Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(12): 21147–21162.
DOI: 10.3934/mbe.2023935
Received: 09 August 2023
Revised: 18 October 2023
Accepted: 19 October 2023
Published: 27 November 2023

Perspective

Factorization in molecular modeling and belief propagation algorithms

Bochuan Du[1] **and Pu Tian**[1,2,*]

[1] School of Life Sciences, Jilin University, Changchun 130012, China
[2] School of Artificial Intelligence, Jilin University, Changchun 130012, China

* **Correspondence:** Email: tianpu@jlu.edu.cn; Tel: +86(0)43185155287.

Abstract: Factorization reduces computational complexity, and is therefore an important tool in statistical machine learning of high dimensional systems. Conventional molecular modeling, including molecular dynamics and Monte Carlo simulations of molecular systems, is a large research field based on approximate factorization of molecular interactions. Recently, the local distribution theory was proposed to factorize joint distribution of a given molecular system into trainable local distributions. Belief propagation algorithms are a family of exact factorization algorithms for (junction) trees, and are extended to approximate loopy belief propagation algorithms for graphs with loops. Despite the fact that factorization of probability distribution is the common foundation, computational research in molecular systems and machine learning studies utilizing belief propagation algorithms have been carried out independently with respective track of algorithm development. The connection and differences among these factorization algorithms are briefly presented in this perspective, with the hope to intrigue further development of factorization algorithms for physical modeling of complex molecular systems.

Keywords: molecular simulation; repetitive local sampling; neural network; local distribution theory; belief propagation; repetitive message passing; loopy belief propagation; gaussian belief propagation

1. Introduction

It is well acknowledged that high dimensional problems are difficult to analyze and predict. Multimedia data (e.g., pictures, videos) and molecular systems (e.g., biomolecules, materials, nano-systems) are typical examples. Since analytical solutions are either extremely challenging to achieve or non-existant for accurate and detailed description of probability distributions in high dimensional spaces, various numerical approximations are essential. The two most important and distinct paths are direct utility of dimensionality reduction algorithms [1–6] and factorizations [7–9], and both are extensive fields. The former path is to transform a n-dimensional probability distribution

$P_n(x_1, x_2, \ldots, x_n)$, usually approximately, into a m-dimensional $P_m(x_1, x_2, \ldots, x_m)$ with $n \gg m$, while the latter path is to factorize $P_n(x_1, x_2, \ldots, x_n)$, either exactly or approximately, into a product of K low dimensional terms $\prod_{i=1}^{K} P_{m_i}(x_{i1}, x_{i2}, \ldots, x_{im_i})$, with the union of variables in all K terms being $(x_1, x_2, \ldots, x_n)$. In this perspective, we focus on different factorizations utilized in molecular simulations and belief propagation algorithms, and hope to intrigue interests in development of more accurate and efficient approximations in relevant applications. In a probabilistic description of multivariate systems, there are three important types of problems. The first type is to find the most probable (set of) configuration(s) at a given resolution of representation, and more generally the global landscape of wells with significant probability weights, which is termed free energy landscapes in molecular systems [10–12] and multimodal learning in AI [13]. A highly related type of problems are to locate transition paths connecting various wells on the global landscape. Understanding such transition paths is of critical importance in accelerating sampling of other wells when starting from an arbitrary well of a landscape. Significant efforts have been invested in finding transition paths in molecular simulations [14–16]. The second type is to focus only on a subset of variables (or a single one in the extreme case), and the process of computing distributions of a subset of variables from the global joint distribution is termed marginalization. For example, marginalizing a two dimensional distribution $P_2(x, y)$ with both $x, y \in (0, 1)$ to the variable x is accomplished by the following two summations: $P_1(x = 0) = P_2(x = 0, y = 0) + P_2(x = 0, y = 1)$ and $P_1(x = 1) = P_2(x = 1, y = 0) + P_2(x = 1, y = 1)$. There are 2^2 terms to be calculated, and for a joint distribution with n variables each has $|X|$ discrete values, $|X|^n$ terms need to be calculated, indicating the exponential computational complexity of marginalization. The third type is the parameterization of model parameters, including parameterizing force fields for molecular modeling and factor/node potentials for belief propagation. After introduction of typical factorization schemes in molecular modeling and belief propagation, we discuss strengths of different factorization in various type of problems. The organization of this perspective is summarized below. Following this introduction, Section 2 gives a brief introduction to exact belief propagation algorithms. Section 3 is on junction trees and approximate loopy belief propagations. Sections 4 and 5 describe factorization in regular molecular simulations and the local distribution theory. Section 6 provides discussions on the potential synergy and distinctions among these factorizations, and deep learning. Finally conclusions are presented.

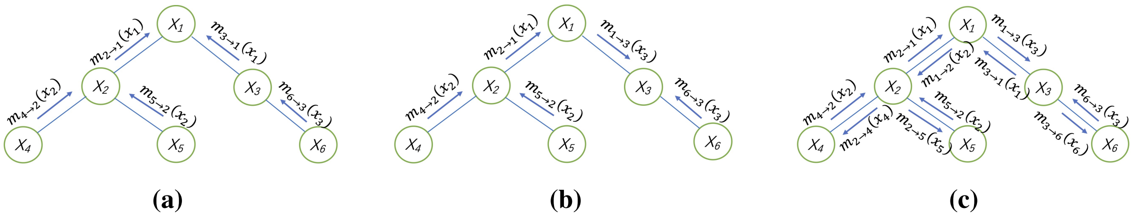
2. Exact belief propagation algorithms for trees

As mentioned in the introduction, marginalization of high dimensional joint distributions is difficult, and in many cases intractable. Fortunately, factorization significantly reduces this cost whenever it is possible to be performed exactly, or approximately with sufficient accuracy. A n-variable system $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ forms an undirected graph with each variable being a node and an edge between all pairs of interacting variables. In graph theory [17], a clique is a set of nodes with an edge between each pair of nodes in this set, and a maximal clique is one that will be destroyed by an addition of a new node into this set. For a graph $\mathcal{G}$ with a set of maximal cliques C [17]:

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{c \in C} f(x_c) \tag{2.1}$$

with $f(x_c)$ being the potential over the variable set c and Z being the partition function $\sum_{\boldsymbol{x}} \prod_{c \in C} f(x_c)$, which is summation of the product of all clique potentials over all possible configurations. In a tree, the maximal clique set is the edge set $\mathcal{E}$. For the specific example with 6 nodes shown in Figure 1, Eq (2.1) may be written as the following:

$$p(\boldsymbol{x}) \propto f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) f_{2,4}(x_2, x_4) f_{2,5}(x_2, x_5) f_{3,6}(x_3, x_6). \tag{2.2}$$



(a) **(b)** **(c)**

Figure 1. A schematic illustration of the marginalization in Eqs (2.3), (2.5) and (2.8). Blue arrows indicate direction of message passing, and corresponding messages are shown beside each arrow. (a) Illustration of the process presented in Eq (2.3) for the marginal $p_1(x_1)$; (b) Illustration of the process presented in Eq (2.5) for the marginal $p_3(x_3)$ and (c) illustration of the belief propagation presented in Eq (2.8).

Given this factorization, the computational cost bound of marginalization goes from $|x|^6$ to $5 \times |x|^2$. Specifically, the marginal probability $p_1(x_1)$ may be calculated in the marginalization process shown below (and is illustrated in Figure 1(a)) :

$$\begin{aligned}
p_1(x_1) &\propto \sum_{x_2,x_3,x_4,x_5,x_6} f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) f_{2,4}(x_2, x_4) f_{2,5}(x_2, x_5) f_{3,6}(x_3, x_6) \\
&= \sum_{x_2,x_3,x_4,x_5} f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) f_{2,4}(x_2, x_4) f_{2,5}(x_2, x_5) \underbrace{\sum_{x_6} f_{3,6}(x_3, x_6)}_{\triangleq m_{6\to3}(x_3)} \\
&= \sum_{x_2,x_3,x_4} f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) f_{2,4}(x_2, x_4) m_{6\to3}(x_3) \underbrace{\sum_{x_5} f_{2,5}(x_2, x_5)}_{\triangleq m_{5\to2}(x_2)} \\
&= \sum_{x_2,x_3} f_{1,2}(x_1, x_2) f_{1,3}(x_1, x_3) m_{6\to3}(x_3) m_{5\to2}(x_2) \underbrace{\sum_{x_4} f_{2,4}(x_2, x_4)}_{\triangleq m_{4\to2}(x_2)} \\
&= \sum_{x_2} f_{1,2}(x_1, x_2) m_{4\to2}(x_2) m_{5\to2}(x_2) \underbrace{\sum_{x_3} f_{1,3}(x_1, x_3) m_{6\to3}(x_3)}_{\triangleq m_{3\to1}(x_1)} \\
&= m_{3\to1}(x_1) \underbrace{\sum_{x_2} f_{1,2}(x_1, x_2) m_{4\to2}(x_2) m_{5\to2}(x_2)}_{\triangleq m_{2\to1}(x_1)} \\
&= m_{3\to1}(x_1) m_{2\to1}(x_1),
\end{aligned} \tag{2.3}$$

and $p_1(x_1)$ may be subsequently obtained with the normalization shown below:

$$p_1(x_1) = \frac{m_{2\to1}(x_1)m_{3\to1}(x_1)}{\sum_{x'\in\mathcal{X}} m_{2\to1}(x')m_{3\to1}(x')} \tag{2.4}$$

Similarly, the marginal probability $p_3(x_3)$ may be obtained as shown below and is illustrated in Figure 1(b):

$$\begin{aligned}
p_1(x_3) &\propto \sum_{x_1,x_2,x_4,x_5,x_6} f_{1,2}(x_1,x_2)f_{1,3}(x_1,x_3)f_{2,4}(x_2,x_4)f_{2,5}(x_2,x_5)f_{3,6}(x_3,x_6)\\
&= \sum_{x_1,x_2,x_4,x_5} f_{1,2}(x_1,x_2)f_{1,3}(x_1,x_3)f_{2,4}(x_2,x_4)f_{2,5}(x_2,x_5)\underbrace{\sum_{x_6} f_{3,6}(x_3,x_6)}_{\triangleq m_{6\to3}(x_3)}\\
&= \sum_{x_1,x_2,x_4} f_{1,2}(x_1,x_2)f_{1,3}(x_1,x_3)f_{2,4}(x_2,x_4)m_{6\to3}(x_3)\underbrace{\sum_{x_5} f_{2,5}(x_2,x_5)}_{\triangleq m_{5\to2}(x_2)}\\
&= \sum_{x_1,x_2} f_{1,2}(x_1,x_2)f_{1,3}(x_1,x_3)m_{6\to3}(x_3)m_{5\to2}(x_2)\underbrace{\sum_{x_4} f_{2,4}(x_2,x_4)}_{\triangleq m_{4\to2}(x_2)}\\
&= \sum_{x_1} f_{1,3}(x_1,x_3)m_{6\to3}(x_3)\underbrace{\sum_{x_2} f_{1,2}(x_1,x_2)m_{4\to2}(x_2)m_{5\to2}(x_2)}_{\triangleq m_{2\to1}(x_1)}\\
&= m_{6\to3}(x_3)\underbrace{\sum_{x_1} f_{1,3}(x_1,x_3)m_{2\to1}(x_1)}_{\triangleq m_{1\to3}(x_3)}\\
&= m_{1\to3}(x_3)m_{6\to3}(x_3),
\end{aligned} \tag{2.5}$$

and $p_1(x_3)$ may be subsequently obtained with the normalization shown below:

$$p_1(x_3) = \frac{m_{1\to3}(x_3)m_{6\to3}(x_3)}{\sum_{x'\in\mathcal{X}} m_{1\to3}(x')m_{1\to3}(x')} \tag{2.6}$$

In the calculation of both marginals $p_1(x_1)$ and $p_1(x_3)$, the following messages are computed:

$$\begin{aligned}
m_{6\to3}(x_3) &= \sum_{x_6} f_{3,6}(x_3,x_6),\\
m_{5\to2}(x_2) &= \sum_{x_5} f_{2,5}(x_2,x_5),\\
m_{4\to2}(x_2) &= \sum_{x_4} f_{2,4}(x_2,x_4),\\
m_{2\to1}(x_1) &= \sum_{x_2} f_{1,2}(x_1,x_2)m_{4\to2}(x_2)m_{5\to2}(x_2).
\end{aligned} \tag{2.7}$$

Computing the marginal for each node in a tree takes $n-1$ messages, and consequently $n(n-1)$ messages are necessary for n nodes when computed respectively (see Figure 1(a),(b)). However, instead of computing marginals for each variable separately, $2(n-1)$ messages over $n-1$ edges of an n-node tree maybe computed, such that all marginals maybe obtained without repetitive message calculation as illustrated in Figure 1(c) and demonstrated in the equation below.

This message passing is the sum-product, or the belief propagation algorithm for a general tree structure with pair potentials [7–9].

$$m_{i\to j}(x_j) = \sum_{x_i} f_{i,j}(x_i, x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i) \tag{2.8}$$

the message from i to j ($m_{i\to j}$) is sent once i receives messages from all of its neighbors $N(i)$ except the node j. After two sweeps of message passing, the marginal for variable i is obtained by:

$$p_i(x_i) \propto \prod_{j\in N(i)} m_{j\to i}(x_i). \tag{2.9}$$

When a unary potential is included for each node, we have:

$$m_{i\to j}(x_j) = \sum_{x_i} g_i(x_i) f_{i,j}(x_i, x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i), \tag{2.10}$$

$$p_i(x_i) \propto g_i(x_i) \prod_{j\in N(i)} m_{j\to i}(x_i). \tag{2.11}$$

One may easily compute the maximum probability configuration $\boldsymbol{x}^*$ by replacing the sum operations in the Eq (2.8) or (2.10) with max operations, and by bookkeeping the corresponding state of each variable in each maximum operation for backtracking. This results in the max-product algorithm for calculating the maximum probability configuration.

3. Junction tree algorithm, loopy and gaussian belief propagation

In graphs where one or more loop(s) exist(s), the belief propagation algorithm ensures neither convergence nor correct calculation of marginals, and the same is true regarding the max product algorithm. Exact belief propagation maybe achieved for a general graph by constructing a chordal graph through triangulation [9], which involves adding edges to a graph such that no chordless loop of the length greater than 4 exists (see Figure 2 for a schematic illustration).

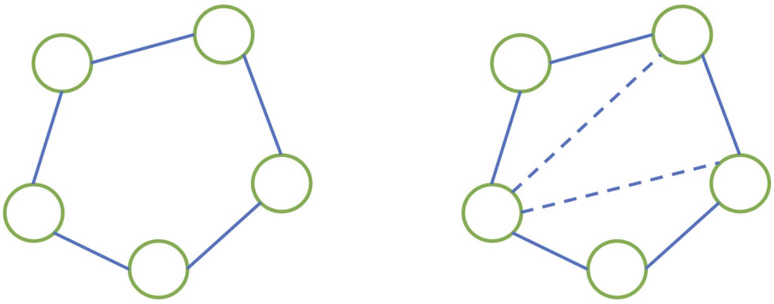


Figure 2. A graph with a five-edge loop (left) is triangulated with addition of two edges (dashed blue lines) and become chordal (right).

Maximal cliques in a triangulated graph can be joined to form a junction tree. The most important property of such a tree is the junction tree property, which ensures that if a node x_i is in two maximal cliques c_j and c_k, then it must be a member of every maximal clique that is on the path from c_j to c_k [18,19]. The computational complexity of a junction tree is determined by the maximum tree width, which is the size of the largest maximal clique. Many different chordal graphs with different tree widths may be generated from a given graph with loops, and a number of algorithms are available to perform triangulation and construct chordal graphs [20–23]. However, no existing algorithm is available to ensure finding the best triangulation scheme, which is expected to generate one or more chordal graphs with the smallest possible tree width.

Even without assurance that exact marginals would be obtained by the belief propagation algorithm for graphs with loops, we may carry out such message passing computations anyway, and the algorithm is accordingly termed loopy belief propagation (LBP). Turbo code [24] is an outstanding example of utilizing LBP for telecommunication coding. From the parallel version [25] of the belief propagation algorithm (see Eq (3.1)) shown below:

$$m_{i\to j}^{t+1} \propto \sum_{x_i} g_i(x_i) f_{i,j}(x_i, x_j) \prod_{k\in N(i)\backslash j} m_{k\to i}^{t}(x_i). \tag{3.1}$$

Messages are usually initialized to be 1 and normalized at each time step. If a loopy belief propagation did converge, then Eq (3.1) suggests that the converged message would be a fixed point of the messaging updating function. For a general graph with pair potentials, the fixed point of belief propagation message updates are some local extrema of the Bethe variational problem [26]. The details of both the Bethe approximation and other treatments of loopy belief propagation are well beyond the scope this perspective, and interested readers are encouraged to find relevant information elsewhere [27,28].

Belief propagation was initially developed for discrete variables [7]. The summation in message computation becomes integration for continuous variables. Such integration is usually intractable for arbitrary distributions. Fortunately, gaussian distributions may be both analytically integrated and resulting in gaussians. This property may be utilized in belief propagation to treat continuous variables as gaussians. In cases where such direct approximation with gaussians is not sufficient, linearization with the first order Taylor expansion may be utilized to improve the accuracy [29].

4. Factorization in conventional molecular modeling

Conventional molecular simulation is a mature methodology with a wide variety of tools. These methods are routinely utilized in many research fields, including but not limited to chemistry, biology and materials science with more than 30,000 relevant publications each year [30]. An essential component of the foundation for molecular simulation is a parameter set termed force fields, which represents interaction energies between/among various molecular degrees of freedom as a function of relevant coordinates. Conventional molecular simulation is deemed by many of its practitioners as an independent research field from, while being in fact a well defined form of, statistical machine learning. The force fields parameterization is the learning stage and the simulation and sampling constitute the inference process.

The total energy of a given molecular configuration $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ for a simple Lennard-Jones

(LJ) system is defined as [31]:

$$E_{total}(\boldsymbol{x}) = \sum_{i-j,\, r_{ij} \le r_{cut}} 4\epsilon \left(\left(\frac{r_{ij}}{\sigma} \right)^{12} - \left(\frac{r_{ij}}{\sigma} \right)^{6} \right). \tag{4.1}$$

Such LJ terms are usually utilized to describe various Van de Waals interactions between neutral atoms/particles with respective parameters ϵ and σ. Apparently, this equation is essentially a factorization:

$$\begin{aligned} p(\boldsymbol{x}) \propto Exp(-E_{total}) &= \prod_{i-j,\, r_{ij} \le r_{cut}} Exp\left(-f_{i-j,\, r_{ij} \le r_{cut}}(r_{ij}(x_i, x_j))\right), \\ f_{i-j} &= 4\epsilon \left(\left(\frac{r_{ij}}{\sigma} \right)^{12} - \left(\frac{r_{ij}}{\sigma} \right)^{6} \right). \end{aligned} \tag{4.2}$$

with $\boldsymbol{x}$ representing the set of coordinates for all comprising atoms/particles of a specific configuration. Let us denote each particle as a node, and the corresponding interaction between a pair of particles as an edge. Each factor represents an interacting pair of particles. This factorization has a very similar form as that in the belief propagation in a tree (Eq (2.1)). However, there are three major differences. Firstly, in Eq (2.1), the graph is static with fixed number of edges. However, for a graph formed by LJ particles, edges are dynamic in two aspects. One is that as the system propagates with time, edges break as some pairs of particles drift from within to beyond cutoff and new ones form as others drift from beyond to within. The other is that for pairs within cutoff, their interactions change as a function of distance. Secondly, nodes in Eq (2.1) have discrete states, while properties of LJ particles (e.g., position and momentum) are continuous. Thirdly, in Eq (2.1), each factor is over a maximal clique comprising a pair of connected nodes. For LJ particles, when the cutoff for direct interaction is taken as 3σ, all particles within a sphere with the cutoff as its diameter are certainly within a single clique (since they all interacts) as shown in Figure 3(a). The number of particles ranges from a few to more than a dozen. LJ potential is factorized over pairs of particles instead of maximal cliques. Such factorization is apparently an approximation, and the resulting error is not easy to quantify for arbitrary LJ interactions. Apparently, the exact number of particles in each maximal clique can be different as local packing of LJ particles continuously changes over space and time. In a graph, two maximal cliques share one or more nodes are effectively connected. For densely packed molecular systems that may be represented as LJ particles, each maximal clique usually overlaps with multiple neighboring maximal cliques. Therefore, it is essentially a sure thing that most maximal cliques participate in loops. Constructing junction trees on such systems certainly results in huge tree width. Exact inference is consequently prohibitive, and the utility of LBP is necessary.

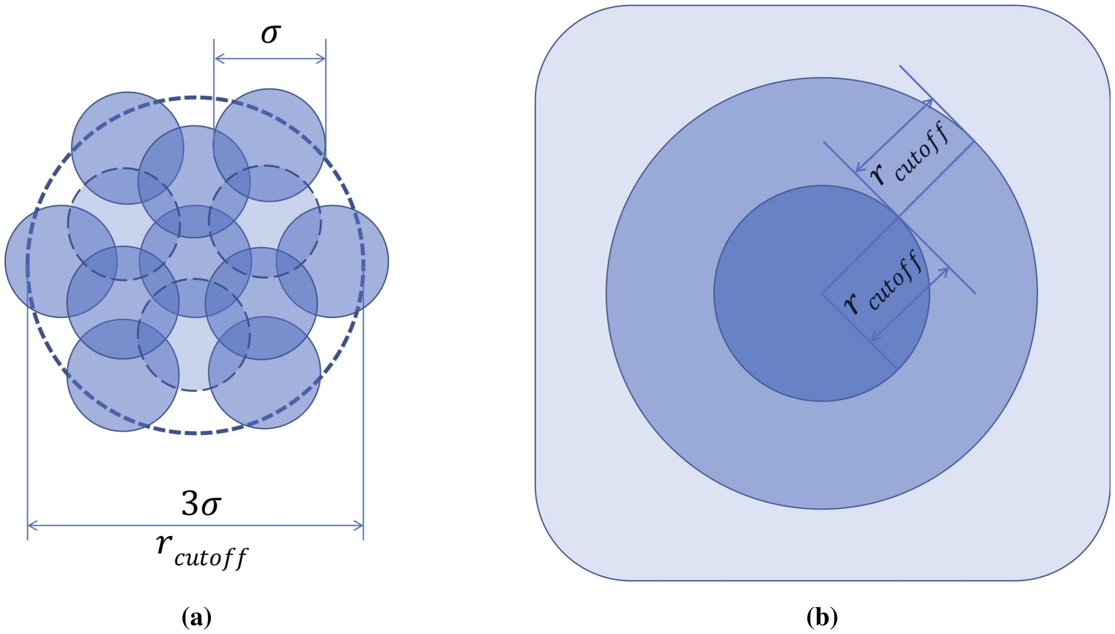


Figure 3. (a) An example of 13 Lennard-Jones particles of diameter σ enclosed within a sphere of diameter 3σ (r_{cutoff}). Three particles partially covered by other particles are shown with dashed circles. These 13 particles form a clique with 78 edges, all should be represented by a single factor/potential for exact inference while in conventional molecular simulation, 78 edges correspond to 78 factors. (b) A schematic illustration of LDT theory in which each spherical space of radius r_{cutoff} is represented as a single factor, a central particle experiences mediated interactions from particles beyond r_{cutoff} but within $2r_{cutoff}$, only interactions further than $2r_{cutoff}$ are treated as long-range interactions.

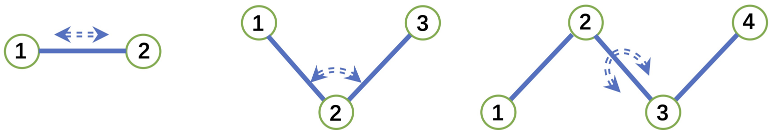


Figure 4. Schematic representation of bonded interactions. Left: A bond between the atom 1 and the atom 2, Center: A bend formed by the bond $1--2$ and the bond $2--3$, Right: A dihedral formed by the plane $1--2--3$ and the plane $2--3--4$.

For more general molecular systems, energy contributions are more diverse [32–34] with addition of various bonded (including bonds, bends and dihedrals) (see Figure 4) and electrostatic interaction terms between ions or partially charged polar atoms, and are usually calculated as shown below:

$$E_{total} = E_{bond} + E_{bend} + E_{dihedral} + E_{VdW} + E_{Electro} \tag{4.3}$$

$$= \sum_{bond_i} E_{bond_i} + \sum_{bend_j} E_{bend_j} + \sum_{dihedral_k} E_{dihedral_k} + \sum_{VdW_l} E_{VdW_l} + \sum_{ES_{esi}} E_{ES_{esi}}. \tag{4.4}$$

Here i, j, k, l and esi are indices for bonds, bends, dihedrals, Van de Waals and electrostatic interactions respectively. Taking exponentiation of this equation results in:

$$\begin{aligned} Exp(-E_{total}) &= \prod_{bond_i} Exp(-E_{bond_i}) \prod_{bend_j} Exp(-E_{bend_j}) \\ &\times \prod_{dihedral_k} Exp(-E_{dihedral_k}) \prod_{VdW_l} Exp(-E_{VdW_l}) \prod_{ES_{esi}} Exp(-E_{ES_{esi}}) \end{aligned} \tag{4.5}$$

Bonded terms are fundamentally local in space. Van der Waals interactions are usually modeled by LJ potentials. Similar to simple LJ systems, interactions in maximal cliques are approximately represented by assuming independent local interactions involving single (VdW and bonds), double (bends) or triple (dihedrals) edge(s). There is also challenge of addressing continuous variables with non-gaussian distributions and dynamic edges. More importantly, electrostatic interactions are long-range. Should such interactions to be treated exactly, then factorization is not possible and a molecular system needs to be treated as one single maximal clique. The exponential complexity with respect to the size of the maximal clique renders exact treatment intractable in many realistic molecular systems, so treating long range interactions as being independent from both local ones and each other is an essential approximation in computational molecular science. In Eq (4.5), all interaction terms are treated as being independent regardless of interaction range in space. Fortunately, such approximations have been widely tested in molecular simulations with great successes [35]. In reality, naive pairwise calculation of electrostatic interaction diverges and usually some form of Ewald summation is utilized to ensure convergence [36, 37].

5. Factorization in the local distribution theory

Just like the repetitive local computation in the calculation of marginals as demonstrated in Eqs (2.3), (2.5) and (2.7), ubiquitous repetitive local sampling (RLS) exists in molecular simulations [30, 38, 39]. From a graph perspective, such RLS is fundamentally associated with the intrinsic dynamic property of graphs describing molecular systems. The local distribution theory (LDT) was developed to address RLS and tremendously increases computational efficiency without reducing accuracy/resolution [30, 39]. As shown in Eq (5.1), LDT is another approximate factorization scheme for facilitating statistical machine learning of dynamic graphs in high dimensional space.

$$\begin{aligned} P(\Phi, \boldsymbol{x}) &= Q(\Phi, R) \\ &= \frac{Q(\Phi, R)}{\prod_{i=1}^{m} q(\Phi, r_i)} \prod_{i=1}^{m} q(\Phi, r_i) \\ &\approx \prod_{i=1}^{m} q(\Phi, r_i) exp(-\sum F_{MED}(\Phi, R)) exp(-\sum F_{LR}(\Phi, R)) \end{aligned} \tag{5.1}$$

Here, $\Phi = (\phi_1, \phi_2, \ldots, \phi_k)$ is relevant thermodynamic and environmental variables, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ are molecular coordinates. $R = (r_1, r_2, \ldots, r_m)$ ($m \leq n,\ m = n$ is preferred) are dynamic local regions, each of which represent a dynamic collection of coordinates ($r_i = (x_{i1}, x_{i2}, \ldots, x_{il})$) for molecular degrees of freedom. A translation, rotation and permutation invariant coordinate transformation is necessary to feed r_i into neural networks or any other powerful nonlinear fitting machinery for local free energy computation. Effectively, taking a local region r_i and calculating its local free energy (high dimensional potential of mean force) is similar to taking two particles in conventional molecular simulation and calculating their two-body potential of mean force. $\prod_{i=1}^{m} q(\Phi, r_i)$ is the product of local distributions and the global correlation factor $\frac{Q(\Phi,R)}{\prod_{i=1}^{m} q(\Phi,r_i)}$ is approximately factorized into mediated ($\sum F_{MED}(\Phi, R)$) and long range ($\sum F_{LR}(\Phi, R)$) contributions. The learning stage of LDT is to train local distributions, which cache statistical significance of various molecular configurations within local spaces defined by a preset cutoff from the origin of a local coordinates originating from a given molecular DOF. Such local distributions, defined by trainable neural network parameters with selected neural network architectures, store both energetic and entropic contributions from extensive local sampling data of either experimental or computational origin. By caching and repetitively utilizing such local distributions, LDT eliminates RLS to a great extent, and therefore saves computational resources in sampling dynamic graphs. In the inference stage, local distributions are assembled under sampling constraints due to mediated interactions. Direct long range interactions are treated as independent from local ones (Eq (5.1)), and selection of specific method does not impact local computation. Searching for the best methodology to match the LDT scheme is an important issue to be tackled in the future. Hierarchical definition of "local" with different spatial coverage might be an efficient path to be explored for large systems with hierarchical dynamics spanning multiple time scales.

In conventional molecular simulations, pairs of interactions (or edges in a graph representing a molecular system) within cutoff are included as independent factors (Figure 3(a)). As illustrated in Figure 3(b), both local interactions confined within cutoff spherical space and mediated interactions are jointly considered in LDT. Independence assumptions are therefore significantly weaker and accordingly, improvement of accuracy is expected with tremendously increased efficiency due to substitution of RLS by local distributions. Therefore, LDT has potential to address the tradeoff dilemma between accuracy and efficiency in molecular simulations. However, this does come with a significant one-time price to pay for accurate fitting of local distributions, which is fundamentally a density estimation task in high dimensional system, an open problem with many feasible approaches and respective limitation [40,41].

6. Distinction and potential synergy among different factorization schemes

One fundamental difference between factorization in the belief propagation algorithm and that in the molecular modeling of physical systems is that the former deals with a static graph (V, E) while the latter addresses a dynamic one $(V, E_{dynamic})$. The variation of edges in molecular systems is continuous, but may be simplified as binary in the case of a protein contact map [42]. Specifically, as the distance between two atoms/particles changes over time, so the corresponding interaction potential. In the belief propagation, an isolated maximal clique has a fixed factor potential, the computation is to figure out how interactions with neighboring cliques influence the distribution of internal states of nodes. However, we assume that forces exerted on each atom/particle does not change within a time step of molecular dynamics simulations, we may similarly deem a molecular system in a sufficiently short time period as a static graph to apply belief propagation algorithms.

When the exact belief propagation formulation is possible, it provides two great advantages. One is the rapid determination of the most probable configuration, and the other is the explicit normalization of local probability distributions, which is a mechanism missing in the molecular modeling methodology tool box, and is potentially helpful in keeping errors under control if successfully applied in molecular systems. To realize such a goal, the first step is to define nodes with internal states. For example, in a full quantum mechanical treatment, one may define each atom as a node with rich internal electronic states. For proteins, residues may be defined as distinct nodes with internal rotameric states, which may be approximately represented either discretely or continuously. Apparently, definition of nodes is flexible for a given type of molecular systems and requires significant effort to explore. For a given molecular system with specific definition of nodes, we may perform loopy belief propagation to search for the most probable global configuration (or a number of configurations with the largest statistical weights) of internal node states, followed by sampling of different edge configuration with LDT or some other properly selected sampling strategy. Three important factors for the application of the belief propagation in molecular systems are definition of nodes, selection/development of approximate loopy belief propagation algorithms and selection of sampling strategy. It is likely that each may impact and influenced by the other two factors. Exploring different combinations of these algorithms are likely fertile grounds for developing next generation molecular modeling methodologies. An additional advantage of formulating a molecular system with the belief propagation is that marginalization, which is of critical importance in docking and design of key spots in biomolecular interactions, becomes straight forward.

While exact belief propagation need only one full sweep (to and from each node) of message passing, LBP requires iteration, and therefore the speed of convergence becomes a practical challenge. For the first type of problems stated in the introduction (to find maximum probability configuration in particular and to map out full landscape in general), the process is essentially some form of walking over the probability landscape and settling down at the most easy-to-reach local well from a start configuration. To map out the global landscape, one has to design ways to escape from any given local well and reach other potentially deeper or shallower wells. Unfortunately, no systematic theory is available for belief propagation algorithms to achieve this goal. The rich history of enhanced sampling of free energy landscapes [11, 43–48], coarse graining [49, 50] (which is effectively smoothing of free energy landscape), and transition path sampling [14–16] (which focus on finding transition paths between/among different free energy wells) in molecular systems may provide hints to inventing new algorithms for the accelerated convergence of LBP.

Belief propagation algorithms reduces repetitive local computation in the redistribution of marginals for internal node states by surrounding connected nodes. LDT reduces the repetitive local sampling of dynamic edges by utilizing pre-trained neural networks which implicitly holds the probability density of a given configuration. Deep learning has found extensive applications in molecular simulations [51, 52], especially in the construction of deep potentials (parameterization of force fields) [53–56]. It is likely that there are more potential ways of combining different factorization schemes in molecular simulations, belief propagation and deep learning to be invented. For example, when compared with other conventional dimensionality reduction algorithms, deep learning has the advantage of being much more flexible, and such dimensionality reduction might be combined with molecular simulation and belief propagation to construct novel approximate multi-scale solutions.

Matrix factorization is widely utilized in transforming input for deep learning [57–59]. It is important to note that such factorization is a global behavior in the probability distribution space. This is in strong contrast to decomposing a global distribution into local ones in the belief propagation and molecular simulations.

7. Conclusions and prospects

Belief propagation is a family of algorithms in a large class of factorization methodology called probabilistic graphical models with focus on static graphs. These algorithms are mainly developed by informational scientists and are applied in coding, communication and machine learning. Variants of factorization in molecular simulations, including the local distribution theory, are fundamentally factorization focusing on dynamic graphs representing complex molecular systems, and are mainly developed by computational chemists and physicists. Both are extensive research fields with very rich contents and numerous important contributions from many scientists. The most outstanding difference is that factorization in belief propagation has an exact solutions for simple and junction trees but lacks methodology development in enhanced sampling, while factorization in molecular simulations has rich history in accelerating sampling of free energy landscapes but no exact solution. Comprehensive survey of these two fields is apparently well beyond the scope of present work. This perspective is intended to emphasize this discipline gap, and intrigue interest for developing next generation algorithms that assimilates the best of both fields.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

We appreciate the Interdisciplinary Integration and Innovation Project of JLU with grant number JLUXKJC2021ZZ05. We thank Professor Zhonghan Hu for comments on the local distribution theory when the manuscript was in preparation.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. I. T. Jolliffe, *Principal Component Analysis*, Springer, New York, 2002. https://doi.org/10.1007/b98835
2. T. F. Cox, M. A. A. Cox, *Multidimensional Scaling*, 2nd eddition, Chapman and Hall/CRC, New York, 2000. https://doi.org/10.1201/9781420036121
3. J. B. Tenenbaum, V. de Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science*, **290** (2000), 2319–2323. https://doi.org/10.1126/science.290.5500.2319

4. S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, **290** (2000), 2323–2326. https://doi.org/10.1126/science.290.5500.2323

5. R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, et al., Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods. *Proc. Natl. Acad. Sci.*, **102** (2005), 7432–7437. https://doi.org/10.1073/pnas.0500896102

6. M. Ceriotti, G. A. Tribello, M. Parrinello, Simplifying the representation of complex free-energy landscapes using sketch-map, *Proc. Natl. Acad. Sci.*, **108** (2011), 13023–13028. https://doi.org/10.1073/pnas.1108486108

7. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Morgan Kaufmann, San Mateo, CA, 1988.

8. F. R. Kschischang, B. J. Frey, H. A. Loeliger, Factor graphs and the sum-product algorithm, *IEEE Trans. Inf. Theory*, **47** (2001), 498–519. https://doi.org/10.1109/18.910572

9. D. Koller, N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, Cambridge, MA, 2009.

10. H. Fu, X. Shao, W. Cai, C. Chipot, Taming rugged free energy landscapes using an average force, *Acc. Chem. Res.*, **52** (2019), 3254–3264. https://doi.org/10.1021/acs.accounts.9b00473

11. O. Valsson, P. Tiwary, M. Parrinello, Enhancing important fluctuations: Rare events and metadynamics from a conceptual viewpoint, *Annu. Rev. Phys. Chem.*, **67** (2016), 159–184. https://doi.org/10.1146/annurev-physchem-040215-112229

12. G. Bussi, A. Laio, Using metadynamics to explore complex free-energy landscapes, *Nat. Rev. Phys.*, **2** (2020), 200–212. https://doi.org/10.1038/s42254-020-0153-0

13. D. Ramachandram, G. W. Taylor, Deep multimodal learning: A survey on recent advances and trends, *IEEE Signal Process Mag.*, **34** (2017), 96–108. https://doi.org/10.1109/MSP.2017.2738401

14. C. Dellago, P. G. Bolhuis, P. L. Geissler, Transition path sampling, in *Advances in Chemical Physics*, John Wiley & Sons, Ltd, (2002), 1–78. https://doi.org/10.1002/0471231509.ch1

15. J. Rogal, P. G. Bolhuis, Multiple state transition path sampling, *J. Chem. Phys.*, **129** (2008), 224107. https://doi.org/10.1063/1.3029696

16. P. Buijsman, P. G. Bolhuis, Transition path sampling for non-equilibrium dynamics without predefined reaction coordinates, *J. Chem. Phys.*, **152** (2020), 044108. https://doi.org/10.1063/1.5130760

17. R. J. Trudeau, *Introduction to Graph Theory*, Dover Publications, New York, 1993.

18. S. L. Lauritzen, D. J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, *J. Roy. Statist. Soc. Ser. B*, **50** (1988), 157–194. https://doi.org/10.1111/j.2517-6161.1988.tb01721.x

19. F. V. Jensen, S. L. Lauritzen, K. G. Olesen, Bayesian updating in causal probabilistic networks by local computations, *Comput. Statist. Quart.*, **5** (1990), 269–282.

20. V. Gogate, R. Dechter, A complete anytime algorithm for treewidth, in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, Arlington, Virginia, (2004), 201–208.

21. E. H. Bachoore, H. L Bodlaender, A branch and bound algorithm for exact, upper, and lower bounds on treewidth, in *Algorithmic Aspects in Information and Management, AAIM 2006, Lecture Notes in Computer Science*, Springer, (2006), 255–266. https://doi.org/10.1007/11775096_24

22. T. J. Ottosen, J. Vomlel, All roads lead to rome–new search methods for the optimal triangulation problem, *Int. J. Approximate Reasoning*, **53** (2012), 1350–1366. https://doi.org/10.1016/j.ijar.2012.06.006

23. C. Li, M. Ueno, An extended depth-first search algorithm for optimal triangulation of bayesian networks, *Int. J. Approximate Reasoning*, **80** (2017), 294–312. https://doi.org/10.1016/j.ijar.2016.09.012

24. C. Berrou, A. Glavieux, *Turbo Codes*, John Wiley & Sons, Ltd, New York, 2003. https://doi.org/10.1002/0471219282.eot346

25. J. Gonzalez, Y. Low, C. Guestrin, Parallel splash belief propagation, *J. Mach. Learn. Res.*, **1** (2009), 1–48.

26. J. S. Yedidia, W. T. Freeman, Y. Weiss, Generalized belief propagation, in *NIPS'00: Proceedings of the 13th International Conference on Neural Information Processing System*, (2000), 668–674.

27. M. P. Kumar, P. H. S. Torr, Fast memory-efficient generalized belief propagation, in *Computer Vision–ECCV 2006, Lecture Notes in Computer Science*, Springer, (2006), 451–463. https://doi.org/10.1007/11744085_35

28. S. Y. Chen, H. Tong, Z. Wang, S. Liu, M. Li, B. Zhang, Improved generalized belief propagation for vision processing, *Math. Probl. Eng.*, **2011** (2011). https://doi.org/10.1155/2011/416963

29. J. Ortiz, T. Evans, A. J. Davison, A visual introduction to gaussian belief propagation, *arXiv preprint*, (2021), arXiv:2107.02308. https://doi.org/10.48550/arXiv.2107.02308

30. P. Tian, The repetitive local sampling and the local distribution theory, *WIREs Comput. Mol. Sci.*, **12** (2021), e1588. https://doi.org/10.1002/wcms.1588

31. X. Wang, S. Ramirez-Hinestrosa, J. Dobnikar, D. Frenkel, The lennard-jones potential: When (not) to use it, *Phys. Chem. Chem. Phys.*, **22** (2020), 10624–10633. https://doi.org/10.1039/c9cp05445f

32. B. R. Brooks, C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, et al., CHARMM: The biomolecular simulation program, *J. Comput. Chem.*, **30** (2009), 1545–614. https://doi.org/10.1002/jcc.21287

33. D. A. Case, T. E. Cheatham, T. Darden, H. Gohlke, R. Luo, K. M. Merz, et al., The amber biomolecular simulation programs, *J. Comput. Chem.*, **26** (2005), 1668–1688. https://doi.org/10.1002/jcc.20290

34. D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, H. J. Berendsen, Gromacs: Fast, flexible, and free, *J. Comput. Chem.*, **26** (2005), 1701–1718. https://doi.org/10.1002/jcc.20291

35. R. H. French, V. A. Parsegian, R. Podgornik, R. F. Rajter, A. Jagota, J. Luo, et al., Long range interactions in nanoscale science, *Rev. Mod. Phys.*, **82** (2010), 1887–1944. https://doi.org/10.1103/RevModPhys.82.1887

36. A. Y. Toukmaji, J. A. Board, Ewald summation techniques in perspective: A survey, *Comput. Phys. Commun.*, **95** (1996), 73–92. https://doi.org/10.1016/0010-4655(96)00016-1

37. C. Pan, Z. Hu, Rigorous error bounds for ewald summation of electrostatics at planar interfaces, *J. Chem. Theory Comput.*, **10** (2014), 534–542. https://doi.org/10.1021/ct400839x

38. X. Cao, P. Tian, Molecular free energy optimization on a computational graph, *RSC Adv.*, **11** (2021), 12929–12937. https://doi.org/10.1039/d1ra01455b

39. X. Cao, P. Tian, "Dividing and conquering" and "caching" in molecular modeling, *Int. J. Mol. Sci.*, **22** (2021), 5053.

40. Z. Wang, D. W. Scott, Nonparametric density estimation for high-dimensional data–algorithms and applications, *WIREs Comput. Stat.*, **11** (2019), e1461. https://doi.org/10.1002/wics.1461

41. Q. Liu, J. Xu, R. Jiang, W. H. Wong, Density estimation using deep generative neural networks, *Proc. Nat. Acad. Sci.*, **118** (2021), e2101344118. https://doi.org/10.1073/pnas.2101344118

42. H. Zhang, Z. Bei, W. Xi, M. Hao, Z. Ju, K. M. Saravanan, et al., Evaluation of residue-residue contact prediction methods: From retrospective to prospective, *PLoS Comput. Biol.*, **17** (2021), e1009027. https://doi.org/10.1371/journal.pcbi.1009027

43. Y. Q. Gao, An integrate-over-temperature approach for enhanced sampling, *J. Chem. Phys.*, **128** (2008), 064105. https://doi.org/10.1063/1.2825614

44. L. Yang, C. W. Liu, Q. Shao, J. Zhang, Y. Q. Gao, From thermodynamics to kinetics: Enhanced sampling of rare events, *Acc. Chem. Res.*, **48** (2015), 947–955. https://doi.org/10.1021/ar500267n

45. R. C. Bernardi, M. C. R. Melo, K. Schulten, Enhanced sampling techniques in molecular dynamics simulations of biological systems, *Biochim. Biophys. Acta*, **1850** (2015), 872–877. https://doi.org/10.1016/j.bbagen.2014.10.019

46. J. Comer, J. C. Gumbart, J. Hénin, T. Lelièvre, A. Pohorille, C. Chipot, The adaptive biasing force method: everything you always wanted to know but were afraid to ask, *J. Phy. Chem. B*, **119** (2015), 1129–1151. https://doi.org/10.1021/jp506633n

47. V. Mlynsky, G. Bussi, Exploring RNA structure and dynamics through enhanced sampling simulations, *Curr. Opin. Struct. Biol.*, **49** (2018), 63–71. https://doi.org/10.1016/j.sbi.2018.01.004

48. Y. I. Yang, Q. Shao, J. Zhang, L. Yang, Y. Q. Gao, Enhanced sampling in molecular dynamics, *J. Chem. Phys.*, **151** (2019), 070902. https://doi.org/10.1063/1.5109531

49. W. Tschöp, K. Kremer, J. Batoulis, T. Bürger, O. Hahn, Simulation of polymer melts. I. Coarse-graining procedure for polycarbonates, *Acta Polym.*, **49** (1998), 61–74. https://doi.org/10.1002/(sici)1521-4044(199802)49:2/3¡61::Aid-apol61¿3.0.Co;2-v

50. H. Chan, M. J. Cherukara, B. Narayanan, T. D. Loeffler, C. Benmore, S. K. Gray, et al., Machine learning coarse grained models for water, *Nat. Commun.*, **10** (2019), 379. https://doi.org/10.1038/s41467-018-08222-6

51. F. Noe, A. Tkatchenko, K. R. Muller, C. Clementi, Machine learning for molecular simulation, *Annu. Rev. Phys. Chem.*, **71** (2020), 361–390. https://doi.org/10.1146/annurev-physchem-042018-052331

52. P. Gkeka, G. Stoltz, A. B. Farimani, Z. Belkacemi, M. Ceriotti, J. D. Chodera, et al., Machine learning force fields and coarse-grained variables in molecular dynamics: Application to materials and biological systems, *J. Chem. Theory Comput.*, **16** (2020), 4757–4775. https://doi.org/10.1021/acs.jctc.0c00355

53. J. Behler, Perspective: Machine learning potentials for atomistic simulations, *J. Chem. Phys.*, **145** (2016), 170901. https://doi.org/10.1063/1.4966192

54. M. Ceriotti. Unsupervised machine learning in atomistic simulations, between predictions and understanding, *J. Chem. Phys*, **150** (2019), 150901. https://doi.org/10.1063/1.5091842

55. A. Lunghi, S. Sanvito, A unified picture of the covalent bond within quantum-accurate force fields: From organic molecules to metallic complexes' reactivity, *Sci. Adv.*, **5** (2019), eaaw2210. https://doi.org/10.1126/sciadv.aaw2210

56. T. Mueller, A. Hernandez, C. Wang, Machine learning for interatomic potential models, *J. Chem. Phys.*, **152** (2020), 050902. https://doi.org/10.1063/1.5126336

57. Z. Huang, Y. Wang, X. Ma, Clustering of cancer attributed networks by dynamically and jointly factorizing multi-layer graphs, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **19** (2022), 2737–2748. https://doi.org/10.1109/TCBB.2021.3090586

58. X. Gao, X. Ma, W. Zhang, J. Huang, H. Li, Y. Li, et al., Multi-view clustering with self-representation and structural constraint, *IEEE Trans. Big Data*, **8** (2022), 882–893. https://doi.org/10.1109/tbdata.2021.3128906

59. W. Wu, X. Ma, Network-based structural learning nonnegative matrix factorization algorithm for clustering of scrna-seq data, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **20** (2023), 566–575. https://doi.org/10.1109/TCBB.2022.3161131