

Article

Not peer-reviewed version

Indirect Model Predictive Control on Epidemiological Stochastic Blockmodels

[Jonas Hjulstad](#)* and Morten Hovd

Posted Date: 6 May 2024

doi: 10.20944/preprints202405.0241.v1

Keywords: identification for control; optimization and control of large-scale network systems; Monte Carlo methods; high performance computing



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Indirect Model Predictive Control on Epidemiological Stochastic Blockmodels

Jonas Hjulstad * and Morten Hovd

Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway;
morten.hovd@ntnu.no

* Correspondence: bagoien.j.hjulstad@ntnu.no

Abstract: This paper aims to demonstrate how the detection of communities for epidemiological networks can be used to reduce the dimensionality of optimal control problems. A range of planted partition stochastic blockmodels are generated, where the underlying communities in the model vary in detectability. A high-performance computing workflow is then used to simulate and estimate reduced model dynamics for the networks, which in turn is used to find optimal control solutions. The solutions are validated with new simulations on the networks, which demonstrates that the workflow is able to detect communities that reduce the dimensionality of the control problem, with robust optimal control capabilities.

Keywords: identification for control; optimization and control of large-scale network systems; Monte Carlo methods; high performance computing

1. Introduction

1.1. Background

Ranking as one of the deadliest epidemics in history [1], the COVID-19 pandemic has shown the need for accuracy in the modeling and response to epidemics. In order to mitigate the spread of such diseases, it is important to develop models that are able to utilize all data available that can be used to determine the epidemics course.

Epidemics on large populations are usually modeled with compartmental models like the Kermack-McKendrick (SIR) model [2] which models disease infection and recovery. These models yield accurate predictions for homogenous populations.

In contrast to these frequently used compartmental models, the true dynamics of an epidemic is heterogenous and exhibits the small-world and scale-free properties exhibited by real-world networks [3]. Network dataset analysis on mobile communication records and mobility fluxes [4] shows strong correlations between mobility and communication patterns in populations, which encourages usage of heterogenous epidemiological network models. These models also enable data association on more detailed levels, which enables the implications of targeted control inputs (such as disease spread mitigation strategies, self-quarantine recommendations and local restrictions on facilities and events) to be compared to implications of more global control strategies (such as regional lockdowns and restrictions on large population gatherings).

Model predictive control on compartmental models is generally computationally feasible, both for nonlinear programs (NLPs) and mixed-integer ones. In contrast, due to the number of distinct control inputs that can be present in an epidemiological network model, direct optimization on the network model is not computationally feasible. A remedy for this issue is to approximate parts of these networks with simpler models, and instead solve control problems for these approximations. These approximations need to account both for the underlying graph structure and the dynamics of the network.

Extensive work has previously been done on the approximation of graph structures, and it is frequently used for community detection and cluster identification in networks. Bayesian inference methods are usually applied to networks, in order to categorize nodes into distinct partitions. Stochastic

blockmodels (SBMs) are generally used to retrieve partitions that capture clusters and communities, which can be used to form hierarchical layers of graphs [5]. The majority of theory related to SBM-inference in this paper is summarized in [6]. This includes the detectability-indetectability phase transition, which is used to demonstrate how control input dimensionalities can be reduced when the number of detected communities are reduced.

One potential solution to avoid the infeasibility of graph optimization is to treat network models as black-box models, which only considers input signals and output measurements of the network. Black-box models lack interpretability and transparency, which are important properties for control strategies that potentially have fatal consequences. A remedy for the interpretability in black-box models is sparse nonlinear regression methods, which have gained traction over the recent years [7]. Sparse nonlinear regression enable the identification of models based on input signals and output measurements, such that the resulting models consists of as few parameters as possible.

Monte Carlo (MC) methods are used with Bayesian inference in order to approximate posterior distributions for the quantities of Susceptible (S), Infectious (I) and Recovered (R) individuals in the networks populations. Importance sampling [8] is frequently used to reduce the dimensionality of the sample space, which in this paper is used on the stochastic dynamics found in the edges of dynamical networks.

1.2. Overview

This paper presents a high performance computing (HPC) workflow for simulation, model reduction and optimal control of epidemics, from the level of an individual-based network model. The workflow is demonstrated by performing MC-simulations, community detection, regression and optimal control over networks whose population clusters become increasingly more interconnected (Figure 1). The relation between them will be explained by first introducing the community and individual level dynamics, followed by a bayesian inference approach to identify communities for weighted graphs.

MC-simulations are run on the identified communities, which generates data used to fit Least Squares (LS) and Quantile Regression (QR) models. These models are then used to approximate Robust Model Predictive Control (RMPC) problems aimed at minimizing the total number infected of an epidemic with minimal intervention. Finally, the control input solutions are validated with new MC-simulations on the networks.

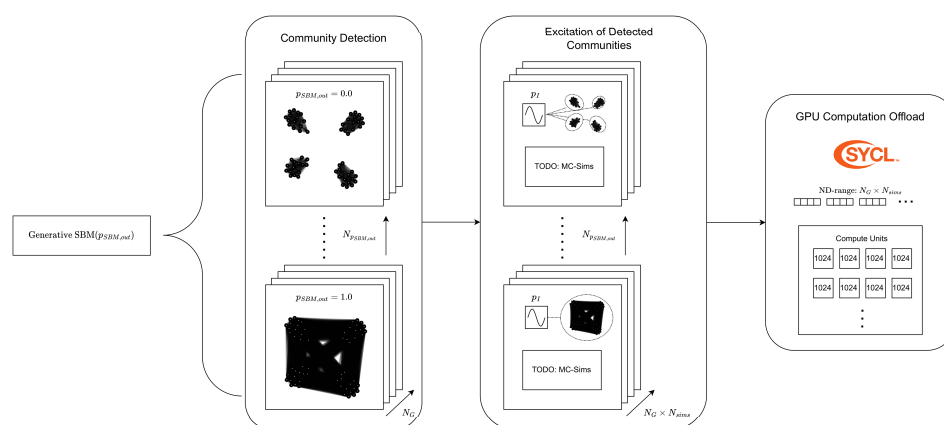


Figure 1. Workflow for generating epidemiological networks and offloading MC-simulations.

2. Epidemiological Models

This paper utilize three levels of graph-structured epidemiological models, which are all described by separate adjacency matrices (individual-level A_{SBM} , partition-level A_{part} and community-level A_{com}). The next sections formulate dynamics for the individual and community levels.

2.1. Community SIR Model

A community SIR model can be defined for a population in order to account for infections caused by external populations. Compartmental populations N_S , N_I and N_R denote the susceptible, infected and recovered groups of each community.

Individuals are divided into communities $c = \{c_1, \dots, c_{N_c}\}$, where n and m denotes source and target communities. Infections inside ($n = m$) and between communities occur over discrete timesteps t with bilinear infection dynamics.

$$\Delta N_{I,t}^{(rs)} = \theta_{rs} \frac{N_{S,t}^{(r)} N_{I,t}^{(s)}}{N_{S,t}^{(s)} + N_{I,t}^{(r)} + N_{R,t}^{(s)}} \quad (1)$$

$\theta = [\theta_{11}, \theta_{12}, \dots, \theta_{N_c N_c}]$ contains parameters the average number of contacts between communities for each person over a timestep. A known recovery rate α applies to all infected individuals, which in combination with (1) can be used to derive discrete dynamics (2-4) for the health states of all communities.

$$\Delta N_{S,t}^{(r)} = - \sum_{s \in c} \Delta N_{I,t}^{(rs)} \quad (2)$$

$$\Delta N_{I,t}^{(r)} = \sum_{s \in c} \Delta N_{I,t}^{(rs)} - \alpha N_{I,t}^{(r)} \quad (3)$$

$$\Delta N_{R,t}^{(r)} = \alpha N_{I,t}^{(r)} \quad (4)$$

2.2. Stochastic Blockmodel

Let $\mathbf{b}$ denote vertex partitions in the individual-level graph with adjacency matrix A_{SBM} . The stochastic blockmodel is a generative graph model parameterized by a probabilities $\mathbf{p}_{SBM}$, where $\mathbf{p}_{SBM,rs}$ denotes the probability of generating an edge between vertices contained in partitions r and s . From a Bayesian perspective, $\mathbf{p}_{SBM}$ parameterize the maximum entropy probability distribution for generating an adjacency matrix A_{SBM} over partitions $\mathbf{b}$ [6]. Graphs generated in this paper utilize the *planted partition* SBM, which is parameterized by internal probability $p_{SBM,in}$ (for $r = s$), and outgoing probability $p_{SBM,out}$. It should be emphasized that this papers graphs are static, in contrast to dynamic contact networks.

2.3. Stochastic SIR Network Model

Each vertex $v_i \in \{0, \dots, N_V - 1\}$ is assigned a state $V_i(t) \in \{S = 0, I = 1, R = 2\}$. Initial infections are assigned with Bernoulli probability $p_{I,0}$, with remaining individuals initially assigned as susceptible. Edges between vertices of partitions r and s are assigned uniform infection probability $p_I^{(rs)}$. $q_{r,t}^{(i)}$ is used to denote the number infectious contact events inflicted upon an individual v_i by partition r .

$$q_{r,t}^{(i)}(\omega) = \sum_{j \in \mathcal{N}_{I,t}^{(i,r)}} q_{I,t}^{(ij)}(\omega) \sim \text{Bin}(N_{I,t}^{(r)}, p_I^{(rb(i))}) \quad (5)$$

For each sample ω from the full space of possible outcomes Ω , $q_{I,t}^{(ij)}(\omega) \sim \text{Be}(p_I^{(rs)})$ are contact events occurring between communities at time t , and $q_{R,t}^{(i)}(\omega) \sim \text{Be}(p_R)$ are recovery events. $\mathcal{N}_{I,t}^{(i,r)}$ denotes the infected neighborhood of v_i located in r . Since infections between partitions are uniformly Bernoulli-distributed, the resulting number of community contact events inflicted on an individual are binomially distributed.

Under Markov assumptions, the dynamics for an individual $V^{(i)}$ can be formulated.

$$V_{t+1}^{(i)} = \begin{cases} \mathbb{1}(\sum_{r \in \mathbf{b}} q_{r,t}^{(i)}(\omega)), & V_t^{(i)} = S \\ 1 + q_{R,t}^{(i)}(\omega), & V_t^{(i)} = I \end{cases} \quad (6)$$

The purpose of the detailed notation is to obtain infection likelihoods between all communities. The likelihood for an individual to be infected by a community is determined by the infection event contributions from all communities.

$$\mathcal{L}_r(V_{t+1}^{(i)} = I) = \frac{q_{r,t}^{(i)}(\omega)}{\sum_{s \in \mathcal{C}} q_{s,t}^{(i)}(\omega)} \quad (7)$$

Since an individual can be exposed to multiple contact events during a timestep, the likelihood will be used to sample the events that actually cause the infections.

3. Weighted Graph Bayesian Inference

Structural inference will be performed on the partition-level (A_{part}), where partitions b labels individuals that are subject to common control inputs (Figure 2a). Probability for generating an adjacency matrix A over communities c is given as [6]:

$$P(A|\lambda, c) = \prod_{n < m} \frac{e^{-\lambda_{cn,cm}} \lambda_{cn,cm}^{A_{nm}}}{A_{nm}!} \times \prod_i \frac{e^{-\lambda_{cn,cm}/2} (\lambda_{cn,cm}/2)^{A_{nn}/2}}{(A_{nn}/2)!} \quad (8)$$

Where λ_{nm} is the average number of edges between pairs of nodes in communities n, m .

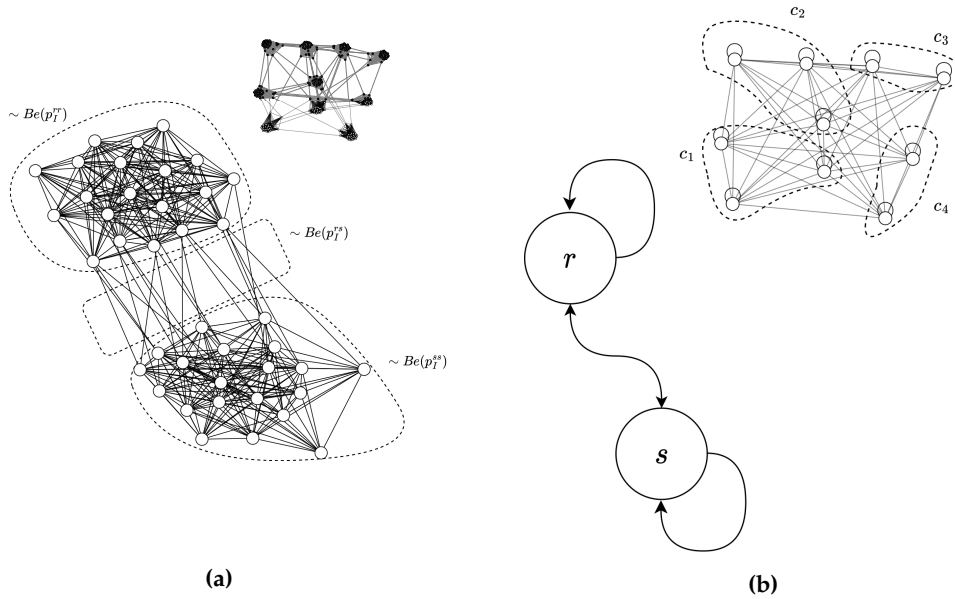


Figure 2. (a) Individual-level planted partition SIR network model, where infection probabilities are uniform within and between partitions r and s . Edges were generated with $p_{SBM,in} = 0.8$ and $p_{SBM,out} = 0.05$. (b) Partition-level weighted graph (A_{part}) used for community detection. A_{com} connects the inferred communities c , shown with dotted lines.

Decoupling the inference of dynamics and structure can be challenging for arbitrary dynamical networks. Due to uniform assignments of p_I , this paper assumes that all edges are equally important for the structural inference problem, which makes it possible to decouple the inference of communities from the inference of parameters for the community ODE. With prior distribution $P(c)$ the posterior distribution for c becomes:

$$P(c|A) = \frac{\int P(A|\lambda, c) P(\lambda|c) d\lambda P(c)}{P(A)} \quad (9)$$

4. Monte-Carlo Importance Sampling

In order to relate the stochastic network model (6) to the deterministic one (2-4) Monte-Carlo (MC) importance sampling will be used to approximate marginal likelihoods for the communities. Let $p_{I,t}^{(rs)} \sim U(p_{I,min}, p_{I,max})$, ξ denote the vector of all random variables $\mathbf{U}_{I,0:N_t}$, $\mathbf{U}_{R,0:N_t}$, $\mathbf{p}_{I,0:N_t-1}$ in the network models trajectory, and let $\theta_G = \{G_{SBM}, p_{I,0}, p_R\}$ denote all fixed parameters required for initialization of the network.

Community-level dynamics can be captured by measuring infections between and recoveries inside of the network communities.

$$\mathbf{y}_{I,t} = [\Delta N_{I,t}^{(11)}, \Delta N_{I,t}^{(12)}, \dots, \Delta N_{I,t}^{(N_c N_c)}] \quad (10)$$

With $\mathbf{y}_{I,0:N_t-1}$ as $\mathbf{y} \in \mathbb{Z}^{N_t \times \binom{N_c}{2}}$, the marginal distributions for the full network trajectory can be found by integrating over ξ .

$$p(\mathbf{y}|\theta_G) = \int_{\xi} p(\mathbf{y}, \xi|\theta_G) p(\xi) d\xi \quad (11)$$

Variance of the MC-simulations are reduced by advancing the network dynamics with (6), which amounts to importance distribution $\pi(\cdot)$.

$$I = \int_{\mathbf{y}} p(\mathbf{y}|\theta_G) \pi(\mathbf{y}|\theta_G) d\mathbf{y} \quad (12)$$

Statistical moments of interest I can be approximated by drawing samples of $\{\mathbf{y}^{(k)}\}_{k=1}^{N_{MC}}$ from $\pi(\cdot)$, where N_{MC} denotes the total number of MC-simulations. The regression performed in this paper will approximate expected and 'worst-case' number of infections between and inside each community.

$$\mathbb{E}_N(\mathbf{y}|\theta_G) = \sum_{k=0}^{N_{MC}-1} \mathbf{y}^{(k)} p(\mathbf{y}^{(k)}|\theta_G) \quad (13)$$

$$P_N(\mathbf{y} \leq \mathbf{y}_{I,\tau}|\theta_G) = \sum_{\mathbf{y} \leq \mathbf{y}_{I,\tau}} p(\mathbf{y}^{(k)}|\theta_G) \quad (14)$$

$\mathbf{y}_{I,\tau}$ denotes the conditional median number of infected for $\mathbf{y}$, given quantile τ . In order to avoid an overrepresentation of endemic states (and degeneracy in the regression problem) MC-simulations are terminated early whenever $N_{I,t} < N_{I,min}$.

Bounds for p_I are retrieved from the reproduction number $\mathcal{R}_0$ of the SBM.

$$p_I = 1 - e^{-\frac{\log 1 - p_R \mathcal{R}_0}{\langle k \rangle}} \quad (15)$$

Where $\langle k \rangle$ is the expected degree for the vertices.

5. Regression

5.1. Regression Model

The regression algorithm aims to identify the bilinear dynamics found in (3), which is achieved by transforming MC-data to the same form.

$$F_{\theta_{rs}}(x_t^{(r)}, x_t^{(s)}, p_{I,t}^{(rs)}) = p_{I,t}^{(rs)} \theta_{rs} \frac{N_{S,t}^{(s)} N_{I,t}^{(r)}}{N_{S,t}^{(s)} + N_{I,t}^{(r)} + N_{R,t}^{(s)}} \quad (16)$$

$F_{\theta_{rs}}$ captures the identified infection dynamic between a single community pair.

5.2. Objective

The regression objectives aims to find good estimators for (13) and (14). For the infection dynamics, this can be decoupled into $\binom{N_c}{2}$ regression objectives.

$$\arg \min_{\hat{\theta}_{\theta_{rs}}} \|y_{I,MC}^{(rs)} - F_{\theta_{rs}}(x_t^{(r)}, x_t^{(s)}, p_{I,t}^{(rs)})\| \quad (17)$$

(17) is linear, and can be solved using least squares in order to approximate (13) with expected infection rate $\hat{\theta}_{rs}$. (14) can be approximated with quantile regression, resulting in a linear program (LP).

$$\arg \min_{\theta_{rs,\tau}} \|f_{\tau}(\theta_{rs,\tau}, y_{MC}^{(rs)}, x_{MC}^{(r)}, x_{MC}^{(s)})\| \quad (18)$$

$$f_{\tau}(\theta_{rs,\tau}, y_t, x_t^{(r)}, x_t^{(s)}) = \begin{cases} (1-\tau)|F_{\theta_{rs}}(x_t^{(r)}, x_t^{(s)}, p_{I,t}^{(rs)})| & \text{if } y_t < F_{\theta_{rs}}(x_t^{(r)}, x_t^{(s)}, p_{I,t}^{(rs)}) \\ \tau|y_t - F_{\theta_{rs}}(x_t^{(r)}, x_t^{(s)}, p_{I,t}^{(rs)})| & \text{else} \end{cases} \quad (19)$$

6. Optimal Control Problem

6.1. Robust Model Predictive Control

$$\arg \min_{w,u} \Phi(w,u) \quad (20)$$

$$s.t. \quad w_0 = \bar{w}_0 \quad (21)$$

$$\tau \leq P(g(w_t) \leq 0) \quad \forall t \setminus 0 \quad (22)$$

$$p_{I,\min} \leq u_t \leq p_{I,\max} \quad \forall t \setminus N_t \quad (23)$$

(21-23) presents a robust Optimal Control Problem (OCP) with objective Φ , state trajectory w and control inputs u . Robustness of the OCP is enforced with *single chance constraints* [9] in (22), which relate probability distributions to the OCP as inequality constraints.

6.2. Regression Model Predictive Control

The OCP for this paper balance the cost of disease spread mitigation against the cost of disease spread. Let $w_t \in \mathbb{R}^{3N_c}$, $u_t \in \mathbb{R}^{2\binom{N_c}{2}+N_c}$ be the compartmental states and control inputs at time t for the OCP. An ordinary differential equation for w can be composed from the identified regression models.

$$F_r(w_t, u_t) = \begin{bmatrix} -\sum_{s \in c} F_{rs}(w_t^{(r)}, w_t^{(s)}, u_t^{(rs)}) \\ -\sum_{s \in c} F_{rs}(w_t^{(r)}, w_t^{(s)}, u_t^{(rs)}) + \alpha w_{I,t}^{(0)} \\ \alpha w_{I,t}^{(0)} \end{bmatrix} \quad (24)$$

$$F(w_t, u_t) = \begin{bmatrix} F_1(w_t, u_t) \\ \vdots \\ F_{N_c}(w_t, u_t) \end{bmatrix} \quad (25)$$

With F identified, the chance constraint (22) can be approximated. Φ can be defined to minimize the number of infected while restricting the usage of the control input, which together is used to formulate the full regression model OCP (26-29)

$$\arg \min_{w,u} \sum_{t=1}^{N_t} \left\{ \frac{\sum w_{I,t}}{N_{pop}} - \frac{W_u}{N_{pop}} (u_t - u_{\max}) \right\} \quad (26)$$

$$s.t \quad w_0 = \bar{w}_0 \quad (27)$$

$$w_{t+1} = F(w_t, u_t | \theta) \quad \forall t \setminus N_t \quad (28)$$

$$p_{I,\min} \leq u_t \leq p_{I,\max} \quad \forall t \setminus N_t \quad (29)$$

7. Simulations

7.1. Structural Inference

In order to demonstrate the control and prediction capabilities of the presented method, a large number of individual-level graphs (N_G) is generated for the MC-simulations. The partitions are given a population size large enough to be approximated by ODEs, but small enough to give the partitions a considerable variance. The number of partitions (N_b) is set high in order to provide a more granular transition through the community detection threshold (Figure 3).

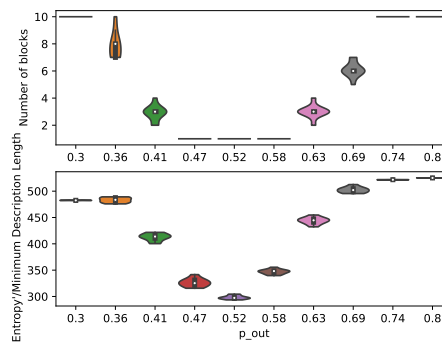


Figure 3. Inferred number of communities and minimum description length for graphs generated under different $p_{SBM,out}$.

Table 1. Graph generation parameters.

$p_{SBM,out}$	$[0.3 : 0.05 : 0.8]$	$N_{pop,b}$	100
$p_{SBM,in}$	0.5	N_G	100
N_b	10		

Estimates for (9) are computed using an importance sampling algorithm [10] provided in graph-tool [11], with a default uninformative prior distribution $P(c)$.

7.2. Monte-Carlo Sampling

Other parameters related to MC-simulations, regression and optimal control remain fixed in order to generate comparable results. $p_{I,t}$ is for the MC-simulations constrained to change at the same time instances as the control input u_t , which in this situation changes every 7 timesteps. Since the expected vertex degree is uniform over all vertices in the planted SBM, bounds for u can be found using basic reproduction numbers $\mathcal{R}_0$.

Table 2. Fixed parameter set for the simulation scenarios

p_{I0}	0.1	p_{R0}	0.05
$\mathcal{R}_{0,min}$	0.1	$\mathcal{R}_{0,max}$	1.5
W_u	300	N_t	56
N_{MC}	100		

Due to the high dimensionality of the inference problem, the MC-algorithm is implemented in Data-Parallel C++ using SYCL [12].

Note: Draft submission is run on a consumer-grade, local GPU, (Nvidia RTX3060, $N_{pop,b} = 40, N_G = 10$), the final result will be processed on a HPC-cluster (toolchain is verified to work on the cluster, access and setup is complete, missing final results).

Note: The network scale, MC-simulations and number of graphs is therefore insufficient to properly excite the network and identify accurate regression parameters.

7.3. Optimization

The regression OCP (26-29) is constructed symbolically in Casadi [13] and solved using IPOPT [14].

8. Results

Figures 3, respectively show the structural, regression and optimization results from the simulations.

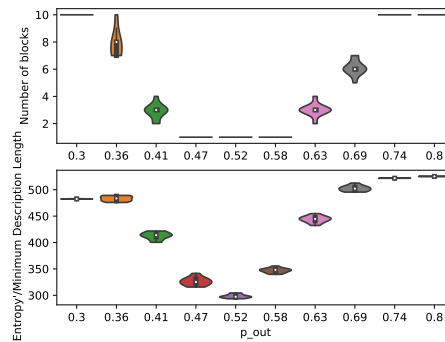


Figure 4. Distributions for internal (θ_{in}) and outgoing (θ_{out}) regression parameters, averaged over the N_G generated graphs. (Result incomplete, Needs HPC to converge)

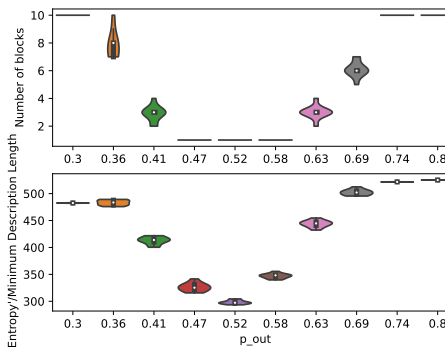


Figure 5. Distributions for objective values, averaged over the N_G generated graphs. (Needs HPC to converge)

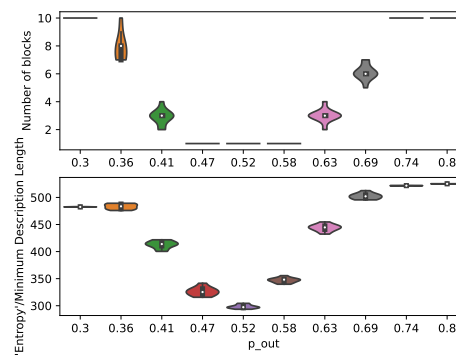


Figure 6. Optimal control trajectory for $p_{SBM,out} = 0.41$. Predictions and solutions for least squares and quantile regression are shown with dotted lines and percentiles. (Needs HPC to converge)

9. Discussion

The results from structural inference (Figure 3) illustrates the varying heterogeneity between the partitions of the planted SBMs. These structural changes impacts the number of communities identified, and thereby the number of regression parameters θ and control inputs ($N_u = \binom{N_c}{2}$).

The best optimal solutions are achieved when the partition layer is controlled directly based on its $\binom{N_b}{2}$ known control inputs, which is illustrated both in the objective plot and the trajectory plot(s). On the contrary, objective values for simulations with uniform control inputs ($N_u = 1$) are penalized in (26) because of its excessive control input usage.

Optimal control on the identified communities manages to exploit the structural properties of the networks to reduce the dimensionality of the control inputs, which can be seen when $p_{SBM,out}$ is close (but not equal) to $p_{SBM,in}$.

The assignment of u to p_I lacks practical reliability, since it is difficult to associate the adjustment of p_I between individuals/communities to real intervention measures. Its continuous, real value instead gives it ideal reliability to the community SIR-ODE, which it will converge to in the large population limit (given a complete graph, $p_{SBM,in} = p_{SBM,out} = 1$). Associating spatial dynamics of epidemics (mobility tracking of individuals, restrictions on schools, workplaces, ...) would improve the practical reliability.

References

1. Ritchie, H.; Mathieu, E.; Rod s-Guirao, L.; Appel, C.; Giattino, C.; Ortiz-Ospina, E.; Hasell, J.; Macdonald, B.; Beltekian, D.; Roser, M. Coronavirus Pandemic (COVID-19). *Our World in Data* **2020**. <https://ourworldindata.org/coronavirus>.
2. Kermack, W.O.; McKendrick, A.G. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character* **1927**, *115*, 700–721.
3. Barab si, A.L.; P sfai, M. *Network science*; Cambridge University Press: Cambridge, 2016.
4. Deville, P.; Song, C.; Eagle, N.; Blondel, V.D.; Barab si, A.L.; Wang, D. Scaling identity connects human mobility and social interactions. *Proceedings of the National Academy of Sciences* **2016**, *113*, 7047–7052, [<https://www.pnas.org/doi/pdf/10.1073/pnas.1525443113>]. doi:10.1073/pnas.1525443113.
5. Peixoto, T.P. Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. *Phys. Rev. X* **2014**, *4*, 011047. doi:10.1103/PhysRevX.4.011047.
6. Peixoto, T.P. Bayesian Stochastic Blockmodeling, 2019. doi:10.1002/9781119483298.ch11.
7. Brunton, S.L.; Proctor, J.L.; Kutz, J.N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* **2016**, *113*, 3932–3937, [<https://www.pnas.org/doi/pdf/10.1073/pnas.1517384113>]. doi:10.1073/pnas.1517384113.
8. Kloek, T.; van Dijk, H.K. Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo. *Econometrica* **1978**, *46*, 1–19.
9. Shapiro, A.; Dentcheva, D.; Ruszczy ski, A. *Lectures on stochastic programming. Modeling and theory*; Society for Industrial and Applied Mathematics, 2009. doi:10.1137/1.9780898718751.

10. Peixoto, T.P. Nonparametric weighted stochastic block models. *Physical Review E* **2018**, *97*. <https://doi.org/10.1103/physreve.97.012306>
11. Peixoto, T.P. The graph-tool python library **2017**. doi:10.6084/m9.figshare.1164194.v14.
12. Maria Rovatsou, L.H.; Keryell, R. SYCL Specification. *Khronos Registry* **2022**.
13. Andersson, J.A.E.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation* **2019**, *11*, 1–36. doi:10.1007/s12532-018-0139-4.
14. Wächter, A.; Biegler, L.T. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming, 2004.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.