the approximations made by loopy BP, partially justifying its application to graphs with cycles [21, 23, 24].

2.2. Nonparametric Representations

For graphical models with continuous hidden variables, analytic evaluation of the BP update integral (2) is often intractable. Instead, we represent the resulting message nonparametrically as a kernel density estimate [18]. Let $\mathcal{N}(x; \mu, \Lambda)$ denote a normalized Gaussian density with mean μ and covariance Λ, evaluated at x. An M component mixture approximation of $m_{ts}(x_s)$ takes the form

$$m_{ts}(x_s) = \sum_{i=1}^{M} w_s^{(i)} \mathcal{N}\Big(x_s; \mu_s^{(i)}, \Lambda_s\Big) \qquad (4)$$

where $w_s^{(i)}$ is the weight associated with the i^{th} kernel mean $\mu_s^{(i)}$, and Λ_s is a bandwidth or smoothing parameter. The weights are normalized so that $\sum_{i=1}^{M} w_s^{(i)} = 1$. Other kernel functions may be used [18], but in this paper we consider only mixtures of diagonal–covariance Gaussians.

In the following section, we describe stochastic methods for determining the kernel centers $\mu_s^{(i)}$ and associated weights $w_s^{(i)}$. The resulting nonparametric representation is only meaningful when the message $m_{ts}(x_s)$ is finitely integrable.[1] To guarantee this, we assume that

$$\begin{aligned} \int_{x_s} \psi_{s,t}(x_s, x_t = \bar{x})\, dx_s < \infty \qquad & \forall\, (s,t) \in \mathcal{E} \\ \int_{x_s} \psi_s(x_s, y_s = \bar{y})\, dx_s < \infty \qquad & \forall\, s \in \mathcal{V} \end{aligned} \qquad (5)$$

A simple induction argument then shows that all messages are normalizable. Heuristically, equation (5) requires each potential to be "informative," so that observing one variable constrains the likely locations of the other. In most applications, this assumption is easily satisfied by constraining all variables to a (possibly large) bounded range.

3. Nonparametric Message Updates

Conceptually, the BP update equation (2) naturally decomposes into two stages. First, the message product $\psi_t(x_t, y_t) \prod_u m_{ut}^{n-1}(x_t)$ combines information from neighboring nodes with the local evidence y_t, producing a function summarizing all available knowledge about the hidden variable x_t. We will refer to this summary as a likelihood function, even though this interpretation is only strictly correct for an appropriately factorized tree–structured graph. Second, this likelihood function is combined with the compatibility potential $\psi_{s,t}(x_s, x_t)$, and then integrated to produce likelihoods for x_s. The nonparametric belief propagation (NBP) algorithm stochastically approximates these two stages, producing consistent nonparametric representations of the messages $m_{ts}(x_s)$. Approximate marginals $\hat{p}(x_s \mid y)$ may then be determined from these messages by applying the following section's stochastic product algorithm to equation (3).

3.1. Message Products

For now, assume that the potential functions of equation (1) are weighted Gaussian mixtures (such potentials arise naturally from learning–based approaches to model identification [7]). The product of d Gaussian densities is itself Gaussian, with mean and covariance given by

$$\begin{gathered} \prod_{j=1}^{d} \mathcal{N}(x; \mu_j, \Lambda_j) \propto \mathcal{N}\big(x; \bar{\mu}, \bar{\Lambda}\big) \\ \bar{\Lambda}^{-1} = \sum_{j=1}^{d} \Lambda_j^{-1} \qquad \bar{\Lambda}^{-1}\bar{\mu} = \sum_{j=1}^{d} \Lambda_j^{-1}\mu_j \end{gathered} \qquad (6)$$

Thus, a BP update operation which multiplies d Gaussian mixtures, each containing M components, will produce a Gaussian mixture with M^d components. The weight $\bar{w}$ associated with product mixture component $\mathcal{N}\big(x; \bar{\mu}, \bar{\Lambda}\big)$ is

$$\bar{w} \propto \frac{\prod_{j=1}^{d} w_j \mathcal{N}(x; \mu_j, \Lambda_j)}{\mathcal{N}\big(x; \bar{\mu}, \bar{\Lambda}\big)} \qquad (7)$$

where $\{w_j\}_{j=1}^{d}$ are the weights associated with the input Gaussians. Note that equation (7) produces the same value for any choice of x. Also, in various special cases, such as when all input Gaussians have the same variance $\Lambda_j = \Lambda$, computationally convenient simplifications are possible.

Integration of Gaussian mixtures is straightforward, so in principle each BP message update could be performed exactly using equations (6,7). In practice, however, approximations are required to avoid exponentially large numbers of mixture components. Given d input mixtures of M Gaussians, the NBP algorithm approximates their M^d component product by drawing M independent samples.

Direct sampling from this product, achieved by explicitly calculating each of the product component weights (7), requires $\mathcal{O}(M^d)$ operations. The complexity of this sampling is combinatorial: each product component is associated with d labels $\{l_j\}_{j=1}^{d}$, where l_j identifies a kernel in the j^{th} input mixture. Although the joint distribution of the d labels is complex, the conditional distribution of any individual label l_j is simple. In particular, assuming fixed values for $\{l_k\}_{k \neq j}$, equation (7) can be used to sample from the conditional distribution of l_j in $\mathcal{O}(M)$ operations.

Since the conditional distribution of each mixture label is tractable, we may use a Gibbs sampler [8] to draw asymptotically unbiased samples from the product density. Details are provided in Algorithm 1, and illustrated in Figure 2. At each iteration, the labels $\{l_k\}_{k \neq j}$ for $d-1$ of the input mixtures are fixed, and the j^{th} label is sampled from the corresponding conditional density. The newly chosen l_j is then fixed, and another label is updated. This procedure continues for a fixed number of iterations κ; more iterations lead

[1] Probabilistically, on tree–structured graphs BP messages are likelihood functions $m_{ts}(x_s) \propto p(y = \bar{y} \mid x_s)$, not conditional densities, and are *not* necessarily integrable (e.g., when x_s and y are independent).