that messages $m_{ji}(x_i)$ are approximated by a set of weighted, discrete samples. If $\mathcal{X}_i$ is continuous and these messages are constructed from independent proposal distributions, their particles will be distinct with probability one. For the message product operation underlying the BP algorithm to produce sensible results, some interpolation of these samples to nearby states is thus needed.

We accomplish this interpolation, and ensure that messages are smooth and strictly positive, by convolving raw particle sets with a Gaussian distribution, or kernel:

$$m_{ji}(x_i) = \sum_{\ell=1}^{L} w_{ji}^{(\ell)} \mathcal{N}\left(x_i; x_{ji}^{(\ell)}, \Lambda_{ji}\right), \quad (9)$$

$$q_i(x_i) = \sum_{\ell=1}^{L} w_i^{(\ell)} \mathcal{N}\left(x_i; x_i^{(\ell)}, \Lambda_i\right). \quad (10)$$

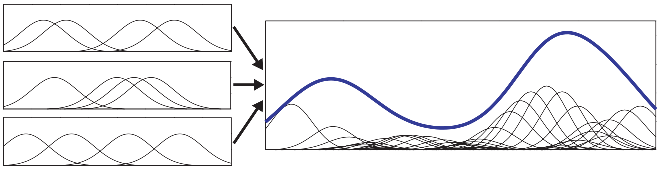
Here, $\mathcal{N}(x; \mu, \Lambda)$ denotes a normalized Gaussian density with mean μ and covariance Λ, evaluated at x. As detailed later, we use methods from the nonparametric density estimation literature to construct these mixture approximations.[42] The product of two Gaussians is itself a scaled Gaussian distribution, a fact which simplifies our later algorithms.

3.2. Message fusion

We begin by assuming that the observation potential is a Gaussian mixture. Such representations arise naturally from learning-based approaches to model identification.[14] The BP belief update of Equation 3 is then defined by a product of $d = (|\Gamma(i)| + 1)$ mixtures: the observation potential $\psi_i(x_i, y)$, and messages $m_{ji}(x_i)$ as in Equation 9 from each neighbor. As illustrated in Figure 3, the product of d Gaussian mixtures, each containing L components, is itself a mixture of L^d Gaussians. While in principle this belief update could be performed exactly, the exponential growth in the number of mixture components quickly becomes intractable.

The NBP algorithm instead approximates the product mixture via a collection of L independent, possibly importance weighted samples $\{(x_i^{(\ell)}, w_i^{(\ell)})\}_{\ell=1}^{L}$ from the "ideal" belief of Equation 3. Given these samples, the bandwidth Λ_i of the nonparametric belief estimate (Equation 10) is determined via a method from the extensive kernel density estimation literature.[42] For example, the simple "rule of thumb" method combines a robust covariance estimate with an asymptotic formula that assumes the target density is Gaussian. While fast to compute, it often oversmooths multimodal distributions. In such cases, more sophisticated cross-validation schemes can improve performance.

Figure 3. A product of three mixtures of $L = 4$ 1D Gaussians. Although the $4^3 = 64$ components in the product density (thin lines) vary widely in position and weight (scaled for visibility), their normalized sum (thick line) has a simple form.



In many applications, NBP's computations are dominated by the cost of sampling from such products of Gaussian mixtures. Exact sampling by explicit construction of the product distribution requires $\mathcal{O}(L^d)$ operations. Fortunately, a number of efficient approximate samplers have been developed. One simple but sometimes effective approach uses an evenly weighted mixture of the d input distributions as an importance sampling proposal. For higher-dimensional variables, iterative Gibbs sampling algorithms are often more effective.[44] Multiscale KD-tree density representations can improve the mixing rate of Gibbs samplers, and also lead to "epsilon-exact" samplers with accuracy guarantees.[25] More sophisticated importance samplers[5] and multiscale simulated or parallel tempering algorithms[39] can also be highly effective. Yet more approaches improve efficiency by introducing an additional message approximation step.[19, 22, 31] By first reducing the complexity of each message, the product can be approximated more quickly, or even computed exactly. When $\psi_i(x_i, y)$ is a non-Gaussian analytic function, we can use any of these samplers to construct an importance sampling proposal from the incoming Gaussian mixture messages.

3.3. Message propagation

The particle filter of Section 2.4 propagates belief estimates to subsequent time steps by sampling $x_{t+1}^{(\ell)} \sim p(x_{t+1} | x_t^{(\ell)})$. The consistency of this procedure depends critically on the HMM's factorization into properly normalized conditional distributions, so that $\int p(x_{t+1} \mid x_t) dx_{t+1} = 1$ for all $x_t \in \mathcal{X}_t$. By definition, such conditional distributions place no biases on x_t. In contrast, for pairwise MRFs, the clique potential $\psi_{ij}(x_i, x_j)$ is an arbitrary nonnegative function that may influence the values assumed by either linked variable. To account for this, we quantify the *marginal influence* of $\psi_{ij}(x_i, x_j)$ on x_j via the following function:

$$\varphi_{ij}(x_j) = \int_{\mathcal{X}_i} \psi_{ij}(x_i, x_j) dx_i. \quad (11)$$

If $\psi_{ij}(x_i, x_j)$ is a Gaussian mixture, $\psi_{ij}(x_j)$ is simply the mixture obtained by marginalizing each component. In the common case where $\psi_{ij}(x_i, x_j) = \tilde{\psi}_{ij}(x_i - x_j)$ depends only on the difference between neighboring variables, the marginal influence is constant and may be ignored.

As summarized in the algorithm of Figure 4, NBP approximates the BP message update of Equation 2 in two stages. Using the efficient algorithms discussed in Section 3.2, we first draw L independent samples $\tilde{x}_j^{(\ell)}$ from a partial belief estimate combining the marginal influence function, observation potential, and incoming messages. For each of these auxiliary particles $\tilde{x}_j^{(\ell)}$, we then interpret the clique potential as a joint distribution and sample particles $x_{ji}^{(\ell)} \in \mathcal{X}_i$ from the conditional density proportional to $\psi_{ij}(x_i, x_j = \tilde{x}_j^{(\ell)})$.

Particle-based approximations are only meaningful when the corresponding BP messages $m_{ji}(x_i)$ are finitely integrable. Some models, however, contain nonnormalizable