TÜBİTAK

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 2568 – 2585
© TÜBİTAK
doi:10.55730/1300-0632.3956

MOCMIN: convex inferring of modular low-rank contact networks over COVID diffusion data

Emre SEFER*
Computer Science Department, Faculty of Engineering, Ozyegin University, Istanbul, Turkey

Received: 11.10.2021 • **Accepted/Published Online:** 27.09.2022 • **Final Version:** 28.11.2022

Abstract: SEIR (which consists of susceptible, exposed, infected, and recovered states) is a common diffusion model which could model different disease propagation dynamics across various domains such as influenza and COVID diffusion. As a motivation, across these domains, observing the node states is relatively easier than observing the network edges over which the diffusion is taking place, or it may not even be possible to observe the underlying network. This paper focuses on the problem of predicting modular low-rank human contact network edges only if a SEIR diffusion dynamics spreading among the human on their contact network can be observed. Such contact networks exhibit high modularity where the graph has dense connections between the vertices within modules, but sparse connections between vertices in different modules. We first formulate such inference problem as an optimization problem, discuss its convexity, and propose *MOCMIN* to optimally infer such unknown contacts of modular human contact network from COVID diffusion data. This modular contact network inference problem is important in the general case where human states such as infected with virus and recovered from virus can be identified more easily than the contacts between humans. Our contributions can be summarized as follows: (1) *MOCMIN* can handle noisy, incomplete, or undersampled diffusion data while inferring the unknown contact network; (2) The inferred contact networks are highly modular which cannot be ensured by the existing methods; (3) This paper applies *MOCMIN* to better understand COVID diffusion on contact network.

We found *MOCMIN* to be accurate in modular real human contact network inference from COVID diffusion data under a number of challenging scenarios. As an example, such high school contact network can be inferred by tracking COVID diffusion among humans approximately 5% better than the compared methods by *MOCMIN*'s ability to model modularity of the network. Via such inference, we can also understand the details of COVID diffusion dynamics in real human contact network. Additionally, inferred human contact graphs nearly mimic the true contact network's known graphical properties. Lastly, *MOCMIN* outperforms the competing approaches while estimating the synthetic networks.

Key words: Diffusion, COVID, SEIR, graph inference, modularity

1. Introduction

Biological and social systems' properties have been significantly modelled and analysed by networks [1, 2, 20, 26–28]. Dynamic process diffusion over the network can be used to model various phenomena of study. Diffusion can be seen as a particular case of these dynamic processes where diffusion (example, an infection) begins at some graph nodes and unrolls progressively to remaining graph parts by the graph edges over time. Some diffusion examples are idea diffusion over social networks [16], COVID, and influenza diffusion dynamics [7, 24]. A diffusion model characterizes a set of candidate states where the graph nodes can belong, as well as different rules,

*Correspondence: emre.sefer@ozyegin.edu.tr

This work is licensed under a Creative Commons Attribution 4.0 International License.

to stochastically change among these states. For instance, SEIR (Susceptible, Exposed, Infected, Recovered) [14] is a commonly-used diffusion model example which is frequently utilized to simulate the virus infection spread. SIR, SIS, and SI models [14] are SEIR model's special cases.

In most real-world scenarios, observing the node states is relatively cheaper or easier than observing the network edges over which the spreading is taking place. As an example, observing an opinion spreading over social networks can be easier but seeing the underlying network cannot be possible mainly due to privacy concerns. Another example is influenza or COVID transmission where measuring the human contact network is difficult [7, 24] but detecting whether people are sick is easier instead. Here, this article focuses on estimating the unknown network edges where we can only observe traces of node state changes while the diffusion is spreading over the network. In our setting, the unknown network models the contacts between humans in terms of COVID diffusion. Recovering transmission network is crucial to generate more effective immunization strategies and design more effective epidemic control strategies [5, 12].

This article comes up with *MOCMIN* (MOdular Convex Minimization to Infer Networks) to tackle modular low-rank graph inference problem from the diffusion traces in more realistic scenarios. Firstly, this article focuses on the scenario where diffusion traces are not perfectly observed. Such uncertainty in the diffusion traces may be explained differently in distinct scenarios. As an example, while the spread of the disease is tracked by measurements, observed symptoms such as fatigue and coughing partly describe state of a node since such symptoms do not perfectly represent the diffusion states (recovered, infected, etc.). Additionally, after an infection, all the relevant symptoms might not suddenly appear in the infected person but the symptoms gradually intensify over time [18, 21]. In this regard, exact diffusion times may not be perfectly known but instead we estimate our confidence or belief that we belong to a certain state. Secondly, diffusion trace collection and measurements are costly, so node status may not be known for all possible time steps. However, diffusion data may be observed at a lower frequency than the true diffusion dynamics operation frequency [21]. Finally, *MOCMIN* estimates modular low-rank contact networks [24] over SEIR diffusion dynamics and its special cases.

This paper addresses those challenges by handling diffusion traces that model possible states for each node as a probabilistic time series. This probabilistic modelling approach has not been employed in the existing diffusion-based graph inference approaches [9, 10, 22], where a node belongs to a state with either 1 or 0 probability. Next, this paper frames the unknown graph estimation problem as a convex expected loss minimization program over SEIR diffusion data which has $\mathrm{L}1$ regularization penalty to enforce the sparsity of the inferred graph and nuclear norm to enforce low-rank modular structure of the inferred graph. For perfect diffusion data, we may remove $L1$ norm penalty in the optimization program, and we can run *MOCMIN* nonparametrically via additional constraints which guarantee existence of at least one edge between the formerly infected graph nodes that have not recovered and recently infected nodes. At microscale, *MOCMIN* is applied to estimate synthetic graphs and high school human contact graph.

MOCMIN can infer the graphs over multiple demanding scenarios, and it frequently outperforms the competing methods in nearly all scenarios mainly owing to its probabilistic formulation capabilities. *MOCMIN*'s performance is not notably impacted by the probabilistic diffusion data while on the contrary the performance of the competing methods drop even when a sophisticated rounding scheme to preprocess the probabilistic diffusion data is applied to turn more general probabilities into $0/1$ probabilities. As an example, *MOCMIN* may attain $F1$ around 0.75–0.85 over a human contact network when we are only provided with diffusion traces over the graph as a prior information in addition to the fact that graph is modular at a certain degree.

As a summary, *MOCMIN* performs better than the other methods on both synthetic and real data under a number of challenging scenarios due to its ability to probabilistically model the observed data, and its ability to integrate low-rank modular graphs into the inference procedure. *MOCMIN* can infer graphs quite accurately when the noise dynamics parameters defining the relationship between observed diffusion data and true state transition times are unknown as well. In this scenario, *MOCMIN* may estimate the unknown graph and noise dynamics parameters at the same time, which may not be handled directly by the existing methods.

1.1. Related work

Even though there are relatively few approaches to model COVID diffusion dynamics with and without lockdown measures [11], similar methods exist in modeling the flu diffusion. A number of approaches model flu diffusion by differential equations where they assume underlying network is homogeneous and ignore the network connection patterns impact on the transmission. Nonetheless, it is invalid to make such assumption for different types of diffusion [8]. A number of methods have been developed to estimate and understand social network connections over information spreading data. Among these methods, NetRate [9] first predicts spreading probabilities and then infers graph edges over such probabilities. Both MultiTree [22] and NetInf [10] come up with maximum likelihood-based approach to infer edges that can only model edge existence.

Among more recent methods, the authors in [23] proposed a model-free nonparametric approach NPDC to infer the connectivity of social networks from diffusion data. NPDC infers the diffusion network according to the statistical difference of the infection time intervals without relying on a transmission model. However, their nonparametric approach may decrease the prediction performance in many realistic settings. LIFT [3] and TWIND [15] study the problem of diffusion network reconstruction in the case that only diffusion sources and final infection statuses of nodes are available. However, both LIFT and TWIND require a prior knowledge on the degree of influence relationships in the resulting diffusion network, as otherwise, it will iteratively add edges until a complete graph is obtained. Another method TENDS [13] does not require the degree of influence relationships. Lastly, the work in [25] focuses on inferring the transmission networks assuming the diffusion data is noisy. However, the proposed method evaluates the performance on social network inference and could not model the inference of modular networks.

There are several deficiencies of the proposed methods which we focus on addressing in this paper. Firstly, all of these methods focus mainly on social network inference and does not consider the problem on COVID diffusion as we do. Most of the existing methods expect diffusion data to include exact and precise observation of spreading over the network, so they do not consider undersampled, possibly missing probabilistic spreading data except the work in [25]. Besides, the probabilistic nature of spreading data may not be modelled by these methods. Additionally, the graph estimated by *MOCMIN* is modular which is a common feature in real-world networks.

2. Methods

2.1. Problem formulation

Let $G = (V, E)$ be an unknown underlying graph over which diffusion such as COVID propagates and which edges E cannot be easily observed. Set of edges, E, define different connections in different contexts such as human relationships in Facebook, protein interactions in PPI, and contacts between human. We assume that graph to be inferred is modular as exhibited by many real-world networks. Apart from such modular assumption, we do not integrate any other prior knowledge for graph. We also assume that vertices do not have any additional attributes. At each moment, each vertex of the graph G may belong to one of distinct *states* $\mathcal{S}$. Those states define an abstract view of vertex's condition in regards to the diffusion such as COVID diffusion.

The model $\mathcal{M}$ defines how a vertex's state may change depending on its neighbours' states at earlier moments in the diffusion. In this paper, we design inference problem over SEIR model which is more commonly used to model an epidemic. SEIR model states are S, E, I, and R. S represents a state where a vertex is susceptible to an ongoing diffusion. E defines a state where a vertex is exposed to an ongoing diffusion but it is not infected yet. I stands for a state that is infected with the diffusion and it has started spreading such infection to other vertices. Lastly, R stands for a state that was infected before but it is now recovered so stopped spreading the infection to others. SEIR model has several important special models such as SIR, SIS, and SI models over which several transitions and states do not exist. These states represent common abstractions, and they can be used in modelling multiple diffusion types across multiple contexts [24]. The SEIR, as a Markovian model, assumes an independent cascade model where one spreading from a vertex's multiple neighbours is sufficient for such vertex to move into an exposed state (or infected if exposed state does not exist).

Mathematically, a *trace* d of SEIR process sampled at T_d is composed of probabilities $\{s_v^d(t)$, $e_v^d(t)$, $i_v^d(t)$, $r_v^d(t)\}$ for each vertex $v \in V$ and each sampled time step $t \in T_d\}$, where $x_v^d(t)$ represents the probability of vertex v being in state x in trace d's time t. Among these SEIR states, each vertex v should belong only to one of such states, so $s_v^d(t) + e_v^d(t) + i_v^d(t) + r_v^d(t) = 1$ at time step t. $t_{e,v}^d$, $t_{i,v}^d$, $t_{r,v}^d$ define the true transition times into E, I, R states for vertex v in trace d, respectively. These transition times cannot be known exactly in our process, but they are modified by noise dynamics $\mathcal{N}$ to form the known trace d. We explain the noise dynamics $\mathcal{N}$ in Section 2.3.1. The degree to which we observe probabilistic states for a given trace is impacted by noise dynamics $\mathcal{N}$. In this case, we are interested in solving the following problem:

Problem 1 *We are provided with model $\mathcal{M}$ (SEIR) governing the diffusion dynamics, noise dynamics estimate $\mathcal{N}$, vertices V, and a set of probabilistic diffusion traces of vertex states D. Assuming the unknown graph is modular, we are interested in estimating the graph edges E.*

We propose the following framework to tackle Problem 1: Firstly, we come up with a list of equations which define the probabilistic dynamics of each vertex belonging to each state for SEIR model. Depending on the time steps, each vertex transit into another state and the edge existence, such equations yield a theoretical path over the space of probabilities of being in certain state. Afterwards, we design an optimization problem to infer graph edges which makes the theoretical state paths to be in line with the provided diffusion traces as close as possible, while we are optimizing the chosen loss function's expectation.

2.2. Diffusion dynamics

We define x_{uv} for every different vertex pairs $v \neq u$ such that binary $x_{uv} = 1$ if edge occurs in the graph. We assume trace d is observed for ordered discrete time points $T_d = t_1, t_2, t_3, \ldots, t_w$. Next, for every successive time points t_{j-1}, t_j in this trace, nonlinear SEIR dynamics for these discrete time points can be expressed as:

$$s_v^d(t_j) = s_v^d(t_{j-1})\, ss_v^d(t_j) \tag{1}$$

$$e_v^d(t_j) = e_v^d(t_{j-1})\left(1 - ei_v^d(t_j)\right) + s_v^d(t_{j-1})\left(1 - ss_v^d(t_j)\right) \tag{2}$$

$$i_v^d(t_j) = i_v^d(t_{j-1})\left(1 - ir_v^d(t_j)\right) + e_v^d(t_{j-1})\, ei_v^d(t_j) \tag{3}$$

$$r_v^d(t_j) = i_v^d(t_{j-1})\, ir_v^d(t_j) + r_v^d(t_{j-1}) \tag{4}$$

where I→R, E→I, and S→S state transition probabilities are represented by $ir_v^d(t_j)$, $ei_v^d(t_j)$, $ss_v^d(t_j)$ symbols, respectively. The above equations in (1)–(4) define each vertex's probability of ending up in *S*, *E*, *I*, *R* states at time t_j respectively. As an example, Eq. 3 consists of two components: 1- Vertex v is infected at t_j if it is infected at previous time step t_{j-1} and did not pass into *recovered* state, 2- Vertex v belonged to exposed states at previous time step t_{j-1} and passed into *infected* state. The only transition impacted by edges is S→E. Such transition is exogenous, and edge dependence in S→E comes from $ss_v^d(t_j)$ part. Overall, Figure 1 displays all state transitions in addition to this exogenous transition. In Figure 1, S→E state transition for vertex v is only impacted by states of vertices 1, 4 as an edge exists between v and 1, 4. A list of vertex v's state probabilities for sampled time steps are provided as part of trace d_v in the same figure.

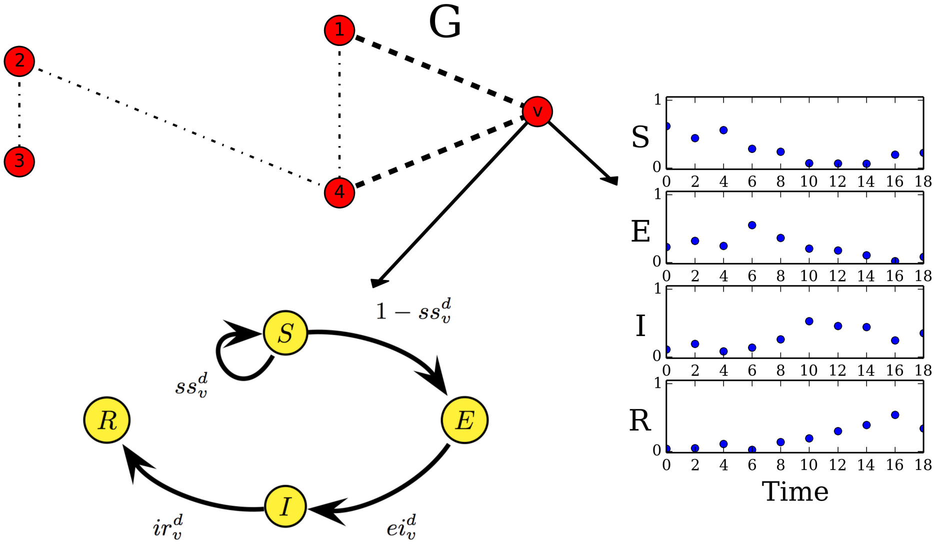


Figure 1. A list of vertex v's state probabilities for sampled time steps are provided as part of trace d_v, while S→E is the only transition impacted by graph edges.

According to SEIR model, if vertex v does not receive any infection from one of its infected neighbours until after time t, then vertex v does not transition into state E at time t. Diffusion probability from vertex u into v in trace d is represented by s_{uv}^d, and eventually the diffusion from vertex u to v occurs at time step $t_{i,u}^d + t$. Here, t is sampled from a distribution with probability mass function (pmf) p_{uv}^d and cumulative distribution function (cdf) f_{uv}^d. Given vertex u transitioned into infected state at time step t, we can compute $\mathrm{p}_{uv}^d(t', t''|t)$, the probability of vertex u exposing a neighbour vertex v over time interval $[t', t'']$ provided that u has not exposed v till t' as in Eq. (5) below:

$$\mathrm{p}_{uv}^d(t', t''|t) = \frac{P(u \text{ infected at } t \text{ exposed } v \text{ between } t' \text{ and } t'')}{P(u \text{ infected at } t \text{ has not exposed } v \text{ until } t')} = \frac{f_{uv}^d(t''-t)s_{uv}^d - f_{uv}^d(t'-t)s_{uv}^d}{1 - f_{uv}^d(t'-t)s_{uv}^d} \quad (5)$$

which is derived from Bayes rule for $t'' \geq t' \geq t$. Here, the subtraction of the terms in the nominator defines the exposure probability from vertex u in the interval $[t', t'']$, and $f_{uv}^d(\Delta t)$ is the cumulative distribution function of spreading time from vertex u to v in trace d. By integrating Eq. 5, $ss_v^d(t_j)$ which is the probability of vertex v not receiving the infection from any neighbouring vertex u, can be calculated as in:

$$ss^d_v(t_j) = \prod_{u \in V} \prod_{t < t_j} \left(1 - \mathrm{p}^d_{uv}(t_{j-1}, t_j | t)\right)^{x_{uv}\, \tilde{i}^d_u(t) \left(1 - \sum_{t' < t_j} \tilde{r}^d_u(t')\right)} \tag{6}$$

Particularly, $ss^d_v(t_j)$, vertex v's probability of staying in S state at time step t_j over diffusion d is calculated as the product of all neighbouring vertices u's probability such that 1- Vertex u transitioned into I state at time $t < t_j$ but has not recovered until t_j, 2- None of the vertices u diffused the infection to v over $[t_{j-1}, t_j]$. In Eq. (6), we have binary indicator variables $\tilde{r}^d_v(t)$, $\tilde{i}^d_v(t)$ and $\tilde{e}^d_v(t)$ for trace d which become 1 if vertex v transitions into R, I, E states at time step t respectively ($t^d_{r,v} = t$, $t^d_{i,v} = t$, $t^d_{e,v} = t$). Eqs. (7)–(8) below show how $ei^d_v(t_j)$, $ir^d_v(t_j)$ probabilities in (1)–(4) can be expressed by the probability of vertex v being E, I at time t, and E→I / I→R transition probabilities of v.

$$ei^d_v(t_j) = \sum_{t=t_1}^{t_j} p^{ei}_v(t_j - t)\, \tilde{e}^d_v(t) \tag{7}$$

$$ir^d_v(t_j) = \sum_{t=t_1}^{t_j} p^{ir}_v(t_j - t)\, \tilde{i}^d_v(t) \tag{8}$$

Table 1 summarizes and defines the diffusion model symbols.

Table 1. Symbols for diffusion dynamics.

Symbol	Definition
α^s_m, α^e_m, α^i_m, α^r_m	Dirichlet distribution parameter vector for mixture component m and states $S, \ldots, R$, respectively
$g_s(a)$, $g_e(a)$, $g_i(a)$, $g_r(a)$	The probability of sampling 4×1 state vector a rather than perfect $S, \ldots, R$ states, respectively, in any trace at any time
$\tilde{e}^d_v(t)$, $\tilde{i}^d_v(t)$, $\tilde{r}^d_v(t)$	Boolean variable which is 1 if vertex v transitions into E, I, R states in trace d at time t, respectively
$ss^d_v(t_j)$	The probability of vertex v not leaving state S between t_{j-1} and t_j
$\mathrm{p}^d_{uv}(t', t'' \| t)$	The probability of vertex v transitioning into state E during $[t', t'']$ interval by vertex u which has previously been infected at time t in trace d given u has not spreaded the infection to v until t'
$ei^d_v(t_j)$, $ir^d_v(t_j)$	The probability of $E \rightarrow I$, $I \rightarrow R$ transition for vertex v at time t_j
p^{ei}_v, p^{ir}_v	The probability distribution of $E \rightarrow I$, $I \rightarrow R$ transition time for vertex v, respectively
p^d_{uv}, f^d_{uv}	pmf, cdf of diffusion time from vertex u to vertex v in trace d
s^d_{uv}	The spreading probability from vertex u to vertex v in trace d

2.3. Expected loss minimization formulation

After defining the diffusion equations in previous sections, we now define the graph estimation Problem 1. x_{uv} are the only unknown variables in spreading Eqs. (1)–(4) as we are provided with diffusion data as well as estimation of the noise in the observed diffusion data $\mathcal{N}$ is also provided. Let $b = \{b_v, v \in V\}$ be an exact diffusion trace without any noise where set $b_v = \{t^b_{e,v}, t^b_{i,v}, t^b_{r,v}\}$ and the set components $t^b_{r,v}$, $t^b_{i,v}$, $t^b_{e,v}$ represent node v's correct recovery, infection, and exposure times in exact perfect trace b, respectively. Let $L_B : X \times B \to R$, $L_b : X \times b \to R$ be loss functions which calculate the loss of graph edges X for B and b, respectively, from diffusion equations (1)–(4) where B is list of traces without any noise. For Problem 1, list of noisy diffusion traces D is observed which describes SEIR state probabilities for each vertex at every sampled time point as introduced in Section 2.1. However, exact diffusion traces B is latent. Provided with D, we can infer the most plausible graph edges $X \subseteq V \times V$ by optimizing the expected loss function over all possible D:

$$R(X, D) = \mathbb{E}_B[L_B] = \sum_B L_B(X, B)\, P(B|D) \tag{9}$$

where $P(B|D)$ is the probability of the provided noisy data D being produced from the hidden unknown exact diffusion traces B. $P(B|D)$ incorporates $\mathcal{N}$ in its definition. In our problem, we are given multiple traces as input. $P(B|D) = \prod_{d \in D} P(b|d)$ since the noise impacts each trace d independent of other traces and every trace d is assumed to be independently collected.

Let $\mathcal{Q}(d) = \{(t_e(v), t_i(v), t_r(v)) : t_e(v) \in T_d, t_e(v) < t_i(v) < t_r(v), v \in V\}$ be the list of unknown exact diffusion data possibilities which may have generated the noisy d, we can express the complete expected loss minimization as:

$$R(X, D) = \sum_{d \in D} R(X, d) = \sum_{d \in D} \sum_{b \in \mathcal{Q}(d)} L_b(X, b)\, P(b|d) \tag{10}$$

2.3.1. Estimating $P(b|d)$

We can express $P(b|d) = \prod_{v \in V} P(b_v|d_v)$ since the noise in each observed trace impacts every node independent of each other. We can express $P(b_v|d_v)$ as below by using Bayes theorem:

$$P(b_v|d_v) = \frac{P(d_v|b_v)\, P(b_v)}{\underbrace{\sum_{b^*_v \in \mathcal{Q}(d)[v]} P\left(d_v|b^*_v\right) P(b^*_v)}_{P(d_v)}} \tag{11}$$

In the definition above, $P(d_v|b_v)$ is the probability of collecting d_v provided that the exact diffusion trace for node v is b_v. We can express $P(d_v|b_v)$ as in Eq. (12) below as collecting diffusion trace at each time point is also independent of other time points:

$$P(d_v|b_v) = \prod_{t < t^b_{e,v}} g_s(d_v[t]) \prod_{t^b_{e,v} \leq t < t^b_{i,v}} g_e(d_v[t]) \prod_{t^b_{i,v} \leq t < t^b_{r,v}} g_i(d_v[t]) \prod_{t^b_{r,v} \leq t} g_r(d_v[t]) \tag{12}$$

In Eq. ((12)), $g_x(d_v[t])$ functions for states $x \in \{s, e, i, r\}$ define the probability of collecting 4×1 vector $d_v[t]$ at t as a trace sample rather than exact S, E, I, R trace collections respectively. Since normalized $d_v[t]$ terms sum to 1, we approximate every $g_x(d_v[t])$ function by a 4-dimensional Dirichlet distributions mixture as in:

$$g_x(d_v[t]) = \sum_{m \in M} w^x_m\, g^m_x(d_v[t]) \tag{13}$$

where there are M components in the distribution and such mixture can model arbitrary functions precisely.

The concentration parameters $\alpha_m^{x,d,t}$ determine the distribution characteristics of each mixture component m for time t, trace d, and state x. We make two additional assumptions: 1- The concentration parameters for each trace d $\alpha_m^{x,d,t} = \alpha_m^x$ and time t are the same, 2- Mixture weights w_m^x for every trace d are also the same. Let $\alpha_m^x[y]$ be the concentration parameter for state y, $d_v^y[t]$ be the value of state y in $d_v[t]$, $\mathbf{B}(\alpha_\mathbf{m}^\mathbf{x})$ be the normalizing constant, we can express every component in (13) as in:

$$g_x^m(d_v[t]) = \frac{1}{\mathbf{B}(\alpha_\mathbf{m}^\mathbf{x})} \prod_{y\in\{s,e,i,r\}} (d_v^y[t])^{\alpha_m^x[y]-1} \tag{14}$$

We can express another term in (11), prior $P(b_v)$ as in:

$$\begin{aligned} P(b_v) &= P(t_{e,v}^b)\,P(t_{i,v}^b|t_{e,v}^b)\,P(t_{r,v}^b|t_{i,v}^b) \\ P(b_v) &= P(t_{e,v}^b)\,p_v^{ei}\,p_v^{ir} \end{aligned} \tag{15}$$

where $P(b_v)$'s definition includes probabilities of transitions between the states. As there is no additional knowledge on node state transition times, $P(t_{e,v}^b) = \frac{1}{|T_d|+1}$ is uniform and $P(t_{i,v}^b|t_{e,v}^b) = p_v^{ei}$, $P(t_{r,v}^b|t_{i,v}^b) = p_v^{ir}$. We also have an extra 1 in $P(t_{e,v}^b)$'s denominator to model vertex v always staying in susceptible state, without moving to exposed state.

Eq. (11) defines a generative model to explain trace noise dynamics which may be considered a slight modification of latent semi-Markov model. Our generative model can be seen as having a latent state for each time step in T_d alongside four SEIR state values R, I, E, S.

2.3.2. Estimating $L_b(X, b)$

$L_b(X, b)$ can take almost any type of loss function. However, diffusion dynamic equations are expressed in terms of probabilities so negative log-likelihood is used for loss function in our formulation. $L_b(X, b) = -\log(\mathcal{L}(X|b))$, where (16) defines such likelihood. Eq. (16) expresses the likelihood in terms of vertex state probabilities multiplication at every sampled time step in the exact trace b. The second equality in the equation is derived by putting the diffusion terms in Eq. (1)–(4) into the equation where we obtain constant C over state transitions not being affected by graph edges.

$$\mathcal{L}(X|b) = \prod_{v\in V}\left(\prod_{t<t_{e,v}^b} s_v^d(t) \prod_{t_{e,v}^b \le t < t_{i,v}^b} e_v^d(t) \prod_{t_{i,v}^b \le t < t_{r,v}^b} i_v^d(t)\right) = C \prod_{v\in V}\left(\left(1 - ss_v^d\left(t_{e,v}^b\right)\right) \prod_{t\in T_d, t<t_{e,v}^b} ss_v^d(t)\right) \tag{16}$$

When put into the loss function $R(X, D)$ in combination with (6), we can express the negative log-likelihood loss as below which is convex as proven in [25]:

$$\begin{aligned} R(X,D) = \sum_{d\in D}\sum_{b\in \mathcal{Q}(d)} P(b|d) \Bigg(& \overbrace{\sum_{v\in V} -\log\left(1 - ss^d_v\left(t^b_{e,v}\right)\right)}^{-\log(\mathcal{L}(X|b))} \\ & + \overbrace{\sum_{v\in V}\sum_{u\in V}\sum_{t^b_{i,u}\le t<\min(t^b_{r,u},t^b_{e,v})} -\log\left(1-\mathrm{p}^d_{uv}\left(t-1,t|t^b_{i,u}\right)\right)x_{uv}}\Bigg) \end{aligned} \tag{17}$$

2.3.3. An efficient relaxation

Let $\mathcal{Q}(d)$ represent the whole set of perfect transition time step possibilities, in the main optimization problem, we need to estimate the loss function's expectation over $\mathcal{Q}(d)$. The calculation of the expectation is based on exponential number of additions which can be prohibitive for even medium-size problems. We can estimate graphs more effectively by rather optimizing the relaxed expected loss ($\hat{\mathcal{R}}(X,D)$) as in:

$$\hat{\mathcal{R}}(X,D) = \sum_{d\in D}\sum_{b\in \mathcal{Q}(d)} P(b|d)\Bigg(\sum_{v\in V}\mathcal{T}^b_v + \sum_{v\in V}\sum_{u\in V}\sum_{t^b_{i,u}\le t<\min(t^b_{r,u},t^b_{e,v})} -\log\left(1-\mathrm{p}^d_{uv}(t-1,t|t^b_{i,u})\right)x_{uv}\Bigg) \tag{18}$$

where every nonlinear term $\log(1-ss^d_v(t_j))$ in the original expected loss function is replaced with first-order Taylor approximation $\mathcal{T}^b_v$ defined as in:

$$\mathcal{T}^b_v = \sum_{u\in V}\log\left(\mathrm{p}^d_{uv}(t^b_{e,v}-1,t^b_{e,v}|t^b_{i,u})\right)(x_{uv}-1) \tag{19}$$

As discussed earlier, noise dynamics for each node is independent so $P(b|d)=\prod_{v\in V}P(b_v|d_v)$. As a result, Eq. (18) could be expressed as in:

$$\hat{\mathcal{R}}(X,D) = \sum_{d\in D}\sum_{b\in \mathcal{Q}(d)}\sum_{u,v\in V\times V} P(b_u|d_u)P(b_v|d_v)\,\mathbf{M^b_{uv}}\,x_{uv} \ + \ C \tag{20}$$

where

$$\mathbf{M^b_{uv}} = \log(\mathrm{p}^d_{uv}(t^b_{e,v}-1,t^b_{e,v}|t^b_{i,u})) - \sum_{t^b_{i,u}\le t<\min(t^b_{r,u},t^b_{e,v})}\log(1-\mathrm{p}^d_{uv}(t-1,t|t^b_{e,u})) \tag{21}$$

Relaxed expected loss in Eq. (20) linearly depends on graph edges X. In Eq. (20), once expressed as $P(b|d)=\prod_{v\in V}P(b_v|d_v)$, existence of every edge x_{uv} is based only on u and v's exact transition time steps as the remaining probabilities in $P(b|d)$ have marginalized out. Every edge (u,v)'s expected loss is solely dependent upon the true recovery and infection time steps in $P_u(b|d)$, and the true exposure time in $P_v(b|d)$. As a result, Eq. (20) might be expressed in tensor form as in:

$$\hat{\mathcal{R}}(X,D) = \sum_{d\in D}\sum_{v\in V}\sum_{u\in V}\sum_{t^i_u\in T_d}\sum_{t^i_u<t^e_v}\sum_{t^e_v\le t^r_u}\left(\mathbf{P^d_{v,e}}\left[t^e_v\right]\times\mathbf{P^d_{u,i,r}}\left[t^i_u,t^r_u\right]\mathbf{M^d_{uv}}\left[t^i_u,t^r_u,t^e_v\right]x_{uv}\right) \tag{22}$$

where $(|T_d|+1)\times(|T_d|+1)$ matrix $\mathbf{P^d_{u,i,r}}\left[t^i_u, t^r_u\right]$ and $(|T_d|+1)\times 1$ vector $\mathbf{P^d_{v,e}}\left[t^e_v\right]$ are explitly defined as in:

$$\mathbf{P^d_{v,e}}\left[t^e_v\right] = \begin{cases} \sum_{t^e_v<t_1}\sum_{t_1<t_2} P_v(b=\{t^e_v, t_1, t_2\}|d) & \text{if } t\in T_d \\ 1-\sum_{t\in T_d}\mathbf{P^d_{v,e}}[t] & \text{else} \end{cases} \tag{23}$$

$$\mathbf{P^d_{u,i,r}}\left[t^i_u, t^r_u\right] = \begin{cases} \sum_{t_1<t^i_u} P_v(b=\{t_1, t^i_v, t^r_v\}|d) & \text{if } t^i_u < t^r_u \\ 0 & \text{else} \end{cases} \tag{24}$$

where the last $(|T_d|+1)$ entries are used in modelling the situation where node does not transition to the corresponding state. Given exact state transition times $t^i_u, t^r_u, t^e_v \in (T_d+1)^3$, each coefficient in $(|T_d|+1)^3$ tensor $\mathbf{M^d_{uv}}\left[t^i_u, t^r_u, t^e_v\right]$ describes the existence of the edge from u to v as in:

$$\mathbf{M^d_{uv}}\left[t^i_u, t^r_u, t^e_v\right] = \begin{cases} \log\left(\mathrm{p}^d_{uv}\left(t^e_v-1, t^e_v|t^i_u\right)\right) \text{ if } t^i_u < t^e_v \leq t^r_u \\ -\sum_{t<t^e_v}\log\left(1-\mathrm{p}^d_{uv}\left(t-1, t|t^e_u\right)\right) \\ 0 \qquad \text{else} \end{cases} \tag{25}$$

Eq. (22) can be expressed more compactly as in:

$$\hat{\mathcal{R}}(X,D) = \sum_{d\in D}\sum_{v\in V}\sum_{u\in V}\sum_{jk}\left(\mathbf{P^d_{u,i,r}}\odot\left(\mathbf{M^d_{uv}}\cdot\mathbf{P^d_{v,e}}\right)\right)x_{uv} \tag{26}$$

where every edge coefficient x_{uv} can be expressed in terms of inner and Hadamard products as in:

$$\hat{\mathcal{R}}(X,D) = \sum_{d\in D}\sum_{v\in V}\sum_{u\in V}\sum_{jk}\left(\mathbf{P^d_{u,i,r}}\odot\left(\mathbf{M^d_{uv}}\cdot\mathbf{P^d_{v,e}}\right)\right)x_{uv} \tag{27}$$

Relaxed $\hat{\mathcal{R}}(X,D)$ in Eq. (27) is a linear function of x_{uv} coefficients. We can minimize $\hat{\mathcal{R}}(X,D)$ quickly by estimating the whole set of x_{uv} coefficients via $O(|D||V|^2\max(|T_d|)^3)$ operations rather than $O(|D||V|^2\max(|T_d|)^V)$. We can find the optimal X by minimizing $\hat{\mathcal{R}}(X,D)$ with the following program:

$$\underset{X}{\mathrm{argmin}} \quad \hat{\mathcal{R}}(X,D) + \lambda_1\|X\|_* + \lambda_2\sum_{(u,v)\in V\times V} x_{uv} \tag{28}$$

$$\text{s.t.} \qquad \sum_{u\in V,\, t^i_u<t^e_v\leq t^r_u} x_{uv} \geq 1,\ \forall d\in D, v\in V \tag{29}$$

$$0\leq x_{uv}\leq 1, \quad \forall (u,v)\in V\times V \tag{30}$$

where at least one edge should exist between the previously infected but not yet recovered vertices and the newly infected vertex v for each trace sample d according to covering constraints (29). For noisy diffusion data, such covering constraints do not exist as exact transition times are unknown. X's nuclear norm is the sum of X's singular value $\sum_{i=1}^{rankX}\sigma_i$ which is represented by $\|X\|_*$. Low-rank matrices can be found efficiently via optimization of the nuclear norm [29] which corresponds to solving for graphs with high modularity in our case. Moreover, $\lambda_2\sum_{(u,v)\in V\times V} x_{uv}$ term corresponds to X's l_1 norm that enforces X's sparsity. This optimization problem is also convex since $\hat{\mathcal{R}}(X,D)$ is convex and additional linear and nuclear norm terms are also convex. Once solved the program, binary solution for edge existence can be estimated via randomly rounding x_{uv}.

2.4. Efficient inference

Despite being convex, objective function in Eq. 28 is nondifferentiable due to the nuclear norm terms enforcing the low-rank structure of X, so it is in general difficult to minimize. Alternating direction method of multipliers (ADMM) [31] idea can be used to minimize such nondifferentiable optimization programs where the optimization problem is converted into a number of subproblems which can be solved easily. ADMM can converge robustly to the solution under moderately mild situations since it belongs to family of Douglas-Rachford splitting method. Particularly, minimization in Eq. 28 can be expressed by the following form via bringing out 2 auxiliary variables X_1 and X_2:

$$\begin{aligned} \min_{X \geq 0, X_1, X_2} \quad & \hat{\mathcal{R}}(X, D) + \lambda_1 \|X_1\|_* + \lambda_2 \|X_2\|_1 \\ s.t. \qquad & X = X_1,\ X = X_2 \end{aligned} \tag{31}$$

ADMM optimizes the following expression which is the augmented Lagrangian of the problem above:

$$\begin{aligned} \hat{\mathcal{R}}_\rho = & \hat{\mathcal{R}}(X, D) + \lambda_1 \|X_1\|_* + \lambda_2 \|X_2\|_1 + \rho \operatorname{trace}(U_1^T (X - X_1)) \\ & + \rho \operatorname{trace}(U_2^T (X - X_2)) + \frac{\rho}{2}(\|X - X_1\|^2 + \|X - X_2\|^2) \end{aligned} \tag{32}$$

where $\|.\|$ defines the Frobenius norm and positive ρ is the penalty parameter. We introduce dual variables U_1 and U_2 in place of constraints $X = X_1$ and $X = X_2$, respectively. We solve the augmented Lagrangian in Eq. 32 by introducing the iterative equations below:

$$X^{k+1} = \operatorname*{argmin}_{X \geq 0} \hat{\mathcal{R}}_\rho (X^k, X_1^k, X_2^k, U_1^k, U_2^k) \tag{33}$$

$$X_1^{k+1} = \operatorname*{argmin}_{X_1} \hat{\mathcal{R}}_\rho (X^{k+1}, X_1^k, X_2^k, U_1^k, U_2^k) \tag{34}$$

$$X_2^{k+1} = \operatorname*{argmin}_{X_2} \hat{\mathcal{R}}_\rho (X^{k+1}, X_1^k, X_2^k, U_1^k, U_2^k) \tag{35}$$

$$U_1^{k+1} = U_1^k + (X^{k+1} - X_1^{k+1}) \tag{36}$$

$$U_2^{k+1} = U_2^k + (X^{k+1} - X_2^{k+1}) \tag{37}$$

By this sequential scheme, we can solve for multiple variables iteratively and independently. Next, we define the minimization problem for X_1 and X_2, and afterwards define the method to minimize X.

2.4.1. Solution for X_1 and X_2:

When solving for X_1 in Eq. 34, the relevant $\hat{\mathcal{L}}_\rho$ terms are $\lambda_1 \|X_1\|_* + \rho \operatorname{trace}((U_1^k)^T (X^{k+1} - X_1)) + \frac{\rho}{2}(\left\|X^{k+1} - X_1\right\|^2$ which becomes:

$$X_1^{k+1} = \operatorname*{argmin}_{X_1} \lambda_1 \|X_1\|_* + \frac{\rho}{2} \left\|X^{k+1} - X_1 + U_1^k\right\|^2 \tag{38}$$

This problem has a closed form solution:

$$X_1^{k+1} = S_{\frac{\lambda_1}{\rho}} (X^{k+1} + U_1^k) \tag{39}$$

where $S_\alpha(Q)$ is a soft-thresholding function defined as $S_\alpha(Q) = U\text{diag}((\sigma_i - \alpha)_+)V^T$ with SVD $Q = U\text{diag}(\sigma_i)V^T$. Similar to X_1, we can express the minimization of X_2 as a closed-form solution:

$$X_2^{k+1} = \underset{X_2}{\text{argmin}}\ \lambda_2 \|X_2\|_1 + \frac{p}{2}\left\|X^{k+1} - X_2 + U_2^k\right\|^2 \tag{40}$$

Given $X^{k+1} + U_2^k$, $(X_2^{k+1})_{ij}$ entry's update can be expressed as below where sgn is sign function:

$$(X_2^{k+1})_{ij} = \begin{cases} 0 & |(X^{k+1} + U_2^k)_{ij}| < \frac{\lambda_2}{\rho} \\ D^{k+1} & \text{else} \end{cases} \tag{41}$$

where $D^{k+1} = (X^{k+1} + U_2^k)_{ij} - sgn\left((X^{k+1} + U_2^k)_{ij}\right)\frac{\lambda_2}{\rho}$.

2.4.2. Solution for X

Eq. 32 defines the minimization of X which may also be expressed as in:

$$X^{k+1} = \underset{X \geq 0}{\text{argmin}}\ \hat{\mathcal{R}}(X, D) + \frac{p}{2}\left(\left\|X - X_1^k + U_1^k\right\|^2 + \left\|X - X_2^k + U_2^k\right\|^2\right) \tag{42}$$

This problem can be efficiently solved via gradient descent by using backtracking line search for optimal step size selection which is based on satisfying the Armijo-Goldstein condition. Overall, our method is a combination of alternating direction method of multipliers [31] and gradient descent.

3. Results

3.1. Diffusion trace generation

While generating a possible synthetic trace for COVID diffusion, we choose an initial vertex at random, and simulate the diffusion dynamics till the spread terminates without reaching out all the graph vertices, or all vertices move to a terminal model state (recovered state for SIR and SEIR model, or infected state for SI). If infection spreads to a vertex via different vertices through multiple time points, such vertex transitions into the infected state at the earliest receival of the infection. We model COVID transmission by SEIR, SIR, and SI models, and use realistic COVID diffusion parameters to simulate COVID diffusion as described in [18]. We model p_{uv} by weibull distribution with $k = 2.3, \lambda = 9.5$, s_{uv} as 0.2, p_v^{ir} as exponential distribution with $\lambda = 0.2$, and p_v^{ei} as exponential distribution with $\lambda = 0.5$.

Let p be the normalized noise rate between 0 and 1, synthetic noise is added to the diffusion traces as follows: Probabilistic vector of size 4 is sampled over Dirichlet distribution with concentration parameters $\alpha = \left[\frac{p}{4}, \frac{p}{4}, \frac{p}{4}, 1 - \frac{3p}{4}\right]$, and this sampled state vector is added to each time point and vertex where $1 - \frac{3p}{4}$ is the current state's concentration parameter. When the noise level is high, such concentration parameter vector turns into a more uniform vector, and recovering the true vertex state from noisy diffusion trace is not possible.

3.2. Real and synthetic networks

We evaluated the performance of *MOCMIN* in estimating the unknown human contact network at an American high school [24], called *Contact-static*, over diffusing COVID infection data. The vertices of *Contact-static* graph represent tracked people and an edge represents a COVID transmission possibility existing when two people are reasonably close to each other.

In terms of synthetic data, we have created 10 graphs over each of Erdos-Renyi (RDS), ForestFire (FF) [17], LPA [4], DMC [30] models. For all these models, each graph includes 500 nodes and 5000 edges, and we have sampled the networks at uniform over their parameter space.

3.3. Experiment details

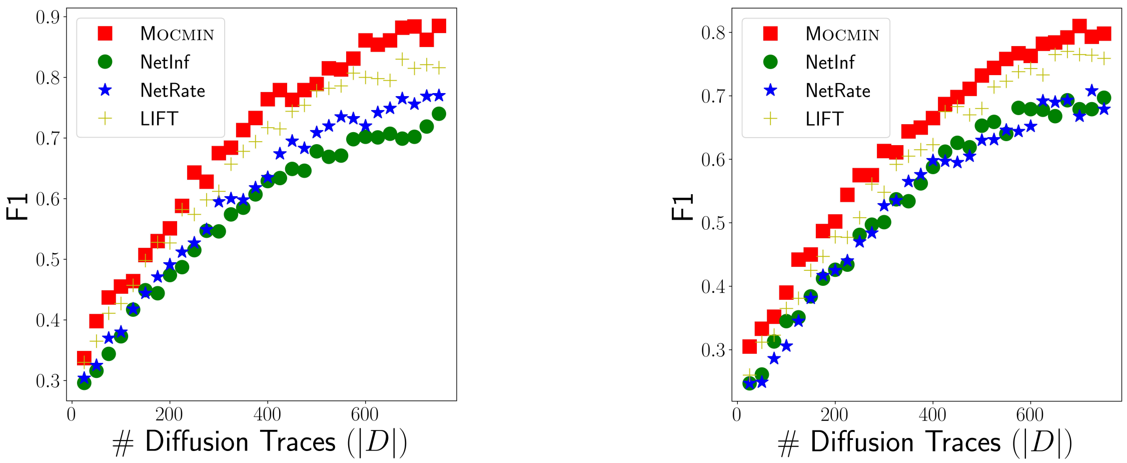
MOCMIN is implemented mainly in Python. Datasets and code are publicly available at https://github.com/seferlab/mocmin. *MOCMIN* is sensibly fast. For instance, graph of 500 nodes and 5000 edges can be estimated fairly accurately in less than 15 min over 100 diffusion traces on a personal laptop. The performance of *MOCMIN* is evaluated with respect to the current similar approaches NetInf [10], NetRate [9], and MultiTree [22], LIFT [3]. In our experiments, we do not report MultiTree as it performs worse than NetInf and NetRate in almost all experiments. We provide the true edge count as a parameter to NetInf and MultiTree in our experiments even though such ideal knowledge does not exist a priori. When we are provided with the fully observed diffusion data without any noise, *MOCMIN* uses the covering constraints and it is executed nonparametrically. However, when the diffusion data is noisy, the sparsity parameter λ_2 in (28) is estimated via 5-fold cross-validation. Such cross-validation is performed by using the diffusion traces as follows: Firstly, unknown graph edges are estimated over diffusion data's training portion for 500 λ_2 parameters uniformly sampled between 0 and 100. Next, the error of observing the previously unused traces over the estimated graph is calculated for each λ_2. We repeat this process for 5 parts of the cross-validation, and select the parameter that minimizes the overall error as our λ_2.

While evaluating the estimation performance over the contact network, the vertex pairs without an edge between them are the negative examples whereas graphs' edges are the positive ones. We evaluate the performance mainly by $F1 = \frac{2\,\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$ where recall is portion of edges in contact graph which exist in the estimated graph, and precision is portion of edges in the estimated graph which exist in the contact graph.

3.4. Inferring human contact graph from COVID diffusion data

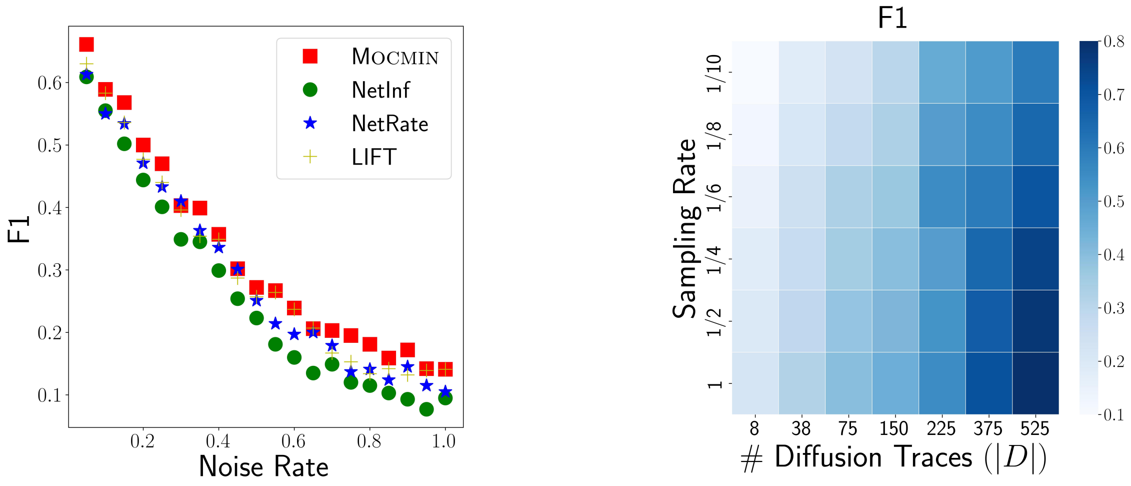
We estimated *Contact-static* via COVID diffusion traces on SEIR, SIR, and SI models which have been produced by using the realistic COVID transmission parameters as described in Section 3.1. According to those COVID diffusion data in *Contact-static*, exposed state describes the people who have received the COVID infection from their neighbours but has not begun transmitting it to the other connected people in the school graph, whereas the infected state describes the people who have received the COVID infection from their neighbours and currently transmitting the infection to the remaining connected people. *MOCMIN* outperforms all the competing approaches for perfectly observed diffusion traces under SI model despite performing a nonparametric inference as in Figure 2. Furthermore, *MOCMIN* also outperforms the competing approaches under SIR model as in Figure 2 where the performance enhancement of *MOCMIN* over the competing approaches is bigger than the relative enhancement for SI model in Figure 2.

MOCMIN starts to outperform the competing approaches more significantly for partially-observable noisy diffusion traces where cross-validation is used to calculate its low-rank and sparsity parameters λ_1, λ_2 . The scenario of noisy diffusion traces is quite realistic for COVID transmission dynamics since exact tracking of the COVID dynamics can be quite expensive for many reasons: COVID symptoms can be quite similar to the symptoms of other diseases so they may be confusing, and a number of collected diffusion traces can be few particulary for novel COVID variants such as delta, gamma when these variants started to appear. *MOCMIN*'s $F1$ score of 0.7 can be obtained from 350 perfectly known traces, and the same score can be achieved by almost 700 noisy diffusion traces as seen in Figure 2. In comparison, 0.5 $F1$ score may be achieved via the competing approaches by using the same set of partially observable diffusion traces. *MOCMIN* is also robust to varying levels of noise in the diffusion data. *MOCMIN* can still attain $F1$ score around 0.4 when the diffusion traces are remarkably noisy as in Figure 3. However, the competing approaches' $F1$ scores become less than 0.2 as these approaches are remarkably impacted by the rising noise levels in the same figure.



(a) SIperfect (b) SIRperfect

Figure 2. The performance of *Contact-static* prediction in terms of $F1$ vs. diffusion trace count under (a) SI and (b) SIR models by using perfectly known diffusion data.

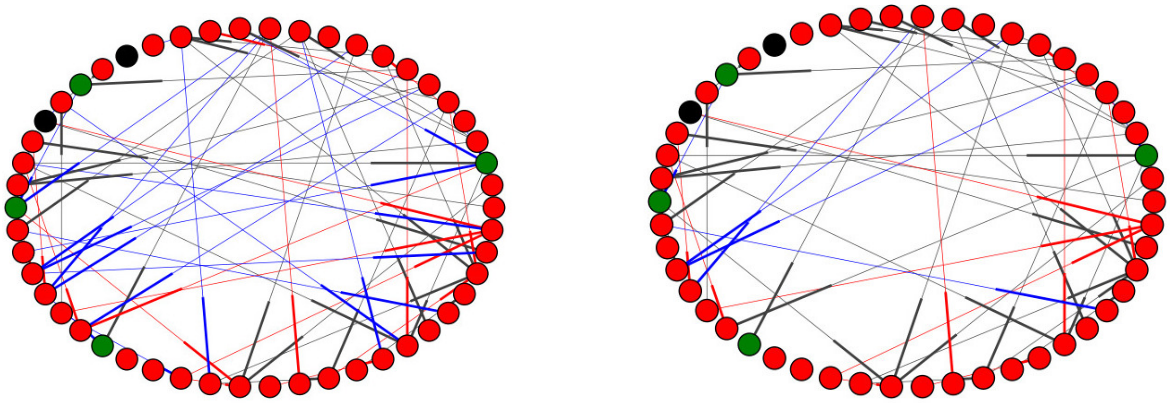


(a) SI partial (b) SEIR Heatmap

Figure 3. a) The performance of *Contact-static* prediction in terms of $F1$ vs. noise rate under SI model from 250 traces, b) Heatmap of performance by $F1$ in terms of number of traces vs. sampling rate under SEIR.

MOCMIN's overall performance drops by decreasing the rate at which diffusion traces are sampled as in Figure 3, since such sampling at a low rate extracts less knowledge. In this figure, $\frac{1}{x}$ rate shows that a single time step is observed in each interval of x steps. According to the figure, collecting the diffusion data at a low sample rate in general decreases the performance of *MOCMIN*, but its performance is acceptable when diffusion traces are sampled at rate greater than $\frac{1}{5}$ for SEIR models across varying numbers of diffusion traces. As a summary, *MOCMIN* tends to perform quite accurately over both partially known and exactly known diffusion data, and *MOCMIN*'s performance can tolerate reasonable noise levels in the diffusion traces which is not true for the competing approaches.

MOCMIN infers the connections among people quite accurately as in Figure 4 which illustrates a randomly sampled 50 vertex subgraph of both the true and inferred contact networks over 50 and 200 diffusion traces respectively. In this figure, red edges show the edges that are in the human contact graph but not in the inferred graph, blue edges show the edges which are in the inferred graph but not in the true graph, and gray edges define the edges which are estimated correctly by *MOCMIN*. When vertices are considered in the figure, students are represented by red vertices, the school staff and the teachers are represented by black and green vertices respectively. Figure 4 shows that subgraph induced only by the students in the estimated graph is dense; in addition, misestimated edges are mostly between the students.


(a) 50 traces

(b) 200 traces

Figure 4. *MOCMIN* inferred and true 50 node *Contact-static* subgraphs under SI model from a) 50 diffusion traces, b) 200 diffusion traces.

Properties of the graphs estimated by *MOCMIN* resemble the underlying contact graph's range of properties even when the available trace count is limited as in Table 2. According to the table, we compare properties of the true and estimated *Contact-static* graphs only from 50 diffusion traces. As an example, the true contact graph's node degree distribution is scale-free with exponent 2.134, whereas the inferred graph also exhibits scale-free distribution with a close exponent 2.072. Similarly, the modularity of the true contact network is reasonably high 0.73, whereas the inferred graph has almost the same modularity of 0.74. The similarity of the network attributes can also be seen across many other metrics such as assortativity, average k-core, average shortest path length.

3.5. Estimating synthetic networks

MOCMIN regularly outperforms the competing approaches while estimating the synthetic networks generated by a number of dissimilar models over multiple diffusion models as in Table 3. The bold entries in the table represent reasonable outperformance of *MOCMIN* with respect to the competing approaches. *MOCMIN* outperforms the competing approaches while estimating the networks generated by LPA and FF models. Additionally, the performance of all approaches including *MOCMIN* are similar while estimating RDS graphs, and the performance of all these methods become worse on estimating DMC graphs which may be due to the DMC graphs cyclic structure. In terms of estimating the synthetic graphs, $F1$ score captured by *MOCMIN* in all growth models other than DMC is more than 0.5 only when 250 diffusion traces are used. Even though we report the performance over 250 diffusion traces in Table 3, *MOCMIN* performs consistently accurately across a number of conditions.

Table 2. The properties of inferred and true contact graphs over 50 traces.

	Truth	Inferred
Modularity [19]	0.73	0.74
Diameter	8	10
Average clustering coefficient	0.261	0.23
Assortativity	0.121	0.141
Scale-free exponent [6]	2.134	2.072
Average k-core	4	4
Average shortest path length	3.124	3.054

Table 3. Synthetic network inference performance in terms of $F1$ according to diffusion and graph growth models while using 250 traces without any noise.

	RDS			DMC			LPA			FF		
	SEIR	SIR	SI	SEIR	SIR	SI	SEIR	SIR	SI	SEIR	SIR	SI
MOCMIN	**0.55**	**0.53**	0.52	0.49	0.44	0.45	**0.61**	**0.5**	**0.59**	**0.61**	**0.57**	**0.62**
NetRate	0.36	0.28	0.45	0.47	0.39	0.41	0.44	0.42	0.52	0.43	0.5	0.45
NetInf	0.38	0.33	0.47	0.47	0.41	0.4	0.47	0.42	0.50	0.46	0.45	0.52
LIFT	0.4	0.35	0.49	0.45	0.34	0.44	0.45	0.43	0.51	0.46	0.47	0.54

4. Conclusion

Here, we come up with a convex expected loss minimization-based method *MOCMIN* to estimate modular unknown networks under SEIR models from possibly noisy diffusion data. We find enhanced network recoverability under both noisy and perfect diffusion data; *MOCMIN* can recover the COVID transmission network quite accurately. *MOCMIN*'s better performance can be attributed to its capability to model both edge nonexistence and existence from diffusion data, its capability to handle noisy data more precisely, and mainly its formulation that infers modular graphs from diffusion data. We believe that the COVID transmission networks inferred via *MOCMIN* will be particulary useful in understanding COVID diffusion dynamics, and taking better preventative measures. Overall, our contributions can be summarized as follows: (1) *MOCMIN* can handle noisy, incomplete, or undersampled diffusion data while inferring the unknown contact network; (2) The inferred contact networks are highly modular which cannot be ensured by the existing methods; (3) This paper applies *MOCMIN* to better understand COVID diffusion on contact network. One drawback of this study is that the application of graph inference uses semisynthetic diffusion data instead of real diffusion data, which is due to such real data being limited and nonpublic. As future work, diffusion data collected by mobile phone tracking can be used to further elaborate on the results. We can also forward the findings of this study for controlling the massive spread of COVID and other related diseases. Another limitation of this study is to assume an underlying diffusion model such as SI to derive the mathematical formulation. In this case, another future work might be necessary to predict COVID diffusion dynamics by a model-free approach.

References

[1] Alzubi J, Kumar A, Alzubi O, Manikandan R. Efficient approaches for prediction of brain tumor using machine learning techniques. Indian Journal of Public Health Research & Development 2019; 10:267.

[2] Alzubi OA, Alzubi JA, Alweshah M, Qiqieh I, Al-Shami S et al. An optimal pruning algorithm of classifier ensembles: dynamic programming approach. Neural Computing and Applications 2020; 32 (20):16091-16107.

[3] Amin, K, Heidari, H, Kearns, M. Learning from contagion (without timestamps). In Proceedings of the 31st International Conference on Machine Learning, volume 32 of Proceedings of Machine Learning Research 2014; pages 1845-1853.

[4] Barabási AL, Albert R. Emergence of scaling in random networks. Science 1999; 286 (5439):509-512.

[5] Carcione JM, Santos JE, Bagaini C, Ba J. A simulation of a covid-19 epidemic based on a deterministic seir model. Frontiers in Public Health 2020; 8:230.

[6] Clauset A, Shalizi CR, Newman MEJ. Power-law distributions in empirical data. SIAM Review 2009; 51 (4):661-703.

[7] Coccia M. Factors determining the diffusion of covid-19 and suggested strategy to prevent future accelerated viral infectivity similar to covid. Science of The Total Environment 2020; 729:138474.

[8] Giordano G, Blanchini F, Bruno R, Colaneri P, Di Filippo A et al. Modelling the covid-19 epidemic and implementation of population-wide interventions in italy. Nature Medicine 2020; 26 (6):855-860.

[9] Gomez-Rodriguez M, Balduzzi D, Schölkopf B. Uncovering the temporal dynamics of diffusion networks. In Proceedings of the 28th International Conference on Machine Learning 2011; 561-568.

[10] Gomez-Rodriguez M, Leskovec J, Krause A. Inferring networks of diffusion and influence. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2010; 1019-1028.

[11] Gribaudo M, Iacono M, Manini D. Covid-19 spatial diffusion: A markovian agent-based model. Mathematics 2021; 9 (5).

[12] Gros C, Valenti R, Schneider L, Valenti K, Gros D. Containment efficiency and control strategies for the corona pandemic costs. Scientific Reports 2021; 11 (1):6848.

[13] Han K, Tian Y, Zhang Y, Han L, Huang H et al. Statistical estimation of diffusion network topologies. In 2020 IEEE 36th International Conference on Data Engineering (ICDE); 625-636.

[14] Hethcote HW. The mathematics of infectious diseases. SIAM Review 2000; 42 (4):599-653.

[15] Huang H, Yan Q, Gan T, Niu D, Lu W et al. Learning diffusions without timestamps. Proceedings of the AAAI Conference on Artificial Intelligence 2019; 33 (01):582-589.

[16] Kwak H, Lee C, Park H, Moon S. What is twitter, a social network or a news media? In Proceedings of the 19th International Conference on World Wide Web 2010; 591-600.

[17] Leskovec J, Kleinberg J, Faloutsos C. Graphs over time: Densification laws, shrinking diameters and possible explanations. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining 2005; pages 177-187.

[18] Menni C, Valdes AM, Freidin MB, Sudre CH, Nguyen LH et al. Real-time tracking of self-reported symptoms to predict potential covid-19. Nature Medicine 2020; 26 (7):1037-1040.

[19] Newman MEJ. Modularity and community structure in networks. Proceedings of the National Academy of Sciences 2006; 103 (23):8577-8582.

[20] Patro R, Duggal G, Sefer E, Wang H, Filippova D et al. The Missing Models: a Data-driven Approach for Learning How Networks Grow. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD) 2012; 42-50.

[21] Planas D, Veyer D, Baidaliuk A, Staropoli I, Guivel-Benhassine F et al. Reduced sensitivity of sars-cov-2 variant delta to antibody neutralization. Nature 2021; 596 (7871):276-280.

[22] Rodriguez MG, Schölkopf B. Submodular inference of diffusion networks from multiple trees. In Proceedings of the 29th International Conference on Machine Learning (ICML-12) 2012; 489-496.

[23] Rong Y, Zhu Q, Cheng H. A model-free approach to infer the diffusion network from event cascade. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16 2016; 1653-1662.

[24] Salathè M, Kazandjieva M, Lee JW, Levis P, Feldman MW et al. A high-resolution human contact network for infectious disease transmission. Proceedings of the National Academy of Sciences 2010; 107 (51):22020-22025.

[25] Sefer E, Kingsford C. Convex risk minimization to infer networks from probabilistic diffusion data at multiple scales. In 2015 IEEE 31st International Conference on Data Engineering 2015; 663-674.

[26] Sefer E, Kingsford C. Diffusion archeology for diffusion progression history reconstruction. Knowledge and Information Systems 2016; 49:403–427.

[27] Sefer E, Kingsford C. Semi-nonparametric modeling of topological domain formation from epigenetic data. Algorithms for Molecular Biology 2019; 14 (1):4.

[28] Sefer E, Kingsford C. Metric Labeling and Semimetric Embedding for Protein Annotation Prediction. Journal of Computational Biology 2021; 28:514-525.

[29] Srebro N. Learning with matrix factorizations. PhD thesis 2004; Citeseer.

[30] Vázquez A, Flammini A, Maritan A, Vespignani A. Modeling of Protein Interaction Networks. Complexus 2003; 1 (1):38-44.

[31] Zhou K, Zha H, Song L. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics 2013; 641-649.