Given d mixtures of M Gaussians, where $\{\mu_j^{(i)}, \Lambda_j^{(i)}, w_j^{(i)}\}_{i=1}^M$ denote the parameters of the j^{th} mixture:

1. For each $j \in [1:d]$, choose a starting label $l_j \in [1:M]$ by sampling $p(l_j = i) \propto w_j^{(i)}$.
2. For each $j \in [1:d]$,
   (a) Calculate the mean μ^* and variance Λ^* of the product $\prod_{k \neq j} \mathcal{N}\left(x; \mu_k^{(l_k)}, \Lambda_k^{(l_k)}\right)$ using equation (6).
   (b) For each $i \in [1:M]$, calculate the mean $\bar{\mu}^{(i)}$ and variance $\bar{\Lambda}^{(i)}$ of $\mathcal{N}(x; \mu^*, \Lambda^*) \cdot \mathcal{N}\left(x; \mu_j^{(i)}, \Lambda_j^{(i)}\right)$. Using any convenient x, compute the weight
   $$\bar{w}^{(i)} = w_j^{(i)} \frac{\mathcal{N}\left(x; \mu_j^{(i)}, \Lambda_j^{(i)}\right) \mathcal{N}(x; \mu^*, \Lambda^*)}{\mathcal{N}\left(x; \bar{\mu}^{(i)}, \bar{\Lambda}^{(i)}\right)}$$
   (c) Sample a new label l_j according to $p(l_j = i) \propto \bar{w}^{(i)}$.
3. Repeat step 2 for κ iterations.
4. Compute the mean $\bar{\mu}$ and variance $\bar{\Lambda}$ of the product $\prod_{j=1}^d \mathcal{N}\left(x; \mu_j^{(l_j)}, \Lambda_j^{(l_j)}\right)$. Draw a sample $\hat{x} \sim \mathcal{N}(x; \bar{\mu}, \bar{\Lambda})$.

Algorithm 1. Gibbs sampler for the product of several Gaussian mixtures.

Given d mixtures of M Gaussians and an analytic function $f(x)$, follow Algorithm 1 with the following modifications:

2. After part (b), rescale each computed weight by the analytic value at the kernel center: $\bar{w}^{(i)} \leftarrow f(\bar{\mu}^{(i)}) \bar{w}^{(i)}$.
5. Assign importance weight $\hat{w} = f(\hat{x})/f(\bar{\mu})$ to the sampled particle $\hat{x}$.

Algorithm 2. Gibbs sampler for the product of several Gaussian mixtures with an analytic function.

tationally efficient "rule of thumb" heuristic [18].

The NBP message update procedure is summarized in Algorithm 3. Note that parts of this algorithm may be simplified in certain special cases. For example, Isard [9] has proposed a related generalization of particle filters in which the sampling and bandwidth selection of steps 3-4 are replaced by a deterministic kernel placement. When each pairwise potential contains only a few kernels, this modification is an excellent way to reduce biases inherent in kernel density estimates (see Section 4). However, for graphical models with multimodal potentials containing large numbers of kernels (as in Section 5), a more sophisticated propagation step (like that provided by NBP) is necessary.

4. Gaussian Graphical Models

Gaussian graphical models provide one of the few continuous distributions for which the BP algorithm may be implemented exactly [23]. For this reason, Gaussian models may be used to test the accuracy of the nonparametric approximations made by NBP. Note that we cannot hope for NBP to outperform algorithms (like Gaussian BP) designed to take advantage of the linear structure underlying Gaussian problems. Instead, our goal is to verify NBP's performance in a situation where exact comparisons are possible.

Given input messages $m_{ut}(x_t) = \{\mu_{ut}^{(i)}, \Lambda_{ut}^{(i)}, w_{ut}^{(i)}\}_{i=1}^M$ for each $u \in \Gamma(t) \setminus s$, construct an output message $m_{ts}(x_s)$ as follows:

1. Determine the marginal influence $\zeta(x_t)$ using equation (8):
   (a) If $\psi_{s,t}(x_s, x_t)$ is a Gaussian mixture, $\zeta(x_t)$ is the marginal over x_t.
   (b) For analytic $\psi_{s,t}(x_s, x_t)$, determine $\zeta(x_t)$ by symbolic or numeric integration.
2. Draw M independent samples $\{\hat{x}_t^{(i)}\}_{i=1}^M$ from the product $\zeta(x_t)\psi_t(x_t, y_t)\prod_u m_{ut}(x_t)$ using the Gibbs sampler of Algorithms 1-2.
3. For each $\{\hat{x}_t^{(i)}\}_{i=1}^M$, sample $\hat{x}_s^{(i)} \sim \psi_{s,t}(x_s, x_t = \hat{x}_t^{(i)})$:
   (a) If $\psi_{s,t}(x_s, x_t)$ is a Gaussian mixture, $\hat{x}_s^{(i)}$ is sampled from the conditional of x_s given $\hat{x}_t^{(i)}$.
   (b) For analytic $\psi_{s,t}(x_s, x_t)$, importance sampling or MCMC methods may be used as appropriate.
4. Construct $m_{ts}(x_s) = \{\mu_{ts}^{(i)}, \Lambda_{ts}^{(i)}, w_{ts}^{(i)}\}_{i=1}^M$:
   (a) Set $\mu_{ts}^{(i)} = \hat{x}_s^{(i)}$, and $w_{ts}^{(i)}$ equal to the importance weights (if any) generated in step 3.
   (b) Choose $\{\Lambda_{ts}^{(i)}\}_{i=1}^M$ using any appropriate kernel size selection method (see [18]).

Algorithm 3. NBP update of the nonparametric message $m_{ts}(x_s)$ sent from node t to node s as in equation (2).

We examine NBP's performance on a 5×5 nearest–neighbor grid (as in Figure 1) with randomly chosen inhomogeneous potentials. However, qualitatively similar results have also been observed on tree–structured graphs. To form the test model, we drew samples from the single correlated Gaussian defining each of the graph's pairwise potentials, and then formed a nonparametric density estimate based on these samples. Although the NBP algorithm could have directly used the original correlated potentials, sample–based models are a closer match for the information available in many vision applications (see Section 5).

For each node $s \in \mathcal{V}$, Gaussian BP converges to a steady–state estimate of the marginal mean μ_s and variance σ_s^2 after about 15 iterations. To evaluate NBP, we performed 15 iterations of the NBP message updates using several different particle set sizes $M \in [10, 400]$. We then found the marginal mean $\hat{\mu}_s$ and variance $\hat{\sigma}_s^2$ estimates implied by the final NBP density estimates. For each tested particle set size, the NBP comparison was repeated 100 times.