Detection of human-interaction network using Markov random field

Muneki Yasuda\textsuperscript{1}, Kouta Katou\textsuperscript{1}, Yoshitaka Mikuni\textsuperscript{2}, Yuuki Yokoyama\textsuperscript{3}, Tomochika Harada\textsuperscript{1}, Atushi Tanaka\textsuperscript{1}, and Michio Yokoyama\textsuperscript{1}

\textsuperscript{1} Graduate School of Science and Engineering, Yamagata University
4-3-16 Yonnan, Yonezawa, Yamagata 992-8510, Japan

\textsuperscript{2} Graduate School of System and Information Engineering, University of Tsukuba, 1-1-1 Tennoudai, Tsukuba, Ibaraki 305-8573, Japan

\textsuperscript{3} ALBERT Inc.

\textsuperscript{a}) muneki@yz.yamagata-u.ac.jp

Received January 10, 2019; Published October 1, 2019

Abstract: Discovering network structures among social actors is one of the most fundamental issues related to social networks. In this paper, we propose a novel and effective algorithm for building a human-interaction network from the location data of individuals gathered by sensors such as the GPS system. We model the problem using Markov random field. The proposed approach combines statistical machine learning with sparse modeling, i.e., the $L_1$ regularized maximum likelihood approach. We demonstrate the validity of our method through numerical experiments using artificial location data generated from a simulator of quasi-human-transfer.

Key Words: social network, human-interaction network, graph mining, Markov random field, sparse modeling

1. Introduction
Research on social networks is important in diverse fields such as sociology, social psychology, and engineering. Social network analysis treats a network as comprising a set of (social) actors, such as individuals or organizations, and a set of links (or ties) among the actors. In many cases, the structures of social networks are not trivial, and hence, discovering those structures is a fundamental issue in social network research.

In this paper, we propose a stochastic model based on Markov random field (MRF) and a practical algorithm for discovering structures based on the statistical machine learning approach \cite{1} and the sparse modeling approach \cite{2}. We attempt to discover a human-interaction network among individuals using location data gathered by sensors such as the GPS system. We consider a human-interaction network in which a link represents an interaction between two individuals, and assume two kinds of interactions: friendly interaction and adversarial interaction. Two individuals having a friendly interaction frequently stay close to each other, while those involved in an adversarial interaction...
Fig. 1. Schematic illustration of the problem. In a human-interaction network, links with ‘+’ represent friendly interactions and links with ‘−’ represent the adversarial interactions.

frequently stay far away from each other. The scheme of our problem is illustrated in Fig. 1. We model the problem based on an MRF and discover human-interaction networks via statistical machine learning of the MRF. In our MRF modeling, friendly or adversarial interactions are represented as attractive or repulsive forces, respectively. Our attempt strongly relates to the graph mining problem, for which several researchers have proposed algorithms based on MRFs, e.g. graphical lasso [3] and decimation algorithm [4, 5]. The differences between our model and those proposed in these studies [3–5] are as follows. The graphical lasso basically treats a Gaussian MRF (GMRF), whereas, our model is not based on GMRF. The models proposed in Refs. [4, 5] can treat only binary one-dimensional data, while our model treats discrete or continuous two-dimensional data, because we need to use the location data of individuals. Furthermore, in the learning algorithms in Refs. [4, 5], the maximum pseudo-likelihood method [6] was used, while our method uses the spatial Monte Carlo integration (SMCI) method [7]. It is known that the SMCI method is more effective than the maximum pseudo-likelihood method [8].

This paper is an extension of our previous study [9], and it includes some new results: (i) the previous method was formulated for only discrete data, whereas the reformulated method in this paper is applicable to continuous data; (ii) this paper includes a comparison with an analysis based on the Pearson correlation coefficient (PCC) (Sec. 2.4); and (iii) only numerical results for a small toy example were shown in the previous study, whereas the present paper includes new numerical results with large size location data generated from a simulator of quasi-human-transfer (Sec. 4).

The remainder of this paper is organized as follows. In Sec. 2, we propose an MRF representing the distribution of the locations of individuals. The coupling parameters in the MRF directly correspond to the human-interaction network under analysis. Subsequently, we propose an effective learning algorithm for the proposed model based on the SMCI method [7] in Sec. 2.2. In Sec. 2.3, we demonstrate that the proposed learning algorithm provides better estimates than a learning algorithm based on the contrastive divergence (CD) method [10]. In Sec. 2.4, we compare our method with an analysis based on the PCC and demonstrate the merits of our approach. In Sec. 3, we combine the proposed learning algorithm with the sparse modeling approach, i.e., $L_1$ regularized maximum likelihood approach [2] and propose an algorithm to detect the structure of the human-interaction network. We demonstrate the validity of our method by conducting numerical experiments using a toy example in Sec. 3.2. In Sec. 4, we construct a simulator of quasi-human-transfer and show the performance of our method for location data generated from the simulator. Finally, the conclusions are presented in Sec. 5.

2. Markov random field of human-interaction network

Suppose that there are $n$ actors (i.e., individuals) in the region of interest, and we want to detect an interaction network in the form of an undirected graph among them. In the interaction network, $G(V, E)$, each actor corresponds to a node $i \in V := \{1, 2, \ldots, n\}$, and two actors $i$ and $j$ are connected by an undirected link $(i, j) \in E$ if they have an interaction with each other. We consider two kinds of links, $E = E^+ \cup E^-$: friendly links, $(i, j) \in E^+$, and adversarial links, $(i, j) \in E^-$. We define an MRF for the locations of the $n$ actors, which is expressed as

$$P(\mathbf{r} | \mathbf{c}) := \frac{1}{Z(\mathbf{c})} \exp \left( - \frac{1}{2} \sum_{i<j} c_{ij} ||r_i - r_j||^2 \right),$$  

(1)
where \( r_i := (x_i, y_i)^T \) is the location vector (or the 2D coordinate) of actor \( i \) and \( r := \{ r_i \mid i \in V \} \) is the set of location vectors. The elements in the location vectors take the values in the sample spaces \( X \) and \( Y \), respectively, namely, \( x_i \in X \) and \( y_i \in Y \). The summation in the exponent in Eq. (1) is the sum over all distinct pairs of actors, i.e., \( \sum_{i<j} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \). \( \| \cdot \|_2 \) is the \( L_2 \) norm (or the Euclidean distance) of the assigned vector, and \( Z(c) \) is the partition function, where \( c = \{c_{ij} \in (-\infty, +\infty) \mid i < j \in V \} \) are the connectivity parameters. The connectively parameters are symmetric with respect to their indices, namely, \( c_{ij} \) is identified as \( c_{ji} \) for \( i < j \).

The stochastic model in Eq. (1) can be regarded as a model for gas molecules (packed in a 2D space) in statistical mechanics. The connectivity parameter, \( c_{ij} \), represents the strength of the interaction between two actors \( i \) and \( j \) such that, (i) \( c_{ij} > 0 \) corresponds to an attractive force between them (namely, the friendly interaction), (ii) \( c_{ij} < 0 \) corresponds to a repulsive force between them (namely, the adversarial interaction), and (iii) if \( c_{ij} = 0 \), there is no interaction between them. The connectivity parameters simply correspond to the structure of the interaction network.

2.1 Maximum likelihood estimation for connectivity parameters

The optimal values of the connectivity parameters are obtained from the maximum likelihood estimation (MLE) by using the \( T \) location data of the \( n \) actors: \( \mathcal{D} := \{ r^{(t)} \mid t = 1, \ldots, T \} \), where \( r^{(t)} := \{ (x_i^{(t)}, y_i^{(t)})^T \mid i \in V \} \) is the \( t \)th location data of the \( n \) actors.

Our MRF in Eq. (1) can be decomposed as

\[
P(r \mid c) = P_X(x \mid c)P_Y(y \mid c),
\]

where

\[
P_X(x \mid c) := \frac{1}{Z_X(c)} \exp \left( -\frac{1}{2} \sum_{i<j} c_{ij}(x_i - x_j)^2 \right),
\]

\[
P_Y(y \mid c) := \frac{1}{Z_Y(c)} \exp \left( -\frac{1}{2} \sum_{i<j} c_{ij}(y_i - y_j)^2 \right).
\]

\( Z_X(c) \) and \( Z_Y(c) \) are the partition functions defined by

\[
Z_X(c) := \int_{X^n} \exp \left( -\frac{1}{2} \sum_{i<j} c_{ij}(x_i - x_j)^2 \right) dx,
\]

\[
Z_Y(c) := \int_{Y^n} \exp \left( -\frac{1}{2} \sum_{i<j} c_{ij}(y_i - y_j)^2 \right) dy.
\]

where \( \int_{X^n} \cdots dx \) and \( \int_{Y^n} \cdots dy \) are the multiple integrations over \( x \in X^n \) and \( y \in Y^n \), respectively. Note that when \( X \) and \( Y \) are discrete spaces, the multiple integrations become multiple summations: \( \int_{X^n} \cdots dx \to \sum_{x \in X^n} \cdots \) and \( \int_{Y^n} \cdots dy \to \sum_{y \in Y^n} \cdots \), respectively. This decomposition means that \( x \) and \( y \) are statistically independent, and therefore, we can treat them separately. Equations (3) and (4) represent a type of Boltzmann machine [11]. It is noteworthy that \( X = (-\infty, +\infty) \) or \( Y = (-\infty, +\infty) \) are not allowed. When \( X = (-\infty, +\infty) \) (or \( Y = (-\infty, +\infty) \)), \( P_X(x \mid c) \) (or \( P_Y(y \mid c) \)) is not well-defined, because the partition function \( Z_X(c) \) (or \( Z_Y(c) \)) diverges. Thus, our MRF cannot be a GMRF.

From Eq. (2), the log-likelihood of our model, \( l_{D}(c) := T^{-1} \sum_{t=1}^{T} \ln P(r^{(t)} \mid c) \), is expressed as

\[
l_{D}(c) = \frac{1}{T} \sum_{t=1}^{T} \ln P_X(x^{(t)} \mid c) + \frac{1}{T} \sum_{t=1}^{T} \ln P_Y(y^{(t)} \mid c).
\]

The gradient of Eq. (5) with respect to \( c_{ij} \) is

\[
\frac{\partial l_{D}(c)}{\partial c_{ij}} = -\frac{1}{2T} \sum_{t=1}^{T} (x_i^{(t)} - x_j^{(t)})^2 + \frac{1}{2} \mathbb{E}_X [(x_i - x_j)^2 \mid c].
\]
\[-\frac{1}{2T} \sum_{t=1}^{T} (y_i^{(t)} - y_j^{(t)})^2 + \frac{1}{2} \mathbb{E}_Y [(y_i - y_j)^2 \mid c], \tag{6}\]

where

\[ \mathbb{E}_X [f(x) \mid c] := \int_{X^n} f(x) P_X(x \mid c) dx, \quad \mathbb{E}_Y [f(y) \mid c] := \int_{Y^n} f(y) P_Y(y \mid c) dy \]

are the expectations of the corresponding Boltzmann machines. Since the log-likelihoods of the Boltzmann machines (i.e., the first and second terms in Eq. (5)) are concave with respect to \(c\), the log-likelihood in Eq. (5) is also concave. Therefore, in principle, optimal \(c\) are obtained using a gradient ascent method with the gradients in Eq. (6) and the optimal solution is unique. From the definition, when \(X = Y\), \(\mathbb{E}_X [f(x) \mid c]\) is equivalent to \(\mathbb{E}_Y [f(y) \mid c]\), and consequently, the gradient in Eq. (6) is reduced to

\[ \frac{\partial \varphi}{\partial c_{ij}} = -\frac{1}{2T} \sum_{t=1}^{T} (y_i^{(t)} - y_j^{(t)})^2 - \frac{1}{2T} \sum_{t=1}^{T} (y_i^{(t)} - y_j^{(t)})^2 + \mathbb{E}_X [(x_i - x_j)^2 \mid c]. \]

However, the gradients cannot be computed owing to the multiple integrations (or summations) over \(x\) and \(y\) in the expectations. The computational time grows exponentially with increase in \(n\). Some approximate methods have been developed for Boltzmann-machine learning. The CD method [10] is one of the most commonly used approximation methods in Boltzmann-machine learning. It is known that the performance of the CD method is statistically the same as that of the maximum pseudo-likelihood method in (fully-visible) Boltzmann-machine learning [13].

### 2.2 Spatial Monte Carlo integration method

The SMCI method [7] can also be used in our learning problem. It has been proven that the SMCI method is effective in Boltzmann-machine learning [8]. In this section, we apply the first-order SMCI (1-SMCI) method proposed in Ref. [7] to the Boltzmann-machine learning in our problem.

In the 1-SMCI method, the expectation, \(\mathbb{E}_X [(x_i - x_j)^2 \mid c]\), is approximated by

\[ e_{ij}^X(c, S_X) := \frac{1}{S} \sum_{s=1}^{S} \int_{X^2} (x_i - x_j)^2 P_X(x_i, x_j \mid x_{-(i,j)}^s) dx_i dx_j \tag{7}\]

using \(S\) sampled points, \(S_X := \{x^{(1)}, x^{(2)}, \ldots, x^{(S)}\}\), drawn from \(P_X(x \mid c)\). Here, \(x_{-(i,j)} := \{x_k \mid k \in V \setminus \{i, j\}\}\) and

\[ P_X(x_i, x_j \mid x_{-(i,j)}, c) = \frac{P_X(x \mid c)}{\int_{X^2} P_X(x \mid c) dx_i dx_j} = \frac{1}{Z_{ij}(x_{-(i,j)}, c)} \exp \left( -\frac{1}{2} c_{ij}(x_i - x_j)^2 - \frac{1}{2} \sum_{k \in V \setminus \{i,j\}} c_{ik}(x_i - x_k)^2 - \frac{1}{2} \sum_{l \in V \setminus \{i,j\}} c_{jl}(x_j - x_l)^2 \right) \tag{8}\]

is the conditional distribution of \(P_X(x \mid c)\), where

\[ Z_{ij}(x_{-(i,j)}, c) := \int_{X^2} \exp \left( -\frac{1}{2} c_{ij}(x_i - x_j)^2 - \frac{1}{2} \sum_{k \in V \setminus \{i,j\}} c_{ik}(x_i - x_k)^2 - \frac{1}{2} \sum_{l \in V \setminus \{i,j\}} c_{jl}(x_j - x_l)^2 \right) dx_i dx_j \]

is the partition function. Note that the expectation, \(\mathbb{E}_Y [(y_i - y_j)^2 \mid c]\), is also approximated by \(e_{ij}^Y(c, S_Y)\), in the same manner as Eq. (7), where \(S_Y := \{y^{(1)}, y^{(2)}, \ldots, y^{(S)}\}\) is the set of \(S\) sampled points drawn from \(P_Y(y \mid c)\).

In Eq. (6), by replacing \(\mathbb{E}_X [(x_i - x_j)^2 \mid c]\) and \(\mathbb{E}_Y [(y_i - y_j)^2 \mid c]\) by \(e_{ij}^X(c, S_X)\) and \(e_{ij}^Y(c, S_Y)\), respectively, the gradients for the Boltzmann-machine learning based on the 1-SMCI method are obtained. It is noteworthy that the sampled points, \(S_X\) and \(S_Y\), can be replaced by the given data set \(D\) when the number of data, \(T\), is not very small [7]. In the numerical experiments shown in the following sections, we use \(D\) instead of \(S_X\) and \(S_Y\) in the computations of \(e_{ij}^X(c, S_X)\) and \(e_{ij}^Y(c, S_Y)\). The computational complexity of the Boltzmann-machine learning based on the 1-SMCI method is \(O(n^2T)\) (see, Appendix A), which is the same as that of the CD method.
Fig. 2. Toy example of the network structure, $G(V,E)$, of the generative model. The three positive links correspond to friendly links ($c_{ij} = 0.2$) and the negative link corresponds to an adversarial link ($c_{ij} = -0.2$).

Table I. The mean absolute errors between the exact solution obtained by the MLE and the approximate solutions obtained from the CD method and the proposed method (the 1-SMCI method). The values in each cell are the average and standard deviation over 400 experiments.

| method      | data size | mean-absolute error      |
|-------------|-----------|--------------------------|
| CD method [10] | $T = 100$ | 0.020 ± 0.0048            |
|             | $T = 1000$| 0.0062 ± 0.0014           |
| proposed    | $T = 100$ | 0.0020 ± 0.0012           |
|             | $T = 1000$| 0.00045 ± 0.00019         |

2.3 Numerical experiment with toy example: CD method vs 1-SMCI method

To check the accuracy of the Boltzmann-machine learning based on the 1-SMCI method, we used a generative model corresponding to a network with $n = 5$, which is shown in Fig. 2. The generative model is created by setting $c_{ij} = 0.2$ for $(i,j) \in E_+$, $c_{ij} = -0.2$ for $(i,j) \in E_-$, and $c_{ij} = 0$ for $(i,j) \not\in E$ in Eq. (1). The data set, $D$, was generated from the generative model using the Gibbs sampling method [14]. Here, we set $X = \{0,1,2\}$ and $Y = \{0,1,2,3\}$. Using the data set, we trained our model in Eq. (1) by using the MLE explained in Sec. 2.1. Since the size of the system is small in the current problem, we can obtain the exact solution of the MLE, $e_{\text{MLE}}$.

Table I shows the mean-absolute errors between the exact solution and the approximate solutions, $e_{\text{app}}$, obtained from the CD method and from the proposed 1-SMCI method, i.e., $\frac{2}{n(n-1)} \sum_{i<j} |e_{ij}^{\text{MLE}} - e_{ij}^{\text{app}}|$. The exact and approximate solutions were obtained by using the simple gradient ascent method with the learning rate $\eta = 0.1$. Note that, in the CD method, we set the maximum iteration number of the gradient ascent method to 1000. The result in Table I shows that the performance of the proposed method is much better than that of the CD method.

2.4 Comparison with Pearson correlation coefficient

The PCC is one of the most popular methods used to obtain correlations between two different elements. In this section, we compare our method with the PCC by using a simple example. Consider a generative model in Eq. (1), which consists of only three actors. In the generative model, the two friendly links are put between actors 1 and 2 ($c_{12} = 0.3$) and between actors 2 and 3 ($c_{23} = 0.5$). There are no links between actors 1 and 3 ($c_{13} = 0$). Here, we set $X = Y = \{0,1,2,3,4\}$.

Using $D$ ($T = 500$) generated from the generative model, we compared the solution obtained by our method (the 1-SMCI method) with the PCCs between two actors. The PCC between actors $i$ and $j$ is defined as

$$\xi_{ij} := \frac{\sum_{t=1}^{T} (r_i^{(t)} - R_i) (r_j^{(t)} - R_j)}{\sqrt{\sum_{t=1}^{T} ||r_i^{(t)} - R_i||^2} \sqrt{\sum_{t=1}^{T} ||r_j^{(t)} - R_j||^2}},$$

where $R_i := T^{-1} \sum_{t=1}^{T} r_i^{(t)}$. We hope that the PCCs corresponding to the friendly links, (1, 2) and (2, 3), take positive values and the PCC corresponding to the null link, (1, 3), takes a negligible small value. The result is shown in Table II. In the result, the PCC corresponding to (1, 3) is not small. This is the so-called “spurious correlation”. Although actors 1 and 3 do not have a direct interaction, there is a positive correlation between them via actor 2. In contrast, the value of $c_{13}$ inferred using our method is small, which means that our method can suppress the spurious correlation problem.
This is because the PCC, $\xi_{ij}$, in Eq. (9) is determined by the relationship between only two actors $i$ and $j$, whereas our method determines the value of $c_{ij}$ via the relationships among all the actors.

### 3. Combining sparse Modeling Approach

The goal of this study is to discover the network structure among $n$ actors. In other words, we want to know whether each $c_{ij}$ is zero or not. Unfortunately, the solutions obtained from the MLE are not sparse, i.e., they do not include just $c_{ij} = 0$ (see, Table II for example). To enforce sparsity in solutions to the MLE, we apply the sparse modeling approach [2] to the current problem.

#### 3.1 Maximum likelihood estimation with $L_1$ regularization

According to the context of the sparse modeling approach, we define a regularized log-likelihood as

$$L_D(c; \lambda) := l_D(c) - \lambda \sum_{i<j} |c_{ij}|,$$

where $l_D(c)$ is the original log-likelihood defined in Sec. 2.1, and $\lambda > 0$ is a hyper-parameter, which is sometimes called the regularization coefficient. For a specific $\lambda$, we maximize $L_D(c; \lambda)$ instead of the original log-likelihood with respect to $c$. The solution to the maximization is sparse when $\lambda$ is set to an appropriate value. This type of regularization is known as $L_1$ regularization, and it plays a central role in the recent sparse modeling approach. It is noteworthy that since the regularization term is (strictly) concave, $L_D(c; \lambda)$ is also (strictly) concave. This means that the optimal solution of the maximization of $L_D(c; \lambda)$ is unique.

In the maximization of the regularized log-likelihood in Eq. (10), one cannot directly use a standard gradient method, because it includes an indifferentiable function, i.e., the absolute function. The forward backward splitting (FOBOS) algorithm [15] allows maximization of the regularized log-likelihood by using a gradient-based method. In the FOBOS algorithm, the gradients in Eq. (6) must be computed. As in Sec. 2.2, the gradients can be approximated by the 1-SMCI method. According to the FOBOS algorithm, the connectivity parameters are updated by the following two step updates:

$$\hat{c}_{ij} \leftarrow c_{ij}^{\text{old}} + \eta \frac{\partial l_D(c^{\text{old}})}{\partial c_{ij}},$$

$$c_{ij}^{\text{new}} \leftarrow \text{sign}(\hat{c}_{ij}) \max (0, |\hat{c}_{ij}| - \hat{\eta} \lambda),$$

where

$$\text{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}.$$ 

Here, $\eta$ and $\hat{\eta}$ in Eqs. (11) and (12) are referred to as the forward learning rate and the backward learning rate, respectively. In the numerical experiments in this paper, we set the backward learning rate to one.

After maximizing the regularized log-likelihood in Eq. (10) by using the FOBOS + 1-SMCI method (the proposed method), we create the interaction network, $G(V,E)$, from the obtained solution, $c$, as follows: (i) we put the friendly link, $(i, j) \in E^+$, between actors $i$ and $j$, when $c_{ij} > 0$; (ii) we put the adversarial link, $(i, j) \in E^-$, between actors $i$ and $j$, when $c_{ij} < 0$; and (iii) we do not put links between actors $i$ and $j$, when $c_{ij} = 0$. 

---

**Table II.** The values of the true $c$ (i.e., $c$ in the generative model), the inferred $c$ obtained by the 1-SMCI method, and the PCCs. The values in each cell are the average and standard deviation over 500 experiments.

| link $(i, j)$ | true $c_{ij}$ | inferred $c_{ij}$ | PCC $\xi_{ij}$ |
|--------------|--------------|------------------|-----------------|
| (1, 2)       | 0.3          | $0.30 \pm 0.027$ | 0.42 $\pm 0.025$ |
| (2, 3)       | 0.5          | $0.50 \pm 0.032$ | 0.57 $\pm 0.020$ |
| (1, 3)       | 0            | $0.00016 \pm 0.022$ | 0.24 $\pm 0.030$ |
Table III. Interpretation of Cohen’s kappa coefficient [16].

| value of $\kappa$ | strength of agreement |
|-------------------|-----------------------|
| $< 0$             | poor                  |
| 0–0.2             | slight                |
| 0.21–0.4          | fair                  |
| 0.41–0.6          | moderate              |
| 0.61–0.8          | substantial           |
| 0.81–1            | almost perfect        |

Fig. 3. Cohen’s kappa coefficient for various $\lambda$. (a) Result when $T = 100$ and (b) result when $T = 500$. These plots show the average over 400 experiments. The setups of the learning rate and maximum iteration number of the CD method were the same as those of the experiments in Sec. 2.3.

3.2 Numerical Experiment with toy example: sparse modeling approach

In this section, we demonstrate the performance of the method proposed in the previous section using numerical experiments. In these experiments, we used the generative model introduced in Sec. 2.3, and generated $D$ from the generative model as described in Sec. 2.3. Using the generated data set, we inferred the interaction network, $G(V,E)$, according to the procedure presented in the previous section. Since we know the network structure of the generative model, which is as shown in Fig. 2, we could evaluate how good the inferred network was. We measured the degree of agreement between the true network (in Fig. 2) and the inferred network using Cohen’s kappa coefficient $\kappa$. This coefficient measures the agreement between two raters who each classify $M$ items into $C$ categories. In this case, the two raters are the true network and the inferred network, each of which classify all the distinct pairs of actors, namely $M = n(n - 1)/2$, into $C = 3$ categories: (i) friendly link, (ii) adversarial link, or (iii) null link. Cohen’s kappa coefficient is less than or equal to one, and values in the range $0.81 \leq \kappa \leq 1$ are interpreted as almost perfect agreement (see, Table III).

The plots of the kappa coefficients for various $\lambda$ are shown in Fig. 3. The definition of $\kappa$ is given in Eq. (B-1). In the experiments, we regarded $c_{ij} = 0$ if $|c_{ij}| < 10^{-6}$. For comparison, the results obtained using the CD method instead of the 1-SMCI method are also plotted. The results at $\lambda = 0$ correspond to the solutions of the MLE without $L_1$ regularization (see, Eq. (10)). The kappa coefficients obtained using our method were larger than 0.81 for some values of $\lambda$, and they were almost always larger than those obtained using the CD method. The performance degraded for large values of $\lambda$. This is because, when $\lambda$ is too large, the second term in Eq. (10) is dominant over the first term, and therefore, almost all $c_{ij}$s are zero at the optimal point of the regularized log-likelihood.

4. Numerical experiment using data generated from simulator of quasi-human-transfer

In this section, we demonstrate the performance of our method (i.e., the FOBOS + 1-SMCI method) using the location data generated from a simulator.
In the experiment, we set $n = 4$.  

**4.1 Definition of simulator of quasi-human-transfer**

The setup of the simulator is as follows. We set $X = Y = [0, 1]$ and put $n$ actors in the square region $[0, 1] \times [0, 1]$. All the actors synchronously move according to a certain dynamics. Suppose that the location of actor $i$ at time $t$ is $r_i(t) = (x_i(t), y_i(t))^T$. The location of the actor at time $(t + 1)$ is determined by

$$r_i(t + 1) = r_i(t) + l_i \left\{ \rho_i(t) \left( \frac{\cos \theta_i(t)}{\sin \theta_i(t)} \right) + (\varepsilon_x \varepsilon_y) \right\},$$

where $l_i \in [0, +\infty)$ is the magnitude of the movement of actor $i$, and $\varepsilon_x$ and $\varepsilon_y$ are the additive white Gaussian noise with variance $\sigma^2$ (see, Fig. 4(a)). $\theta_i(t)$ is the angle of the movement, $-\pi/2 \leq \theta_i(t) \leq 3\pi/2$, which is defined by

$$\theta_i(t) := \begin{cases} \tan^{-1} \frac{v_i(t)}{\dot{r}_i(t)} + \pi & (h_i(t) > 0) \\ \tan^{-1} \frac{v_i(t)}{\dot{r}_i(t)} & (h_i(t) < 0) \end{cases},$$

where

$$h_i(t) := \sum_{j \in V \backslash \{i\}} \hat{c}_{ij}(x_j(t) - x_i(t)), \quad v_i(t) := \sum_{j \in V \backslash \{i\}} \hat{c}_{ij}(y_j(t) - y_i(t)).$$

When $h_i(t) = v_i(t) = 0$, we let $\theta_i(t)$ be zero. Here, $\hat{c}_{ij}$ is the connectivity parameter between actors $i$ and $j$ in the simulator, which has the same meaning as $c_{ij}$ in our MRF. When $h_i(t) = v_i(t) = 0$, $\rho_i(t) = 0$, and $\rho_i(t) = 1$ elsewhere. According to the dynamics in Eq. (13), the actors having a friendly interaction, $\hat{c}_{ij} > 0$, tend to be close to each other, whereas those having an adversarial interaction, $\hat{c}_{ij} < 0$, tend to be far away from each other. Consider that there are only two actors, $i$ and $j$, and $\hat{c}_{ij} > 0$, in which the angle of $\theta_i(t)$ in Eq. (14) trends in the direction of actor $j$ (see, Fig. 4(b)). An actor having no interactions moves like a random walk, because $\rho_i(t)$ is always zero.

We obtained $\mathcal{D}$ from the simulator in the following way. The locations of all the actors were randomly initialized, and they moved from their initial locations in accordance with the dynamics in Eq. (13). After they moved $K$ times, we sampled their locations at that time as the location data $r^{(1)}$. We repeated the above procedure (the initializing and $K$ times moving) $T$ times and obtained the data set $\mathcal{D} = \{r^{(1)}, r^{(2)}, \ldots, r^{(T)}\}$. In the following experiments, we set $l_i = 0.001$, $\sigma = 5$, and $K = 400$.

**4.2 Experiment**

In the experiment, we set $n = 100$ and the connectivity parameters in the simulator were set as

$$\hat{c}_{ij} = \begin{cases} 0.1 & \text{with probability } p = 2/(n-1) \\ -0.1 & \text{with probability } p = 2/[5(n-1)] \\ 0 & \text{elsewhere} \end{cases}.$$
This means that \( n \) links were friendly links and \( n/5 \) links were adversarial links on average. The number of location data was set to \( T = 300 \). The plots of the kappa coefficients between the structures of \( \hat{c} \) and of \( c \) for various \( \lambda \) are shown in Fig. 5. For comparison, the results obtained using the CD method instead of the 1-SMCI method are also plotted. The learning rate and maximum iteration number of the CD method were 0.4 and 300, respectively. Since \( x \) and \( y \) are continuous, we have to perform the double integrations in order to use the 1-SMCI method (see, Eq. (7)). We implemented the integrations by using the Gauss-Laguerre quadrature with 3-point Gaussian quadrature rule. For the appropriate \( \lambda \), the kappa coefficients obtained using our method (i.e., the FOBOS + 1-SMCI method) were larger than 0.81, and they were much higher than those obtained using the CD method.

5. Conclusion and some remarks

In this paper, we proposed a novel and effective algorithm to derive the human-interaction network from location data of actors by combining the SMCI method with the sparse modeling approach. Our method provided much better results than those obtained using the CD method, and in our numerical experiments, the human-interaction network was detected with high accuracy when the value of the regularization parameter was appropriate.

Although our method was formulated based on location data in a 2D space, an extension to a 3D space is straightforward. If the location vector of actor \( i \) is extended to \( r_i := (x_i, y_i, z_i) \), our MRF will be decomposed as

\[
P(r | c) = P_X(x | c)P_Y(y | c)P_Z(z | c),
\]

where \( Z \) is the sample space of \( z_i \) (cf. Eq. (2)). This means that \( z \) are statistical independent from \( x \) and \( y \), and therefore, we can treat \( x \), \( y \), and \( z \) independently. In a similar way, an extension to higher-dimensional data is also straightforward.

In order to obtain sparse estimates of \( c \), we imposed the \( L_1 \) penalty to the log-likelihood function (cf. Eq. (10)). The \( L_1 \) penalty typically leads to sparse estimates; however, it tends to suppress the magnitude of the estimates, because it penalizes all the connectivity parameters that are not just zero. The use of the smoothly clipped absolute deviation (SCAD) [17] can reduce the suppression of magnitude, and it may lead to better estimates of \( c \).

However, an important issue in the current approaches, especially, that of finding an optimal value of the regularization parameter is not addressed and remains an open problem. We expect that the empirical Bayes (or the evidence approximation) [1, 18] allows to solve this problem.

Acknowledgments

This work was partially supported by JSPS KAKENHI (Granted Numbers 15K00330, 15H03699, 17K01781, 17H01729, 18K11459, and 18H03303), by JST CREST (Grant Number JPMJCR1402), by the Center of Innovation Program from the JST, and by MIC SCOPE (Strategic Information and Communications R&D Promotion Programme); Grant Number 172302009.
Appendix

A. Computational complexity of 1-SMCI method

Equation (8) can be rewritten as

\[ P_X(x_i, x_j \mid x_{-\{i,j\}}, c) = \frac{1}{\tilde{Z}_{ij}(x_{-\{i,j\}}, c)} \exp \left( -\frac{1}{2} \beta_i x_i^2 - \frac{1}{2} \beta_j x_j^2 + c_{ij} x_i x_j + J_{i,j} x_i + J_{j,i} x_j \right), \]

where

\[ \tilde{Z}_{ij}(x_{-\{i,j\}}, c) := \int_{\mathbb{X}^2} \exp \left( -\frac{1}{2} \beta_i x_i^2 - \frac{1}{2} \beta_j x_j^2 + c_{ij} x_i x_j + J_{i,j} x_i + J_{j,i} x_j \right) dx_i dx_j. \]

Here,

\[ \beta_i := \sum_{k \in V} c_{ik}, \quad J_{i,j} := \gamma_i - c_{ij} x_j, \quad \gamma_i := \sum_{k \in V} c_{ik} x_k. \quad (A-1) \]

In Eq. (A-1), we let \( c_{ii} = 0 \). For a given sampled point \( x^{(s)} \in \mathbb{X} \), we can compute all \( \beta := \{ \beta_i \mid i \in V \} \) and \( J := \{ J_{i,j} \mid i, j \in V \} \) in \( O(n^2) \). For fixed \( \beta \) and \( J \), we can compute the integration

\[ \int_{\mathbb{X}^2} (x_i - x_j)^2 P_X(x_i, x_j \mid x^{(s)}_{\setminus\{i,j\}}, c) dx_i dx_j \]

in \( O(1) \) with respect to \( n \). Thus, we can obtain \( c_{ij}^X(c, \mathbb{X}) \) for all \( i < j \in V \) in \( O(n^2) \).

B. Cohen’s kappa coefficient

Here, we briefly explain how to evaluate Cohen’s kappa coefficient \( \kappa \) [16, 19]. In our problem, the networks have three kinds of links: (i) friendly links, (ii) adversarial links, and (iii) null links. Cohen’s kappa coefficient measures the degree of agreement between two different networks, namely true and inferred networks. Suppose that the cross table is obtained as shown in Table B-I. For example, \( a_{2,1} \) is the number of links that are adversarial links in the true network and are friendly links in the inferred network. The diagonal values, \( a_{1,1}, a_{2,2}, \) and \( a_{3,3} \), represent the numbers of agreements between the two networks. The total sum of the table is \( \sum_{i,j=1}^{3} a_{i,j} = M = n(n-1)/2 \). The standard agreement rate is evaluated by

\[ P_0 := \frac{1}{M} \sum_{i=1}^{3} a_{i,i}. \]

On the other hand, we evaluate the probability of random agreement as

\[ P_1 := \frac{1}{M^2} \sum_{i=1}^{3} \left( \sum_{j=1}^{3} a_{i,j} \right) \left( \sum_{j=1}^{3} a_{j,i} \right). \]

Cohen’s kappa coefficient is obtained by

\[ \kappa := \frac{P_0 - P_1}{1 - P_1}. \quad (B-1) \]

Roughly speaking, Cohen’s kappa coefficient is a measurement that takes into account the possibility of the agreement occurring by chance.

|     | inferred |
|-----|----------|
| (i) | (ii)     | (iii) |
| (i) | \( a_{1,1} \) | \( a_{1,2} \) | \( a_{1,3} \) |
| (ii)| \( a_{2,1} \) | \( a_{2,2} \) | \( a_{2,3} \) |
| (iii)| \( a_{3,1} \) | \( a_{3,2} \) | \( a_{3,3} \) |
References

[1] K.P. Murphy, *Machine Learning: a Probabilistic Perspective*. MIT Press, 2013.
[2] I. Rish and G. Grabarnik, *Sparse Modeling: Theory, Algorithms, and Applications*. CRC Press, 2014.
[3] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
[4] A. Decelle and F. Ricci-Tersenghi, “Pseudolikelihood decimation algorithm improving the inference of the interaction network in a general class of ising models,” *Phys. Rev. Lett.*, vol. 112, p. 070603, February 2014.
[5] S. Yamanaka, M. Ohzeki, and A. Decelle, “Detection of cheating by decimation algorithm,” *Journal of the Physical Society of Japan*, vol. 84, no. 2, p. 024801, 2015.
[6] J. Besag, “Statistical analysis of non-lattice data,” *Journal of the Royal Statistical Society D (The Statistician)*, vol. 24, no. 3, pp. 179–195, 1975.
[7] M. Yasuda, “Monte carlo integration using spatial structure of markov random field,” *Journal of the Physical Society of Japan*, vol. 84, no. 3, p. 034001, 2015.
[8] M. Yasuda, “Learning algorithm of boltzmann machine based on spatial monte carlo integration method,” *Algorithms*, vol. 11, no. 4, p. 42, 2018.
[9] M. Yasuda, Y. Mikuni, Y. Yokoyama, T. Harada, A. Tanaka, and M. Yokoyama, “Detecting human-interaction networks based on statistical machine learning and sparse modeling,” *Proc. 2018 International Symposium on Nonlinear Theory and its Applications*, pp. 34–37, 2018.
[10] G.E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural Computation*, vol. 8, no. 14, pp. 1771–1800, 2002.
[11] D.H. Ackley, G.E. Hinton, and T.J. Sejnowski, “A learning algorithm for boltzmann machines,” *Cognitive Science*, vol. 9, pp. 147–169, 1985.
[12] M.J. Wainwright and M.I. Jordan, “Graphical models, exponential families, and variational inference,” *Foundations and Trends in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.
[13] A. Hyvärinen, “Consistency of pseudo likelihood estimation of fully visible boltzmann machines,” *Neural Computation*, vol. 18, no. 10, pp. 2283–2292, 2006.
[14] S. Geman and D. Geman, “Stochastic relaxation, gibbs distributions and the bayesian restoration of images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.
[15] J. Duchi and Y. Singer, “Efficient online and batch learning using forward backward splitting,” *Journal of Machine Learning Research*, vol. 10, pp. 2899–2934, 2009.
[16] J.R. Landis and G.G. Koch, “The measurement of observer agreement for categorical data,” *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.
[17] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.
[18] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, 2006.
[19] J. Cohen, “A coefficient of agreement for nominal scales,” *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.