A Generalization of Spatial Monte Carlo Integration

Muneki Yasuda
muneki@yz.yamagata-u.ac.jp
Kei Uchizawa
uchizawa@yz.yamagata-u.ac.jp
Graduate School of Science and Engineering, Yamagata University,
Yonezawa, Yamagata 992-8510 Japan

Spatial Monte Carlo integration (SMCI) is an extension of standard Monte Carlo integration and can approximate expectations on Markov random fields with high accuracy. SMCI was applied to pairwise Boltzmann machine (PBM) learning, achieving superior results over those of some existing methods. The approximation level of SMCI can be altered, and it was proved that a higher-order approximation of SMCI is statistically more accurate than a lower-order approximation. However, SMCI as proposed in previous studies suffers from a limitation that prevents the application of a higher-order method to dense systems. This study makes two contributions. First, a generalization of SMCI (called generalized SMCI (GSMCI)) is proposed, which allows a relaxation of the aforementioned limitation; moreover, a statistical accuracy bound of GSMCI is proved. Second, a new PBM learning method based on SMCI is proposed, which is obtained by combining SMCI and persistent contrastive divergence. The proposed learning method significantly improves learning accuracy.

1 Introduction

A pairwise Boltzmann machine (PBM) (Ackley, Hinton, & Sejnowski, 1985) and its variants, including a higher-order Boltzmann machine (Sejnowski, 1986), a restricted Boltzmann machine (RBM) (Hinton, 2002; Smolensky, 1986), and a deep Boltzmann machine (DBM) (Salakhutdinov & Hinton, 2009), are fundamental and important models in the field of statistical machine learning. Except for certain cases, the inference and learning of PBMs are computationally difficult because they include a multiple summation (or integration) over all variables. Therefore, the development of their approximations has attracted attention in the field. For PBM learning, various methods have been proposed, including mean-field learning methods (e.g., mean-field approximation (Kappen & Rodríguez, 1998), Bethe approximation (or loopy belief propagation) (Yasuda & Horiguchi, 2006; Yasuda & Tanaka, 2009; Ricci-Tersenghi, 2012; Yasuda, Kataoka,
Evaluating the expectations of the variables in a PBM is critical for PBM learning. This evaluation is computationally difficult owing to multiple summations. A Monte Carlo integration (MCI) method is the simplest way to approximate such expectations, by which they are approximated using the sample average over the sample points obtained through the use of a sampling method (e.g., Gibbs sampling) on the PBM. An effective MCI, *spatial Monte Carlo integration* (SMCI), was proposed (Yasuda, 2015).

Here, the basic concept of SMCI is informally explained. Imagine a PBM, \( P(x) \), defined on an undirected (connected) graph \( G(V, F_2) \) with \( n \) vertices, where \( x \) is the set of \( n \) variables, \( V := \{1, 2, \ldots, n\} \) is the set of indices of the vertices, and \( F_2 \) is the family of undirected edges \( \{i, j\} \). Note that cliques (including undirected edges and hyperedges) are represented by the sets of corresponding indices throughout this letter. For simplicity, the variables are all \( \{-1, +1\} \)-binary. Consider the expectation of \( x_i \): \( \langle x_i \rangle = \sum x_i P(x) \). The exact evaluation of this expectation costs \( O(2^n) \). In SMCI, this expectation is approximated as follows. Suppose that the sample set, \( S \), consists of \( M \) sample points generated from the PBM using a sampling method. Take a connected region (or subgraph), \( A \), covering vertex \( i \), namely, \( \{i\} \subseteq A \subseteq V \).

For the region \( A \), the target expectation is approximated as

\[
\langle x_i \rangle \approx \frac{1}{M} \sum_{\ell=1}^{M} \sum x_A \rho(x_{\partial A} | x_{\partial A} = (\ell\text{th sample point})),
\]

where \( x_A \) is the set of variables in \( A \), \( x_{\partial A} \) is the set of variables in the neighborhood of \( A \), and \( P(x_A | x_{\partial A}) \) is the conditional distribution of the PBM. In SMCI, the target index, \( i \), is referred to as the *target region*, and \( A \) and \( \partial A \) are referred to as the *sum region* and *sample region*, respectively. The computational cost of equation 1.1 is generally \( O(M2^{|A|} + MA) \), where \( A \) is the cost of generating one sample point. Thus, equation 1.1 can be evaluated if the size of \( A \) is not large. Figure 1 illustrates the scheme of the regions of SMCI. The formal formulation of equation 1.1 is presented in section 3.

The primary concern of SMCI is how to determine the sum region. The original SMCI (Yasuda, 2015) determined the sum region to cover up to the \((k-1)\text{th nearest neighbors of a specific target region } T\). This is referred to as the *k*-th-order SMCI \((k\text{-SMCI})\) method. For the approximation accuracy of the *k*-SMCI method, two important statements were proved (Yasuda, 2015):
Figure 1: Illustration of the target, sum, and sample regions of SMCI.

the kth-order SMCI is statistically more accurate than both (1) the standard MCI method for any $k \geq 1$ and (2) the $(k-1)$-SMCI method, which guarantees that a more accurate approximation can be obtained by increasing the value of $k$ (i.e., the level of approximation of SMCI). The 1-SMCI method (the simplest $k$-SMCI method) can be applied to any graph as long as the size of the target region is not large because the sum region is identified as the target region in this simplest case. However, the 2-SMCI method is not always usable because the sum region can include $O(n)$ variables in a dense graph. This is a major drawback of the $k$-SMCI method.

In the original SMCI (Yasuda, 2015), as the level of approximation increases, the sum region is systematically expanded according to the neighboring relationship among the vertices, causing the problem mentioned above. A more flexible setting of the sum region is desired; for example, the sum region covers a part of the first-nearest neighbors of the target region, which is called the semi-second-order SMCI (s2-SMCI) method here (see section 4.1). However, accuracy bounds for such sum regions have not been clarified. Intuitively, a larger sum region can be more accurate. More concretely, for a specific target region $T$, suppose that there are two different regions $U_1, U_2$ such that $T \subseteq U_1 \subseteq U_2$; in this case, SMCI using $U_2$ as the sum region is more accurate. This intuition is in fact true (see theorem 2) and is the first contribution of this study. This type of SMCI is referred to as a generalized SMCI (GSMCI) method here. Based on this fact, one can adaptively choose the sum region according to the structure of the graph with a clear bound on the approximation accuracy. In section 2, the $k$-SMCI method proposed in the previous study is briefly described. The GSMCI method is introduced in section 3. In these sections, the $k$-SMCI and GSMCI methods are formulated on a higher-order Markov random field (HMRF), which is a generalized Markov random field and includes the PBM as a special case. The application of the SMCI methods (the $k$-SMCI and GSMCI methods) to the PBM and their numerical validation are presented in section 4.

The second contribution of this study is on PBM learning. The $k$-SMCI method was applied to PBM learning (Yasuda, 2015, 2018; Yasuda et al., 2019). For PBM learning, the method based on the 1-SMCI method was
shown to be superior to other known learning methods (MPLE, RM, and MPF) (Yasuda, 2018). In the original learning method (Yasuda, 2015, 2018), the variables in the sample region are fixed by the given training set, leading to an useful deterministic algorithm. However, the accuracy of the learning cannot be improved without increasing the level of approximation (i.e., increasing the value of \( k \)) in this method. In this letter, a new learning method is proposed by combining SMCI with persistent contrastive divergence (PCD) (Tieleman, 2008). This proposed method allows the accuracy of the learning to be improved without increasing the level of approximation of SMCI. The proposed learning method (with its pseudocode) and its numerical validations are described in section 5. Finally, the summary and some discussions are presented in section 6.

2 Spatial Monte Carlo Integration

In this section, the original SMCI, that is, the \( k \)-SMCI method, is described.

2.1 Higher-Order Markov Random Field. Consider a higher-order MRF (HMRF) consisting of \( n \) random variables, \( x := \{ x_i \in \mathcal{X}_i \mid i \in \mathcal{V} \} \), where \( \mathcal{X}_i \) is the sample space of \( x_i \) and \( \mathcal{V} = \{ 1, 2, \ldots, n \} \) is the set of indices of the variables, which is defined by

\[
P(x) := \frac{1}{Z} \exp \left( \sum_{C \in \mathcal{F}} \phi_C(x_C) \right),
\]

where \( C \) denotes a clique, \( C \subseteq \mathcal{V} \), and \( \mathcal{F} \) denotes a family of cliques. An HMRF is regarded as a probabilistic graphical model on a hypergraph, in which \( \mathcal{V} \) is regarded as a set of vertices and \( \mathcal{F} \) is regarded as a family of hyperedges in the hypergraph. The function \( \phi_C(x_C) \) denotes a potential function on \( C \), where \( x_C := \{ x_i \mid i \in C \subseteq \mathcal{V} \} \). In equation 2.1, \( Z \) denotes the partition function defined by

\[
Z := \sum_x \exp \left( \sum_{C \in \mathcal{F}} \phi_C(x_C) \right),
\]

where \( \sum_x := \sum_{x_1 \in \mathcal{X}_1} \sum_{x_2 \in \mathcal{X}_2} \cdots \sum_{x_n \in \mathcal{X}_n} = \prod_{i \in \mathcal{V}} \sum_{x_i \in \mathcal{X}_i} \) denotes the summation over all possible realizations of \( x \). It should be noted that when \( \mathcal{X}_i \) is a continuous space, the corresponding sum \( \sum_{x_i \in \mathcal{X}_i} \) is replaced with the integration \( \int_{\mathcal{X}_i} dx_i \). When \( \phi_C(x_C) = w_C \prod_{i \in C} x_i \), the HMRF is identical to a generalization of a higher-order Boltzmann machine (Sejnowski, 1986). In particular, when \( \mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \) (namely, a pairwise case), where \( \mathcal{F}_1 := \{ \{ i \} \mid i \in \mathcal{V} \} \), \( \mathcal{F}_2 \) is a family of pairs of indices \( \{ i, j \} \) in which \( i \) and \( j \) are directly
Figure 2: Adjacency relations among the indices. For example, $i$ and $j$ are first-nearest-neighboring indices of each other, and $i$ and $l$ are second-nearest-neighboring indices of each other.

interacted with each other, and $\phi_C(x_C) = w_C \prod_{i \in C} x_i$, equation 2.1 is identical to a familiar PBM (Ackley et al., 1985):

$$P(x) := \frac{1}{Z} \exp \left( \sum_{i \in V} w_i x_i + \sum_{\{i,j\} \in F_2} w_{i,j} x_i x_j \right),$$

(2.2)

where $w_{i,j}$ is identical to $w_{j,i}$. Here, $w_i$ and $w_{i,j}$ are called the bias and interaction parameters, respectively. Note that $F_2$ does not include $\{i,j\}$ as its element when $i$ and $j$ have no direct interaction; therefore, $F_2$ is regarded as the family of undirected edges in an undirected graph, $G(V, F_2)$, on which the PBM is defined.

Here, consider the expectation of a function over specific target variables, $x_T$, where $T \subseteq V$, on the HMRF, which is

$$\langle f(x_T) \rangle := \sum_x f(x_T)P(x) = \sum_{x_T} f(x_T)P(x_T).$$

(2.3)

where $P(x_T) = \sum_{x \setminus x_T} P(x)$ is the marginal distribution of the HMRF. However, in general, the evaluation of this expectation is computationally difficult owing to the multiple summations. SMCI, an extension of the standard MCI method, was proposed to efficiently approximate the expectation (Yasuda, 2015).

Hereafter, we assume that the size of $T$ is not large and that the sum over $x_T$ can be numerically evaluated.

2.2 Adjacency Relations in HMRF. For a detailed description of the $k$-SMCI method, an adjacency relation among the indices must be defined. If a clique $C$ involves both $i$ and $j$, both indices are regarded as the first-nearest-neighboring indices of each other (in other words, both indices are connected). If no cliques involve both $i$ and $j$ and two different cliques involving $i$ and $j$, respectively, overlap, both indices are regarded as the second-nearest-neighboring indices of each other (see Figure 2).
Similar to the above, the neighboring regions of a target region $T \subseteq V$ are defined as follows. For target region $T$, the first-nearest-neighboring region of $T$, $N_1(T) \subseteq V$, is defined as

$$N_1(T) := \{ i \mid i \in C \in \mathbb{F}(T), i \notin T \},$$

where $\mathbb{F}(A)$ is a subset of $\mathbb{F}$, which is the family of the cliques overlapping the assigned set $A$, that is, $\mathbb{F}(A) := \{ C \mid C \in \mathbb{F}, C \cap A \neq \emptyset \}$. In other words, the indices in $N_1(T)$ do not belong to $T$ and simultaneously are the first-nearest neighbors of the indices in $T$. The second-nearest-neighboring region of $T$, $N_2(T) \in V$, is defined as

$$N_2(T) := \{ i \mid i \in C \in \mathbb{F}(N_1(T)), i \notin T \cup N_1(T) \}.$$

This means that the indices in $N_2(T)$ do not belong to $T$ or $N_1(T)$ and simultaneously are the second-nearest neighbors of the indices in $T$. In a similar manner, the $k$th-nearest-neighboring region of $T$, $N_k(T) \subseteq V$, is defined as

$$N_k(T) := \{ i \mid i \in C \in \mathbb{F}(N_{k-1}(T)), i \notin R_{k-1}(T) \},$$

where $R_k(T) := \bigcup_{r=0}^{k} N_r(T)$ and $N_0(T) = T$, that is, $R_k(T)$ is the region covering regions $T, N_1(T), \ldots, N_k(T)$.

The adjacency relations can be easily understood in a pairwise case in which $\mathbb{F} = \mathbb{F}_1 \cup \mathbb{F}_2$. In this case, the adjacency relations discussed above are identical to the standard adjacency relations in an undirected graph $G(V, \mathbb{F}_2)$. Examples of the adjacency relations in a pairwise case, in which $G(V, \mathbb{F}_2)$ is a square grid graph, are illustrated in Figure 3.
2.3  $k$th-Order SMCI Method for HMRF. The spatial Markov property of the HMRF ensures that the conditional distribution $P(x_{R_{k-1}(T)} \mid x \setminus x_{R_{k-1}(T)})$ can be expressed as

$$P(x_{R_{k-1}(T)} \mid x \setminus x_{R_{k-1}(T)}) = P(x_{R_{k-1}(T)} \mid x_{N_k(T)}),$$

where

$$P(x_{R_{k-1}(T)} \mid x_{N_k(T)}) \propto \exp \left( \sum_{C \in \mathcal{F}(R_{k-1}(T))} \phi_C(x_C) \right)$$

is also an HMRF. Therefore, the expectation in equation 2.3 can be expressed as

$$\langle f(x_T) \rangle = \sum_{x_{R_{k-1}(T)}} f(x_T) P(x_{R_{k-1}(T)} \mid x_{N_k(T)}) P(x_{N_k(T)}), \quad (2.5)$$

where $P(x_{N_k(T)})$ is the marginal distribution of $P(x)$.

Suppose that $M$ independent and identically distributed (i.i.d.) sample points are drawn from $P(x)$: $S := \{s^{(\ell)} \mid \ell = 1, 2, \ldots, M\}$, where $s^{(\ell)} := \{s_i^{(\ell)} \in X_i \mid i \in V\}$ is the $\ell$th sample point. In the $k$-SMCI method (Yasuda, 2015), the expectation in equation 2.5 is approximated by

$$m^{(k)}_T(S) := \frac{1}{M} \sum_{\ell=1}^M \sum_{x_{R_{k-1}(T)}} f(x_T) P(x_{R_{k-1}(T)} \mid x_{N_k(T)}) P(x_{N_k(T)}), \quad (2.6)$$

where $s_i^{(\ell)} := \{s_i^{(\ell)} \mid i \in A\}$. In this approximation, $R_{k-1}(T)$ is the sum region, and $N_k(T)$ is the sample region. Equation 2.6 is obtained by replacing the marginal distribution $P(x_{N_k(T)})$ in equation 2.5 with the corresponding empirical distribution of the given sample points, which is defined by

$$Q_S(x_{N_k(T)}) := \frac{1}{M} \sum_{\ell=1}^M \delta(x_{N_k(T)} \cdot s_i^{(\ell)}),$$

where $\delta$ is the Kronecker (or Dirac) delta function:

$$\langle f(x_T) \rangle \approx \sum_{x_{R_{k-1}(T)}} f(x_T) P(x_{R_{k-1}(T)} \mid x_{N_k(T)}) Q_S(x_{N_k(T)}) = m^{(k)}_T(S).$$

The $k$-SMCI method is usable when the sum over $x_{R_{k-1}(T)}$ is computable.
2.4 Asymptotic Analysis of k-SMCI Method. Because the sample points \( s^{(\ell)} \) are i.i.d. random variables,

\[
\rho^{(k)}_{T}(s^{(N_{\ell}(T)}) := \sum_{x_{N_{\ell-1}(T)}} f(x_{T})P(x_{R_{k-1}(T)} | s^{(\ell)}_{N_{\ell}(T)})
\]

are also regarded as i.i.d. random variables. Therefore, from the result of the central limit theorem, the distribution of \( m^{(k)}_{T}(S) = M^{-1} \sum_{\ell=1}^{M} \rho^{(k)}_{T}(s^{(\ell)}_{N_{\ell}(T)}) \) is asymptotically close to the gaussian with mean

\[
\mu^{(k)}_{T} := \left( \prod_{\ell=1}^{M} \sum_{s^{(\ell)}} P(s^{(\ell)}) \right) m^{(k)}_{T}(S)
\]
and variance

\[
v^{(k)}_{T} := \left( \prod_{\ell=1}^{M} \sum_{s^{(\ell)}} P(s^{(\ell)}) \right) m^{(k)}_{T}(S)^2 - (\mu^{(k)}_{T})^2 = O(M^{-1}),
\]
for a sufficiently large \( M \), where \( \sum_{s^{(\ell)}} := \prod_{i \in V} \sum_{s^{(\ell)}_{i} \in X_{i}} \) and \( P(s^{(\ell)}) \) is the HMRF in equation 2.1. The asymptotic mean is equivalent to the exact expectation, that is, \( \mu^{(k)}_{T} = \langle f(x_{T}) \rangle \). Therefore, \( m^{(k)}_{T}(S) \) converges to \( \langle f(x_{T}) \rangle \) as \( M \) approaches infinity, and its variance vanishes at a speed of \( O(M^{-1}) \).

From a statistical perspective, a method with a smaller asymptotic variance is preferable. For the asymptotic variance \( v^{(k)}_{T} \), the following theorem was obtained (Yasuda, 2015).

**Theorem 1.** In the HMRF expressed in equation 2.1, the inequality relation \( v^{(1)}_{T} \geq v^{(2)}_{T} \geq v^{(3)}_{T} \geq \ldots \geq 0 \) always holds for any \( M \) and for any choice of target region \( T \), where \( v_{T} \) is the asymptotic variance of the standard MCI method.

This theorem states that for a sufficiently large \( M \), the k-SMCI method is statistically more accurate than the standard MCI method for any \( k \geq 1 \) and that a higher-order SMCI method is statistically more accurate than any lower-order method.

3 Generalization of SMCI

In this section, the GSMCI method is introduced. Here, for region \( A \), satisfying \( T \subseteq A \subseteq V \), and for sample set \( S \), consider an SMCI defined as

\[
m_{T}(A; S) := \frac{1}{M} \sum_{\ell=1}^{M} \sum_{x_{A}} f(x_{T})P(x_{A} | s^{(\ell)}_{\partial A}),
\]

(3.1)
where $\partial A$ denotes the first-nearest-neighboring region of region $A$: $\partial A := \{ i \mid i \in C \in F(A), i \not\in A \}$. The conditional distribution in equation 3.1 is expressed as

$$
P(x_A | x_{\partial A}) \propto \exp \left( \sum_{C \in F(A)} \phi_C(x_C) \right).$$

(3.2)

When $A = R_k(T)$, $\partial A$ is identical to $N_k(T)$; therefore, equation 3.1 is equivalent to equation 2.6 in this case. Therefore, equation 3.1 is regarded as a generalization of the $k$-SMCI method. Analogous to the $k$-SMCI method, it is expected that the approximation accuracy of equation 3.1 will increase as the sum region $A$ increases. In fact, the following argument justifies this expectation.

Consider two sum regions $U_1$ and $U_2$ that satisfy $T \subseteq U_1 \subseteq U_2$. For these two sum regions, based on equation 3.1, the two approximations $m_T(U_1; S)$ and $m_T(U_2; S)$ can be considered for the purpose of approximating $\langle f(x_T) \rangle$. With a similar argument obtained in section 2.4, the asymptotic properties of both approximations are analyzed as follows. For a sufficiently large $M$, the distributions of both $m_T(U_1; S)$ and $m_T(U_2; S)$ are asymptotically close to the different gaussians with mean $\mu(U_1)$ and variance $v(U_1)$, and with mean $\mu(U_2)$ and variance $v(U_2)$, respectively. These asymptotic means and variances are defined as

$$
\mu(A) := \left( \prod_{\ell=1}^{M} \sum_{s^{(\ell)}} P(s^{(\ell)}) \right) m_T(A; S) = \langle f(x_T) \rangle,
$$

(3.3)

$$
v(A) := \left( \prod_{\ell=1}^{M} \sum_{s^{(\ell)}} P(s^{(\ell)}) \right) m_T(A; S)^2 - \mu(A)^2,
$$

(3.4)

for a sum region $T \subseteq A \subseteq V$. Equation 3.4 is rewritten as

$$
v(A) = \frac{1}{M} \left( \sum_{x_{\partial A}} \rho_T(A; x_{\partial A})^2 P(x_{\partial A}) - \langle f(x_T) \rangle^2 \right),
$$

(3.5)

where

$$
\rho_T(A; x_{\partial A}) := \sum_{x_A} f(x_T) P(x_A | x_{\partial A}).
$$

(3.6)

Here, $P(x_{\partial A})$ and $P(x_A | x_{\partial A})$ are the marginal and conditional distributions of the HMRF of equation 2.1, respectively. Therefore, the two approximations $m_T(U_1; S)$ and $m_T(U_2; S)$ converge to the true expectation $\langle f(x_T) \rangle$ as
M approaches infinity, and their variances vanish at speeds of $O(M^{-1})$, as with the $k$-SMCI method.

For the asymptotic variances $v(U_1)$ and $v(U_2)$, the following theorem is obtained:

**Theorem 2.** In the HMRF expressed in equation 2.1, for $T \subseteq U_1 \subseteq U_2$, the inequality relation $v(U_1) \geq v(U_2) \geq 0$ always holds for any $M$ and for any choice of target region $T$.

The proof of this theorem is described in appendix A. This states that $m_T(U_2; S)$ is statistically more accurate than $m_T(U_1; S)$ for a sufficiently large $M$. It is noteworthy that this theorem includes the statement of theorem 1 as its corollary.

Equation 3.1 is referred to as the GSMCI method here. Because equation 3.1 is identical to the 1-SMCI method when $T = A$, the GSMCI method is statistically more accurate than the standard MCI method for any choice of $A$ satisfying $T \subseteq A \subseteq V$. From theorem 2, any sum region can be freely selected in equation 3.1, and it is guaranteed that the approximation accuracy of the GSMCI method will monotonically increase as the size of the selected sum region increases. The GSMCI method is usable when the sum over $x_A$ is computable.

### 4 Application to PBM

In this section, the SMCI methods ($k$-SMCI and GSMCI methods) for the PBM, defined on an undirected graph $G(V, \mathcal{F}_2)$, are considered, where the sample spaces of the variables are fixed to $X_i = \{-1, +1\}$.

The evaluations of $\langle x_i \rangle$ and $\langle x_i x_j \rangle$ for $\{i, j\} \in \mathcal{F}_2$ are essential for PBM learning. From equation 2.6, these expectations are approximated by (Yasuda, 2015)

$$m_i^{(1)}(S) = \frac{1}{M} \sum_{\ell=1}^{M} \tanh \gamma_i(s_{N_i(i)}^\ell)$$

and

$$m_{i,j}^{(1)}(S) = \frac{1}{M} \sum_{\ell=1}^{M} \tanh \left[ \operatorname{atanh} \left( \tanh \gamma_{i,j}(s_{\mathcal{N}_i(i)}^\ell) \tanh \gamma_{j,i}(s_{\mathcal{N}_j(i,j)}^\ell) \right) \right] + w_{i,j},$$

respectively, based on the 1-SMCI method, where

$$\gamma_i(s_{\mathcal{N}_i(i)}^\ell) := w_i + \sum_{j \in \mathcal{N}_i(i)} w_{i,j}s_j^{(\ell)},$$

$$\gamma_{i,j}(s_{\mathcal{N}_i(i,j)}^\ell) := \gamma_i(s_{\mathcal{N}_i(i)}^\ell) - w_{i,j}s_j^{(\ell)}.$$
and \( \text{atanh} \) is the inverse hyperbolic tangent function. The detailed derivations of equations 4.1 and 4.2 are described in appendix B.1. Equations 4.1 and 4.2 are computable for any \( G(V, F_2) \). However, the 2-SMCI methods for \( \langle x_i \rangle \) and \( \langle x_i x_j \rangle \) are not always computable; for example, when \( G(V, F_2) \) is a dense graph such as \( R_1(T) = O(n) \), the computational cost of the evaluation of \( m^{(2)}_T(S) \) is generally \( O(2^n) \) because all variables in \( R_1(T) \) must be summed over.

### 4.1 Semi-Second-Order SMCI Method.

In the GSMCI method in equation 3.1, the sum region \( A \) can be freely selected. The sum region should be as large as possible within the computational limitation. In general, the computational cost of evaluating the sum over the sum region is \( O(2^{|A|}) \). However, it can be drastically reduced when an efficient analytical evaluation can be applied to the sum region. For example, if \( A \) is a (cactus) tree, equation 3.1 can be computed using (generalized) belief propagation (Yedidia, Freeman, & Weiss, 2005) (when \( A \) is a sparse tree, the computational cost of the evaluation of the sum over the sum region may be reduced to \( O(|A|) \)); if \( A \) is a planar graph, it can be computed using a combinational technique (Johnson, Oyen, Chertkov, & Netrapalli, 2016). The appropriate choice of the sum region depends on the structure of \( G(V, F_2) \).

In the following, a setting of \( A \) that is usable in general cases, that is, a semi-second-order SMCI (s2-SMCI) method, is proposed. Consider a subset \( I_1(T) \subseteq N_1(T) \) in which there is no connected (or interacted) pair, so any two different indices in \( I_1(T) \) belong to different cliques. For \( A = T \cup I_1(T) \), the conditional distribution in equation 3.2 is represented as

\[
P(x_A | x_{\partial A}) \propto \exp \left( \sum_{i \in A} \beta_i(x_{\partial A}) x_i + \sum_{\{i, j\} \in \{c | c \subseteq T, c \in F_2\}} w_{i, j} x_i x_j + \sum_{i \in T} \sum_{j \in I_1(T)} w_{i, j} x_i x_j \right), \tag{4.5}
\]

where

\[
\beta_i(x_{\partial A}) := w_i + \sum_{j \in \partial A} w_{i, j} x_j. \tag{4.6}
\]

In equations 4.5 and 4.6, \( w_{i, j} \) are regarded as zero when \( \{i, j\} \notin F_2 \). The second term of the exponent of equation 4.5 denotes the interactions within the target region, and the third term denotes the interactions between \( T \) and \( I_1(T) \). Because the variables in \( I_1(T) \) do not interact, they can be analytically eliminated from equation 4.5 using the marginalization, leading to
Figure 4: Illustration of the sum regions of the 1-, s2-, and 2-SMCI methods (the sum regions are shaded): (a) the 1-SMCI method, (b) an example of the s2-SMCI method, and (c) the 2-SMCI method. In the s2-SMCI method, another choice of the sum region is possible.

\[ P(x_T \mid x_{\partial A}) = \sum_{x_{I_1}(T)} P(x_A \mid x_{\partial A}) \]

\[ \propto \exp \left\{ \sum_{i \in T} \beta_i(x_{\partial A}) x_i + \sum_{\{i, j\} \in \{c \subseteq T \mid c \in F_2\}} w_{i, j} x_i x_j \right\} \]

\[ + \sum_{j \in I_1(T)} \ln z_j \left( \beta_j(x_{\partial A}) + \sum_{i \in T} w_{i, j} x_i \right) \]

where \( z_i(a) = \sum_{x_i \in X_i} \exp(ax_i) = 2 \cosh a \). Therefore, using the marginal distribution in equation 4.7, the GSMCI method in equation 3.1 can be reduced to

\[ m_T(A; S) = \frac{1}{M} \sum_{\ell=1}^{M} \sum_{x_T} f(x_T) P(x_T \mid s^{(\ell)}_{\partial A}), \quad A = T \cup I_1(T). \] (4.8)

Equation 4.8 is the s2-SMCI method proposed in this section. In this method, the computational cost for evaluating the sum for the sum region is reduced to \( O(2^{|T|}) \); therefore, this method can be used in a dense graph (as long as the sum over \( x_T \) can be evaluated). It is noteworthy that the s2-SMCI method can be applied to a large \( T \) if there exists an efficient analytical treatment for the sum over \( x_T \), as mentioned at the beginning of this section.

The s2-SMCI method is regarded as an intermediate approximation between the 1- and 2-SMCI methods because it is identical to the 1-SMCI method when \( I_1(T) = \emptyset \) and is identical to the 2-SMCI method when \( I_1(T) = N_1(T) \); therefore, from the result obtained in theorem 2, the approximation accuracy of the s2-SMCI method should be at an intermediate level between those of the 1- and 2-SMCI methods (see Figure 4). The approximation accuracy of the s2-SMCI method would be closer to that of the 1-SMCI method as the graph becomes denser. In an extremely dense case (e.g., in a
complete graph), the size of $I_1(T)$ will be very small (e.g., $|I_1(T)| = 1$ in a complete graph); therefore, the $s^2$-SMCI method is almost the same as the 1-SMCI method in such a case.

Using equations 4.7 and 4.8, the $s^2$-SMCI methods for the expectations, $\langle x_i \rangle$ and $\langle x_i x_j \rangle$ ($\{i, j\} \in \mathbb{F}_2$), are expressed as

$$m_i(A; \mathcal{S}) = \frac{1}{M} \sum_{i=1}^{M} \tanh \xi_i(s_{iA}^{(i)}),$$

(4.9)

and

$$m_{i,j}(A; \mathcal{S}) = \frac{1}{M} \sum_{i=1}^{M} \tanh \left\{ \frac{\tanh \xi_i(s_{iA}^{(i)}) \tanh \xi_j(s_{jA}^{(i)}) + \omega_{i,j}(s_{iA}^{(i)})}{\tanh \beta_i(s_{iA}^{(i)}) + w_{i,j}} \right\},$$

(4.10)

respectively, where

$$\xi_i(s_{iA}^{(i)}) := \beta_i(s_{iA}^{(i)}) + \sum_{j \in I_i(i)} \tanh \{ \tanh \beta_j(s_{jA}^{(i)}) \tanh w_{i,j} \},$$

(4.11)

$$\xi_{i,j}(s_{iA}^{(i)}) := \beta_i(s_{iA}^{(i)}) + \sum_{k \in I_i(i,j)} \tanh \{ \tanh \beta_k(s_{jA}^{(i)}) \tanh w_{i,k} \}$$

$$+ \frac{1}{4} \sum_{k \in I_i(i,j)} \ln \frac{1 - \tanh^2 \beta_k(s_{jA}^{(i)}) + w_{i,k}}{1 - \tanh^2 \beta_k(s_{jA}^{(i)}) - w_{i,k}} \tanh^2 w_{i,k}$$

(4.12)

$$\omega_{i,j}(s_{iA}^{(i)}) := w_{i,j} + \sum_{k \in I_i(i,j)} \tanh \{ \tanh w_{i,k} \tanh w_{j,k} \}$$

$$+ \frac{1}{4} \sum_{k \in I_i(i,j)} \ln \frac{1 - \tanh^2 (w_{i,k} + w_{j,k}) \tanh^2 \beta_k(s_{jA}^{(i)})}{1 - \tanh^2 (w_{i,k} - w_{j,k}) \tanh^2 \beta_k(s_{jA}^{(i)})}.$$  

(4.13)

In equations 4.11 to 4.13, $w_{i,j}$ are zero when $\{i, j\} \notin \mathbb{F}_2$. The detailed derivations of equations 4.9 and 4.10 are described in section B.2. It is noteworthy that in the sums in the third terms in equations 4.12 and 4.13, the terms for $k \in I_i(i, j)$ with $w_{i,k} = 0$ or $w_{j,k} = 0$ vanish.

From theorem 2, a larger $I_1(T)$ is preferable. However, the maximization of the size of $I_1(T)$ is known as the maximum independent set (MIS) problem, which is an NP-hard optimization problem. The well-known greedy algorithm (Halldórsson & Radhakrishnan, 1997) for this problem is presented in algorithm 1. In algorithm 1, $\deg(i; \mathcal{U})$ denotes the degree of $i$ in subgraph $\mathcal{U}$, and $\partial_i := \{ j \mid j \in \mathcal{U}, \{i, j\} \in \mathbb{F}_2 \}$ denotes the set of indices that are the first-nearest neighbors of $i$ in subgraph $\mathcal{U}$, i.e., $\deg(i; \mathcal{U}) = |\partial_i(i)|$. Algorithm 1 can be implemented in time linear in the number of edges and vertices (Halldórsson & Radhakrishnan, 1997).
Algorithm 1: Greedy Algorithm for MIS Problem.

1: **Input** $N_1(T) \subseteq \mathcal{V}$ and $\mathcal{F}_2$

2: $I_1(T) \leftarrow \emptyset$ and $\mathcal{U} \leftarrow N_1(T)$

3: **repeat**

4: Choose index $r$ such that $r = \arg \min_{j \in \mathcal{U}} \deg(j; \mathcal{U})$

5: $I_1(T) \leftarrow I_1(T) \cup \{r\}$

6: $\mathcal{U} \leftarrow \mathcal{U} \setminus (\partial_T r \cup \{r\})$

7: **until** $\mathcal{U} \neq \emptyset$

8: **Output** $I_1(T)$.

In step 4 in algorithm 1, a case in which multiple indices have the same minimum degree, which are equal candidates of $r$, may be encountered. When the minimum degree is zero, the selection does not affect the final result because all candidates will be included in $I_1(T)$. However, when the minimum degree is larger than zero, one of them must be selected, and the selection can affect the final result. A criterion for the selection is needed. In this letter, a heuristic for the selection is proposed: when multiple indices have the same minimum degree of larger than zero, the index with the maximum $W_j$ is selected, in which $W_j := \sum_{i \in T \cap N_1(j)} |w_{i,j}|$. Here, $W_j$ is regarded as the absolute strength of the interaction between index $j$ and the target region. This heuristic, called $W$-ordering here, is based on a common notion: a pair having a stronger interaction is more important.

4.2 Experiment. In this section, we demonstrate the validity of the proposed method using numerical experiments. We used PBM of $|\mathcal{V}| = n = 20$ defined on two types of undirected graphs $G(\mathcal{V}, \mathcal{F}_2)$: a $4 \times 5$ square grid graph and a random graph with connection probability $p$. The bias and interaction parameters, $w_i$ and $w_{i,j}$, in the PBM were generated from uniform distributions having intervals $[-0.2, +0.2]$ and $[-0.3, +0.3]$, respectively. In the PBM, the approximation accuracies of the SMCI methods (1-, s2-, and 2-SMCI methods) were checked. The accuracy of the approximation was measured using the mean absolute error (MAE) of the covariances,

$$\text{MAE} = \frac{1}{|\mathcal{F}_2|} \sum_{[i,j] \in \mathcal{F}_2} |\chi_{i,j}^{\text{exact}} - \chi_{i,j}^{\text{approx}}|,$$

where $\chi_{i,j}^{\text{exact}} := \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle$ is the exact covariance and $\chi_{i,j}^{\text{approx}}$ is its approximation. Because the size of $\mathcal{V}$ is not large, the exact covariance can
Figure 5: MAEs for $M = 10, 100, 1000$ in (a) a $4 \times 5$ square grid graph, (b) a random graph of $p = 0.2$, and (c) a random graph of $p = 0.4$. The open and filled circles denote the standard MCI and 1-SMCI methods, respectively; the open and filled squares denote the s2- and 2-SMCI methods, respectively. The plots are the average values over 200 experiments.

be numerically evaluated. The sample sets, $S$, used in the SMCI methods were generated from the PBM using Gibbs sampling based on simulated annealing. The plots in Figure 5 show the results in (a) a $4 \times 5$ square grid graph, (b) a random graph of $p = 0.2$, and (c) a random graph of $p = 0.4$ for various $M$. For comparison, the results obtained through the standard MCI method are also plotted. For the 2-SMCI method, the sum over $x_{R_i(T)}$ was numerically evaluated. As expected, the SMCI methods are superior to the standard MCI method, and the accuracy of the s2-SMCI method is between the accuracies of the 1- and 2-SMCI methods. The accuracies of the SMCI methods in panel c are worse than those in panels a and b. This indicates that they are more effective in a sparser graph. The accuracy of the s2-SMCI method is closer to that of the 1-SMCI method as $p$ increases, because the difference between the 1- and s2-SMCI methods tends to be smaller in a denser graph, as mentioned in the previous section.

In Figures 5a to 5c, the accuracy of the standard MCI method of $M = 1000$ is almost the same as that of the 1-SMCI method of $M = 10$; therefore, the standard MCI method needs an approximately 100 times larger sample set to reach the accuracy of the 1-SMCI method (at least in the settings of these experiments). The MAEs were also evaluated using annealed importance sampling (AIS) (Neal, 2001). The results of AIS were almost the same as those of the standard MCI method. For the AIS, the initial distribution was set to uniform over $\{-1, +1\}^n$, and a sequence of the inverse temperatures, $0 = \beta_0 < \beta_1 < \cdots < \beta_K = 1$, was set as $\beta_{k+1} = \beta_k + 10^{-4}$.

5 Application to PBM Learning

In this section, we consider the learning of the PBM defined on $G(V, F_2)$. The PBM is represented with the explicit dependency on its parameters
\[ P(x) = P(x | \theta), \] where \( \theta \) denotes the set of all parameters of the PBM: \( \theta := \{ w_i, w_{i,j} \mid i \in \mathcal{V}, \{ i, j \} \in \mathcal{F}_2 \} \). Suppose that a training data set consisting of \( N \) training data points, \( \mathbb{D} := \{ x^{(\mu)} \mid \mu = 1, 2, \ldots, N \} \), is obtained. For the data set, the log likelihood is defined as

\[
L_\mathbb{D}(\theta) := \frac{1}{N} \sum_{\mu=1}^{N} \ln P(x^{(\mu)} | \theta). \quad (5.1)
\]

The log likelihood is the function with respect to \( \theta \). PBM learning is performed using the maximum likelihood estimation (MLE), that is, by maximizing the log likelihood with respect to \( \theta \). The parameters in \( \theta \) are referred to as the learning parameters. The gradients of the log likelihood with respect to \( w_i \) and \( w_{i,j} \) for \( \mathbb{D} \), are

\[
g_i(\mathbb{D}, \theta) := \frac{\partial L_\mathbb{D}(\theta)}{\partial w_i} = \frac{1}{N} \sum_{\mu=1}^{N} x_i^{(\mu)} - \langle x_i \rangle, \quad (5.2)
\]

\[
g_{i,j}(\mathbb{D}, \theta) := \frac{\partial L_\mathbb{D}(\theta)}{\partial w_{i,j}} = \frac{1}{N} \sum_{\mu=1}^{N} x_i^{(\mu)} x_j^{(\mu)} - \langle x_i x_j \rangle, \quad (5.3)
\]

respectively. These gradients have the intractable expectations in their second terms.

An approximation based on the \( k \)-SMCI method was proposed (Yasuda, 2015), by which the intractable expectations are approximated using the \( k \)-SMCI method: \( \langle x_i \rangle \approx m_i^{(k)}(\mathbb{D}) \) and \( \langle x_i x_j \rangle \approx m_{i,j}^{(k)}(\mathbb{D}) \). In this approximation, the sample set \( S \) is fixed to the data set. This approximation method based on the 1-SMCI method provides better learning results than some existing learning algorithms (Yasuda, 2018): MPLE (Besag, 1975), RM (Hyvärinen, 2007), and MPF (Sohl-Dickstein et al., 2011).

In the previous learning method (Yasuda, 2015), the variables in the sample region were fixed by the given training set, leading to a useful deterministic algorithm. However, with this method, the accuracy of the learning cannot be improved without increasing the level of approximation of SMCI (i.e., increasing the value of \( k \)). Moreover, the accuracy of the learning was degraded in a model-mismatched case (Yasuda, 2018), in which the graph structures of the data generating PBM and those of the learning PBM were different (more precisely, the graph of the learning PBM did not have the graph of the generating PBM as a subgraph). This degradation was mainly caused by fixing the sample set to the data set. An SMCI method is obtained by approximating the marginal distribution over the sample region by the sample distribution (see section 2.3). In a model-mismatched case, the data distribution is no longer a good approximation of the corresponding marginal distribution in the sample region.
### 5.1 Proposed Learning Algorithm

This letter proposes a new approximation for PBM learning. First, a sample set \( S \) with \( M = eN \) is prepared, where \( M \) is the size of the sample set, and \( e \) is a positive integer called the “data-extension rate.” The sample set and the learning parameters are initialized as \( S = S_0 \) and \( \theta = \theta_0 \), respectively. Using \( S_0 \), the learning parameters \( \theta_0 \) are updated to \( \theta_1 \) based on a gradient ascent method with the approximate gradients \( g_i^{\text{app}}(S_0, \theta_0) \) and \( g_{ij}^{\text{app}}(S_0, \theta_0) \) defined as

\[
g_i^{\text{app}}(S, \theta) := \frac{1}{N} \sum_{\mu=1}^{N} x_i^{(\mu)} - m_i(A_i; S), \tag{5.4}
\]

\[
g_{ij}^{\text{app}}(S, \theta) := \frac{1}{N} \sum_{\mu=1}^{N} x_i^{(\mu)} x_j^{(\mu)} - m_{ij}(A_{ij}; S), \tag{5.5}
\]

where \( m_i(A_i; S) \) and \( m_{ij}(A_{ij}; S) \) are the approximations of \( \langle x_i \rangle \) and \( \langle x_i x_j \rangle \) on \( P(x | \theta) \), respectively, based on the GSMCI method proposed in equation 3.1; here, \( A_i \) and \( A_{ij} \) are the sum regions determined for the corresponding target regions. After the update of \( \theta \), the sample set \( S_0 \) is updated to \( S_1 \) using \( M \)-parallel \( \kappa \)-step Gibbs sampling on \( P(x | \theta_1) \). By applying \( S_1 \), the learning parameters are again updated to \( \theta_2 \) using the gradients \( g_i^{\text{app}}(S_1, \theta_1) \) and \( g_{ij}^{\text{app}}(S_1, \theta_1) \). This two-stage updating procedure, the parameter update and sample set update stages, is repeated during learning: \( S_0, \theta_0 \rightarrow \theta_1 \rightarrow S_1 \rightarrow \theta_2 \rightarrow S_2 \rightarrow \ldots \). The proposed procedure is inspired by the PCD method (Tieleman, 2008). The initial state of the sample set, \( S_0 \), is set to the \( e \)-replicated \( D \). The update procedure of the sample set is illustrated in Figure 6. The pseudocode of the proposed learning is presented in Algorithm 2.

### 5.2 Experiment

In this section, we demonstrate the performance of the proposed learning algorithm described in the previous section. In the following experiments, the training data sets of \( N = 50 \) were generated from a generative PBM (g-PBM) with \( n = 20 \), which has the same form as equation 2.2, using Gibbs sampling based on simulated annealing. A training PBM (t-PBM) with the same size as g-PBM was trained using the generated artificial data sets. Two cases are considered: (1) a model-matched case in which g-PBM is defined on a \( 4 \times 5 \) square grid graph and t-PBM is also defined on the same square grid graph, and (2) a model-mismatched case in which g-PBM is defined on a fully connected graph and t-PBM is defined on a \( 4 \times 5 \) square grid graph. Because the size of t-PBM is not too large, MLE can be performed exactly. The parameters \( w_i \) and \( w_{ij} \) in g-PBM were randomly selected according to uniform distributions having the intervals \([-0.2, +0.2]\) and \([-0.3, +0.3]\), respectively. In the following experiments, \( \kappa \) was fixed to one, and the learning rate was fixed to 0.02.
Figure 6: Illustration of the update procedure of the sample set when $e = 3$. $S_0$ is set to the $e$-replicated $\mathbb{D}$. Each sample point in $S_t$ is updated using $\kappa$-step Gibbs sampling on $P(x | \theta_{t+1})$; therefore, $M (= eN)$ parallel Gibbs sampling is run to update the sample set.

The accuracy of the learning was measured using the MAE of the interactions:

$$\text{MAE}(t) = \frac{1}{|F_2|} \sum_{(i,j) \in F_2} |w_{i,j}^{(t)} - w_{i,j}^{\text{MLE}}|,$$

where $w_{i,j}^{\text{MLE}}$ is the value obtained from the exact MLE, and $w_{i,j}^{(t)}$ is the trained value at step $t$. Figures 7a and 7b present the MAEs against the update step $t$, where panel a is the result of the model-matched case and panel b is that of the model-mismatched case. In these figures, “1-SMCI” and “s2-SMCI” correspond respectively to the training based on the 1-SMCI and s2-SMCI methods. Here, “(fix)” means that the sample sets were fixed to the data set ($S = \mathbb{D}$), that is, the learning strategy proposed in the previous studies (Yasuda, 2015, 2018); the others are the proposed method combined with the PCD-like strategy in which $e$ is the data-extension rate. The proposed method significantly improved the accuracy in both cases. The accuracies in the model-mismatched case were worse than those in the model-matched case. However, the proposed method can reduce the accuracy degradation by increasing the value of $e$. In the learning using the s2-SMCI method, the sum regions, $A_t$ and $A_{i,j}$, were rearranged after every update step because,
Figure 7: MAEs of interactions in the (a) model-matched case and (b) model-mismatched case, versus the update step $t$. The plots are the average values over 200 experiments.

**Algorithm 2:** The Proposed Learning Algorithm for PBM.

1: **Input** training data set $\mathcal{D}$

2: Initialize the learning parameters: $\theta = \theta_0$

3: Initialize the sample set using the $e$-replicated $\mathcal{D}$: $\mathcal{S} = \mathcal{S}_0$

4: Set $t = 0$

5: **repeat**

6: Update the learning parameters using a gradient ascent method with $g_i^{\text{app}}(\mathcal{S}_t, \theta_t)$ and $g_{i,j}^{\text{app}}(\mathcal{S}_t, \theta_t)$, e.g.,

$$w^{(t+1)}_i \leftarrow w^{(t)}_i + \varepsilon g_i^{\text{app}}(\mathcal{S}_t, \theta_t),$$

$$w_{i,j}^{(t+1)} \leftarrow w_{i,j}^{(t)} + \varepsilon g_{i,j}^{\text{app}}(\mathcal{S}_t, \theta_t),$$

where $\varepsilon$ is the learning rate.

7: Update the sample set $\mathcal{S}_t$ to $\mathcal{S}_{t+1}$ using ($M$ parallel) $\kappa$-step Gibbs sampling on $P(x \mid \theta_{t+1})$ (starting from $\mathcal{S}_t$)

8: $t \leftarrow t + 1$

9: **until** a certain criterion is satisfied
owing to the $W$-ordering, the solution obtained from algorithm 1 could be changed by the influence of the parameter update.

Finally, we consider the computational cost of the proposed learning method based on the s2-SMCI methods, $C_{s2}$. This cost is generally $C_{s2} = O(T(\Gamma + M\Lambda + R))$, where $\Gamma$ is the cost for evaluating all gradients in equations 5.4 and 5.5, $\Lambda$ is the cost for the $k$-step Gibbs sampling for one sample point (i.e., $\Lambda = O(|F_2|)$), $R$ is the cost for the rearrangement procedure of the sum regions, and $T$ is the total number of updates. Because $R$ does not depend on $M$, it can be eliminated from the current cost estimation by assuming $M \gg n$. The cost for the evaluation of all $m_{i,j}(\mathcal{A}_{i,j}; \mathcal{S})$ using the s2-SMCI method of equation 4.10 is dominant in $\Gamma$; therefore, $\Gamma = O(MK|F_2|)$, where $K = |F_2|^{-1} \sum_{(i,j) \in F_2} |I_1(i,j)|$ which is the average cost for evaluating the sums in equations 4.12 and 4.13. Here, consider three cases: (1) a sparse graph, in which $|F_2| = O(n)$ and $|N_1(i,j)| = O(1)$ for all edges; (2) a partially dense graph, in which $|N_1(i,j)| = O(n)$ for $O(n^2)$ edges and $|N_1(i,j)| = O(1)$ for the remaining edges; and (3) a complete graph. In case 1, $K$ is obviously $O(1)$; therefore, $C_{s2} = O(TMn)$ is obtained. In case 2, because $|F_2| = O(n^2)$, $K \leq |F_2|^{-1} \sum_{(i,j) \in F_2} |I_1(i,j)| = O(1)$; $C_{s2} = O(TMn^2)$ is obtained. In case 3, because $|I_1(i,j)| = 1$ for all edges as mentioned in section 4.1, $C_{s2} = O(TMn^2)$ is obtained.

6 Summary and Future Studies

In this letter, we presented, two different contributions for PBM. The first contribution is a generalization of the original SMCI method (Yasuda, 2015), described in section 3. In the original SMCI method (that is, the $k$-SMCI method), the setting of the sum region was seriously limited; i.e., for a target region $\mathcal{T}$, the sum region must cover up to the $(k-1)$th-nearest-neighboring region, $\mathcal{R}_{k-1}(\mathcal{T})$, of the target region. The statistical accuracy bound of the $k$-SMCI method was proved (Yasuda, 2015; see theorem 1). However, a higher-order $k$-SMCI method cannot be applied in a dense graph because the size of the sum region can be $O(n)$ there. This study investigated a more flexible setting of the sum region and provided a statistical accuracy bound of the setting (see theorem 2). The proposed GSMCI method allows a flexible setting of the sum region, such as in the s2-SMCI method discussed in section 4.1. The statistical accuracy bounds of the $k$-SMCI and GSMCI methods were validated in generalized MRFs.

The second contribution of this study is a new algorithm for PBM learning, as described in section 5. The proposed learning method significantly improved the accuracies of learning in the model-matched and model-mismatched cases, as shown in Figure 7. The model-mismatched case is more important for practical applications. To use the proposed learning method, the structure of the graph must be fixed in advance; however, appropriate structures are usually unknown for real-world data sets.
Therefore, almost all cases will be a model-mismatched case in practical applications. The learning accuracy in the model-mismatched case was improved through the proposed method; however, it seems to be insufficient compared to that in the model-matched case. To apply the proposed method to a real-world problem, an alternative procedure (during the preprocessing or in the outer loop of the proposed method) used to infer the appropriate structure is needed, which is, for example, graph mining based on a sparse modeling approach (Rish & Grabarnik, 2014; Yasuda et al., 2019).

The learning method proposed in the previous study (Yasuda, 2015, 2018) is applicable to only fully visible PBMs, because the values of the variables in the sample region must be filled by the data set. The proposed learning method can be immediately applied to PBM learning with hidden variables, such as RBMs and its variants—gaussian-Bernoulli RBMs (Cho et al., 2011) and gaussian-spherical RBMs (Decelle & Furtlehner, 2020)—and DBMs, because the values of the (hidden) variables in the sample region are filled by the sample points obtained through Gibbs sampling. The application to RBMs and DBMs is considered an important area of future studies.

SMCI uses a sample set drawn from an MRF; therefore, the approximation accuracy of SMCI depends on the quality of the sampling algorithm. In the experiments described in this letter, we used the usual Gibbs sampling. Decelle and Krzakala (2014) proposed an effective sampling method, based on belief propagation, the idea of which is close to that of SMCI. We believe that a combination of both SMCI and this sampling method leads to more effective approximations, an important area for future research.

Appendix A: Proof of Theorem 2

From the definition, the asymptotic variance \( \nu(A) \) for any region \( A \) satisfying \( T \subseteq A \subseteq V \) is always greater than or equal to zero. In the following, the difference between the asymptotic variances \( \nu(U_1) \) and \( \nu(U_2) \), \( \epsilon := M(\nu(U_1) - \nu(U_2)) \), is considered, where \( T \subseteq U_1 \subseteq U_2 \). From equation 3.5, the difference is represented as

\[
E = \sum_{x_{\partial U_1}} \rho_{T}(U_1; x_{\partial U_1})^2 P(x_{\partial U_1}) - \sum_{x_{\partial U_2}} \rho_{T}(U_2; x_{\partial U_2})^2 P(x_{\partial U_2}). \tag{A.1}
\]

For the evaluation of equation A.1, \( \mathcal{Y} := U_2 \setminus U_1 \) is defined. The relation \( \partial U_1 \subseteq \partial U_2 \cup \mathcal{Y} \) is satisfied because

\[
\partial U_1 = \{i \mid i \in C \in \mathcal{F}(U_1), i \notin U_1\} \subseteq \{i \mid i \in C \in \mathcal{F}(U_2), i \notin U_1\} = \{i \mid i \in C \in \mathcal{F}(U_2), i \notin U_2\} \cup \{i \mid i \in C \in \mathcal{F}(U_2), i \in \mathcal{Y}\} = \partial U_2 \cup \mathcal{Y}. \tag{A.2}
\]
The relation
\[ \rho_T(U_2; x_{\tilde{a}tl_2}) = \sum_{x_{\tilde{a}tl_1}} \sum_{x_y} f(x_T) P(x_{\tilde{a}tl_1}, x_y | x_{\tilde{a}tl_2}) \]
\[ = \sum_{x_{\tilde{a}tl_1}} \sum_{x_y} f(x_T) P(x_{\tilde{a}tl_1} | x_y, x_{\tilde{a}tl_2}) P(x_y | x_{\tilde{a}tl_2}) \]
\[ = \sum_{xy} \rho_T(U_1; x_{\tilde{a}tl_1}) P(x_y | x_{\tilde{a}tl_2}) \]  \hspace{1cm} (A.3)
always holds because
\[ P(x_{\tilde{a}tl_1} | x_y, x_{\tilde{a}tl_2}) = P(x_{\tilde{a}tl_1} | x_{\tilde{a}tl_2}) \]
is satisfied owing to both the spatial Markov property of the HMRF and the relation in equation A.2. From equation A.3, the difference in equation A.1 can be rewritten as
\[ E = \sum_{xy} \sum_{x_{\tilde{a}tl_2}} (\rho_T(U_1; x_{\tilde{a}tl_1}) - \rho_T(U_2; x_{\tilde{a}tl_2}))^2 P(x_y, x_{\tilde{a}tl_2}). \]  \hspace{1cm} (A.4)
This equation indicates that \( E \) is always greater than or equal to zero. Therefore, it is proved that \( v_T(U_1) \geq v_T(U_2) \geq 0. \)

Appendix B: Derivations of SMCI Methods on PBMs

**B.1 Derivations of Equations 4.1 and 4.2.** In general, the 1-SMCI method is represented by
\[ m_T^{(1)}(S) = \frac{1}{M} \sum_{\ell=1}^{M} f(x_T) P(x_T | s^{(\ell)}_{N(i)(T)}). \]  \hspace{1cm} (B.1)
In the PBM in equation 2.2, the conditional distribution \( P(x_T | x_{N(i)(i)}) \) is represented as
\[ P(x_i | x_{N(i)(i)}) = \frac{\exp(\gamma_i(x_{N(i)(i)})x_i)}{z_i(\gamma_i(x_{N(i)(i)}))} \]  \hspace{1cm} (B.2)
when \( T = \{i\} \) and as
\[ P(x_i, x_j | x_{N(i)(i,j)}) = \frac{\exp(\gamma_{ij}(x_{N(i)(i,j)})x_i + \gamma_{ji}(x_{N(i)(i,j)})x_j + w_{ij}x_ix_j)}{z_{i,j}(\gamma_{ij}(x_{N(i)(i,j)}), \gamma_{ji}(x_{N(i)(i,j)}), w_{ij})}. \]  \hspace{1cm} (B.3)
when $T = \{i, j\}$, where

$$z_i(a) := \sum_{x_i \in X_i} \exp(ax_i), \quad z_{i,j}(a, b, c) := \sum_{x_i \in X_i} \sum_{x_j \in X_j} \exp(ax_i + bx_j + cx_ix_j).$$

Here, $\gamma_i(x_{N_1(i)})$ and $\gamma_{i,j}(x_{N_1(i), j})$ in equations B.2 and B.3 are defined in equations 4.3 and 4.4. Note that $\gamma_{i,j}(x_{N_1(i), j})$ does not depend on $x_i$ and $x_j$.

When $X_i = \{-1, +1\}$, the equations

$$\sum_{x_i \in X_i} \frac{\exp(ax_i)}{z_i(a)} = \tanh a \quad \text{(B.4)}$$

and

$$\sum_{x_i \in X_i} \sum_{x_j \in X_j} x_i x_j \frac{\exp(ax_i + bx_j + cx_ix_j)}{z_{i,j}(a, b, c)} = \tanh \left[ \text{atanh} \left\{ (\tanh a)(\tanh b) \right\} + c \right] \quad \text{(B.5)}$$

are obtained. Equations B.1 to B.5 lead to the 1-SMCl methods in equations 4.1 and 4.2.

**B.2 Derivations of Equations 4.9 and 4.10.** When $T = \{i\}$, equation 4.7 becomes

$$P(x_T \mid x_{\partial A}) \propto \exp \left( \xi_i(x_{\partial A}) x_i \right), \quad \text{(B.6)}$$

where $\xi_i(x_{\partial A})$ is defined in equation 4.11. Equation B.6 obtained by using the equation

$$\ln \cosh(a + bx) = x \text{atanh} \left\{ (\tanh a)(\tanh b) \right\} + \text{constant},$$

which is satisfied when $x \in \{-1, +1\}$.\(^1\) Using equations 4.8, B.4, and B.6, equation 4.9 is obtained.

When $T = \{i, j\}$, equation 4.7 becomes

$$P(x_T \mid x_{\partial A}) \propto \exp \left( \xi_{i,j}(x_{\partial A}) x_i + \xi_{j,i}(x_{\partial A}) x_j + \omega_{i,j}(x_{\partial A}) x_i x_j \right), \quad \text{(B.7)}$$

\(^1\)For $x \in \{-1, +1\}$, a function $f(x)$ is always represented as $f(x) = A + Bx$, where $A = \sum_{x \in \{-1, +1\}} f(x)/2$ and $B = \sum_{x \in \{-1, +1\}} x f(x)/2$. 


where \( \xi_{i,j}(x_{0,A}) \) and \( \omega_{i,j}(x_{0,A}) \) are defined in equations 4.12 and 4.13, respectively. Equation B.7 is obtained using the following equation:\(^2\)

\[
\ln \cosh(a + bx_i + cx_j) = x_i x_j \left\{ \text{atanh} \left( (\tanh b)(\tanh c) \right) + \frac{1}{4} \ln \frac{1 - (\tanh^2 b)(\tanh^2 (b + c))}{1 - (\tanh^2 a)(\tanh^2 (b - c))} \right\} \\
+ x_i \left\{ \text{atanh} \left( (\tanh a)(\tanh b) \right) + \frac{1}{4} \ln \frac{1 - (\tanh^2 c)(\tanh^2 (a + b))}{1 - (\tanh^2 a)(\tanh^2 (a - b))} \right\} \\
+ x_j \left\{ \text{atanh} \left( (\tanh a)(\tanh c) \right) + \frac{1}{4} \ln \frac{1 - (\tanh^2 b)(\tanh^2 (a + c))}{1 - (\tanh^2 b)(\tanh^2 (a - c))} \right\} \\
+ \text{constant},
\]

which is satisfied when \( x_i, x_j \in \{-1, +1\} \). Using equations 4.8, B.5, and B.7, equation 4.10 is obtained.

**Acknowledgments**

This work was partially supported by JSPS KAKENHI (Grant Numbers 15H03699, 18K11459, and 18H03303), JST CREST (Grant Number JPMJCR1402), and the COI Program from the JST (Grant Number JP-MJCE1312).

**References**

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, 9, 147–169.

Besag, J. (1975). Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society D (The Statistician)*, 24(3), 179–195.

Cho, K., Ilin, A., & Raiko, T. (2011). Improved learning of Gaussian-Bernoulli restricted Boltzmann machines. In *Proc. of the 12th International Conference on Artificial Neural Networks* (pp. 10–17). Berlin: Springer.

Decelle, A., & Furtlehner, C. (2020). Gaussian-spherical restricted Boltzmann machines. *Journal of Physics A: Mathematical and Theoretical*, 53(18), 184002.

Decelle, A., & Krzakala, F. (2014). Belief-propagation-guided Monte-Carlo sampling. *Phys. Rev. B*, 89(21), 214421.

Furtlehner, C. (2013). Approximate inverse Ising models close to a Bethe reference point. *J. Stat. Mech.: Theor. and Exp.*, 2013(9), P09020.

---

\(^2\)For \( x_i, x_j \in \{-1, +1\} \), a function \( f(x_i, x_j) \) is always represented as \( f(x_i, x_j) = A + Bx_i + Cx_j + Dx_i x_j \), where \( A = \sum_{x_i, x_j \in \{-1, +1\}} f(x_i, x_j) / 4 \), \( B = \sum_{x_i, x_j \in \{-1, +1\}} x_i f(x_i, x_j) / 4 \), \( C = \sum_{x_i, x_j \in \{-1, +1\}} x_j f(x_i, x_j) / 4 \), and \( D = \sum_{x_i, x_j \in \{-1, +1\}} x_i x_j f(x_i, x_j) / 4 \).
Halldórsson, M. M., & Radhakrishnan, J. (1997). Greed is good: Approximating independent sets in sparse and bounded-degree graphs. *Algorithmica, 18*, 145–163.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation, 14*(8), 1771–1800.

Hyvärinen, A. (2006). Consistency of pseudo likelihood estimation of fully visible Boltzmann machines. *Neural Computation, 18*(10), 2283–2292.

Hyvärinen, A. (2007). Some extensions of score matching. *Computational Statistics and Data Analysis, 51*(5), 2499–2512.

Johnson, J. K., Oyen, D., Chertkov, M., & Netrapalli, P. (2016). Learning planar Ising models. *Journal of Machine Learning Research, 17*(215), 1–26.

Kappen, H. J., & Rodríguez, F. B. (1998). Efficient learning in Boltzmann machines using linear response theory. *Neural Computation, 10*(5), 1137–1156.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing, 11*(2), 125–139.

Ricci-Tersenghi, F. (2012). The Bethe approximation for solving the inverse Ising problem: A comparison with other inference methods. *Journal of Statistical Mechanics: Theory and Experiment, 2012*(8), P08015.

Rish, I., & Grabarnik, G. (2014). *Sparse modeling: Theory, algorithms, and applications*. Boca Raton, FL: CRC Press.

Salakhutdinov, R., & Hinton, G. E. (2009). Deep Boltzmann machines. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics* (pp. 448–455). Berlin: Springer.

Sejnowski, T. J. (1986). Higher-order Boltzmann machines. In *AIP Conference Proceedings 151, Neural Networks for Computing* (pp. 398–403).

Sessak, V., & Monasson, R. (2009). Small-correlation expansions for the inverse Ising problem. *Journal of Physics A: Mathematical and Theoretical, 42*(5), 055001.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (vol. 1, 194–281). Cambridge, MA: MIT Press.

Sohl-Dickstein, J., Battaglino, P. B., & DeWeese, M. R. (2011). New method for parameter estimation in probabilistic models: Minimum probability flow. *Physical Review Letters, 107*, 220601.

Tanaka, T. (1998). Mean-field theory of Boltzmann machine learning. *Phys. Rev. E, 58*, 2302–2310.

Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proc. of the 25th International Conference on Machine Learning* (pp. 1064–1071). New York: ACM.

Yasuda, M. (2015). Monte Carlo integration using spatial structure of Markov random field. *Journal of the Physical Society of Japan, 84*(3), 034001.

Yasuda, M. (2018). Learning algorithm of Boltzmann machine based on spatial Monte Carlo integration method. *Algorithms, 11*(4), 42.

Yasuda, M., & Horiguchi, T. (2006). Triangular approximation for information Ising model and its application to Boltzmann machine. *Physica A, 368*, 83–95.

Yasuda, M., Kataoka, S., & Tanaka, K. (2012). Inverse problem in pairwise Markov random fields using loopy belief propagation. *Journal of the Physical Society of Japan, 81*(4), 044801.
Yasuda, M., Katou, K., Mikuni, Y., Yokoyama, Y., Harada, T., Tanaka, A., & Yokoyama, M. (2019). Detection of human-interaction network using Markov random field. *Nonlinear Theory and Its Applications, 10*(4), 485–495.

Yasuda, M., & Tanaka, K. (2009). Approximate learning algorithm in Boltzmann machines. *Neural Computation, 21*(11), 3130–3178.

Yedidia, J. S., Freeman, W. T., & Weiss, Y. (2005). Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory, 51*(7), 2282–2312.

Received September 14, 2020; accepted October 28, 2020.