mathbf{h}_{k\to i}, \qquad \hat{\mathbf{J}}_i = \mathbf{J}_{ii} + \sum_{k\in N(i)} \mathbf{J}_{k\to i}. \tag{7}$$

Eqs. (5), (6), and (7) thus define the implementation of the update rules for Gaussian BP. As in the general case, we may run these update rules at each node in parallel.

Having now developed the linear algebra that implements Gaussian belief propagation, we can examine the complexity of Gaussian inference, focusing on the serial version of Gaussian BP. In general, of course, different $\mathbf{x}_i$ can have different dimensions, but to simplify our discussion let us assume that each $\mathbf{x}_i$ has dimension d. At each iteration, the computational complexity is dominated by the matrix inversion in (5), which need not be repeated for computing (6). Now if matrix inversion is implemented via Gaussian elimination, its complexity is $O(d^3)$ complexity[2] Moreover, as

[2]Actually, a lower complexity of $O(d^{2.376})$ is possible via the Coppersmith-Winograd algorithm.

in the case of discrete-valued variables, the number of iterations needed scales with the diameter of the undirected tree, which in turn scales with the number of nodes N. Thus, the overall complexity is $O(Nd^3)$. By contrast, naively inverting the information matrix of the entire graph $\mathbf{J} \in \mathbb{R}^{Nd \times Nd}$ in order to compute marginal means and covariances at each node results in inference complexity of $O((Nd)^3) = O(N^3 d^3)$.

12.3 Connections to Gaussian Elimination

From our initial discussion of Gaussian distributions, recall that the covariance and information forms are related via

$$\mathbf{J} = \mathbf{\Lambda}^{-1} \quad \text{and} \quad \mathbf{h} = \mathbf{J}\boldsymbol{\mu}, \tag{8}$$

from which we see that marginalization, i.e., the computation of $\boldsymbol{\mu}$ from the information form parameterization can be obtained by solving a set of linear equations.

Via this observation, it follows that for any (symmetric) positive-definite matrix $\mathbf{A}$ that corresponds to the information matrix for a tree graph, the linear equation $\mathbf{Ax} = \mathbf{b}$ for $\mathbf{x}$ may be solved by running Gaussian BP on the associated graph! As such, there is a close connection between the Gaussian elimination algorithm for solving linear equations and Gaussian BP, which we give a flavor of by the following simple example.

Example 1. Consider the following pair of jointly Gaussian variables

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim \mathtt{N}^{-1}\left(\begin{pmatrix} 3 \\ 3 \end{pmatrix}, \begin{bmatrix} 4 & 2 \\ 2 & 3 \end{bmatrix} \right).$$

In particular, let's look at solving the following for $\mathbf{x} = (x_1, x_2)$:

$$\begin{bmatrix} 4 & 2 \\ 2 & 3 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}.$$

Subtracting $2/3$ of the second row from the first yields

$$\begin{bmatrix} 4 - \frac{2}{3} \cdot 2 & 2 - \frac{2}{3} \cdot 3 \\ 2 & 3 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 - \frac{2}{3} \cdot 3 \\ 3 \end{pmatrix}.$$

Simplifying yields

$$\begin{bmatrix} \frac{8}{3} & 0 \\ 2 & 3 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix},$$

from which we can back-substitute to obtain $x_1 = 3/8$ and so forth. We leave computing Gaussian BP messages by first eliminating node x_2 as an exercise, noting that the calculations will in fact be identical to those above.

This begs the question: Does this result contradict our earlier computational complexity analysis suggesting that Gaussian BP is more efficient than naively inverting the entire graph's information matrix using Gaussian elimination? We mention two differences here. First, the equivalence assumes the same elimination ordering, which means that while using Gaussian elimination, we need to essentially reorder the rows so that we eliminate the leaves to the root, and choosing a bad elimination ordering will induce a significant computational penalty, akin to using the Elimination Algorithm on a tree with a bad ordering (e.g., trying to first eliminate nodes that are not leaves). Second, Gaussian elimination does not know a priori that matrix $\mathbf{A}$ is sparse and might, for example, perform many multiplications by 0 whereas Gaussian BP for trees will automatically avoid a lot of these operations. We can modify Gaussian elimination to account for such issues, but the intuition for doing so really comes from the tree structure. In fact, once we account for the tree structure and elimination ordering as described above, Gaussian elimination becomes equivalent to Gaussian BP.

12.4 MAP Configurations in the Gaussian Case

For Gaussian distributions, we are also often interested in our other main inference task, viz., MAP estimation. However, conveniently, for jointly Gaussian distributions, marginalization and MAP estimation coincide, i.e., the sum-product and max-product and produce identical results, and thus our development of the sum-product algorithm suffices.

The reason for this is the special tructure and symmetries of jointly Gaussian distributions; indeed, the mean and mode (and median for that matter) of such distributions is identical. Since the mode corresponds to the MAP configuration, and the marginals are parameterized by the mean vector, BP suffices for both.

To verify that the mode of a Gaussian distribution is its mean, it suffices to note that maximizing the distribution $\mathtt{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Lambda})$ over $\mathbf{x}$ amounts to solving the unconstrained maximization problem

$$\begin{aligned}\mathbf{x}^* &= \arg\max_{\mathbf{x}} \left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \right] \\ &= \arg\max_{\mathbf{x}} \left[-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{x} + \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} \right].\end{aligned}$$

Taking the gradient of the objective function with respect to $\mathbf{x}$, we obtain

$$\frac{\partial}{\partial \mathbf{x}} \left[-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{x} + \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} \right] = -\boldsymbol{\Lambda}^{-1}\mathbf{x} + \boldsymbol{\Lambda}^{-1}\boldsymbol{\mu}.$$

In turn, setting the gradient to $\mathbf{0}$ we see that, indeed, the MAP configuration is $\mathbf{x}^* = \boldsymbol{\mu}$.

6.438 Algorithms for Inference

Fall 2014