potentials that nevertheless encode important constraints. For example, the kinematic tracking and sensor localization applications considered in Section 4 both involve "repulsive" potentials, that encourage pairs of variables to *not* take similar values. In such cases, the NBP algorithm in Figure 4 instead stores the weighted particles needed to evaluate $m_{ji}(\bar{x}_i)$ at any location $\bar{x}_i$ of interest. These messages then influence subsequent iterations via importance weighting.

As illustrated in Figure 2, the BP update of message $m_{ji}(x_i)$ is most often expressed as a transformation of the incoming messages from all *other* neighboring nodes $k \in \Gamma(j)\backslash i$. From Equations 2 and 3, however, it can also be expressed as

$$m_{ji}(x_i) \propto \int_{\mathcal{X}_j} \psi_{ij}(x_i, x_j)\, \frac{q_j(x_j)}{m_{ij}(x_j)}\, dx_j. \qquad (12)$$

This transformation suggests an alternative *belief sampling* form of the NBP algorithm, in which the latest belief estimate provides a proposal distribution for auxiliary particles $\tilde{x}_j^{(\ell)} \sim q_j(x_j)$. Overcounting of $m_{ij}(x_j)$ may then be avoided via importance weights $\tilde{w}_j^{(\ell)} \propto 1/m_{ij}(\tilde{x}_j^{(\ell)})$. Computationally, belief sampling offers clear advantages: computation of new outgoing messages to d neighbors requires $\mathcal{O}(dL)$ operations, versus the $\mathcal{O}(d^2L)$ cost of the approach in Figure 4. Statistically, belief sampling also has potentially desirable properties,[26, 29] but can be less stable when the number of particles L is small.[22]

Figure 4. Nonparametric BP update for the message $m_{ji}(x_i)$ sent from node *j* to node *i*, as in Figure 2.

Given input messages $m_{kj}(x_j)$ for each $k \in \Gamma(j)\backslash i$, which may be either kernel densities $m_{kj}(x_j) = \{x_{kj}^{(\ell)}, w_{kj}^{(\ell)}, \Lambda_{kj}\}_{\ell=1}^{L}$ or analytic functions, construct an output message $m_{ji}(x_i)$ as follows:

1. Determine the marginal influence $\varphi_{ij}(x_j)$ of Equation (11).
2. Draw L independent, weighted samples from the product

$$(\tilde{x}_j^{(\ell)}, \tilde{w}_j^{(\ell)}) \sim \varphi_{ij}(x_j)\psi_j(x_j, y) \prod_{k\in\Gamma(j)\backslash i} m_{kj}(x_j).$$

Optionally resample by drawing L particles with replacement according to $\Pr\left[\tilde{x}_j^{(\ell)}\right] \propto \tilde{w}_j^{(\ell)}$, giving evenly weighted particles.

3. If $\psi_{ij}(x_i, x_j)$ is normalizeable ($\int \psi_{ij}(x_i, x_j = \bar{x})\, dx_i < \infty$ for all $\bar{x} \in \mathcal{X}_j$), construct a kernel-based output message:
   (a) For each auxiliary particle $\tilde{x}_j^{(\ell)}$, sample an outgoing particle

$$x_{ji}^{(\ell)} \sim \psi_{ij}(x_i, x_j = \tilde{x}_j^{(\ell)})$$

   Using importance sampling or MCMC methods as needed.
   (b) Set $w_{ji}^{(\ell)}$ to account for importance weights in steps 2–4(a).
   (c) Set Λ_i via some bandwidth selection method (see Silverman[42]).
4. Otherwise, treat $m_{ji}(x_i)$ as an analytic function

$$m_{ji}(x_i) \propto \sum_{\ell=1}^{L} \tilde{w}_j^{(\ell)} \psi_{ij}(x_i, \tilde{x}_j^{(\ell)})$$

parameterized by the auxiliary particles $\{\tilde{x}_j^{(\ell)}, \tilde{w}_j^{(\ell)}\}_{\ell=1}^{L}$.

4. ILLUSTRATIVE APPLICATIONS

In this section we show several illustrative examples of applications that use NBP to reason about structured collections of real-valued variables. We first show examples of kinematic tracking problems in computer vision, in which the variables represent the spatial position of parts of an object. We then show how a similar formulation can be used for collaborative self-localization and tracking in wireless sensor networks. Other applications of NBP include deformable contour tracking for medical image segmentation,[46] image denoising and super-resolution,[38] learning flexible models of dynamics and motor response in robotic control,[17] error correcting codes defined for real-valued codewords,[31, 43] and sparse signal reconstruction using compressed sensing principles.[4] NBP has also been proposed as a computational mechanism for hierarchical Bayesian information processing in the visual cortex.[32]

4.1. Visual tracking of articulated motion

Visual tracking systems use video sequences from one or more cameras to estimate object motion. Some of the most challenging tracking applications involve *articulated* objects, whose jointed motion leads to complex pose variations. For example, human motion capture is widely used in visual effects and scene understanding applications.[33] Estimates of human, and especially hand, motion are also used to build more expressive computer interfaces.[48]

To illustrate the difficulties, we consider a toy 2D object localization problem in Figure 5. The model consists of nine nodes: a central circle, and four jointed arms composed of two rectangular links. The circle node's state x_0 encodes its position and radius, while each rectangular link node's state x_i encodes its position, angle, width, and height. Each arm prefers one of the four compass directions, arms pivot around their inner joints, and geometry is loosely enforced via Gaussian pairwise potentials $\psi_{ij}(x_i, x_j)$; for details see Isard.[26]

Our goal is to find the object in a sea of clutter (white shapes in Figure 5) whose elements look exactly like components of the object. This mimics the difficulties faced in real video sequences: statistical detectors for individual object parts often falsely fire on background regions, and global geometric reasoning is needed to disambiguate them. Applied to this model, NBP's particles encode hypotheses about the pose x_i of individual object parts, while messages use geometric constraints to propagate information between parts. When all of the true object's parts are visible, NBP localizes it after a single iteration. By using Gaussian mixture potentials $\psi_i(x_i, y)$ that allow occasional outliers in observed part locations, NBP remains successful even when the central circle is missing. In this case, however, it takes more iterations to propagate information from the visible arms.

Kinematic tracking of real hand motion poses far greater challenges. Even coarse models of the hand's geometry have 26 continuous degrees of freedom: each finger's joints have four angles of rotation, and the palm