Network Reconstruction via Sensitivity Analysis

Gerrit Großmann

Saarland Informatics Campus, DFKI, Saarbrücken, Germany
gerrit.grossmann@dfki.de

1 Introduction

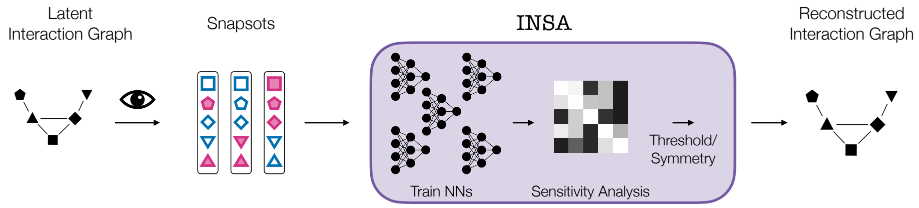

Fig. 1. Method Overview: The data is given as independent snapshots, produced by the interactions of heterogeneous agents on an underlying latent interaction graph with two states (red and blue). We train a neural network for each agent to predict its state from the other four states within each snapshot. The interaction graph is inferred by examining the sensitivities of the trained NNs.

Complex systems comprised of interacting agents or components pervade nature, society, and technology. This work explores the problem of inferring the unknown interaction structure (represented as a graph) from measurements of the individual agents (represented as nodes) in an unsupervised manner.

Motivation. Human interactions catalyze the spread of epidemics or rumors; the interplay between brain regions underpins cognitive processes and diseases; and enzyme interactions steer cellular metabolism. Although these models vary immensely——ranging from stochastic to deterministic dynamics, from homogeneous to heterogeneous agents, and from physical connections to abstract correlations—deciphering interaction structures is crucial to unveil deeper insights, enhance predictions, and infer interventions.

Overview. We examine a scenario in which: (1) each agent's dynamical evolution is intrinsically stochastic; (2) the agents are heterogeneous (each agent is uniquely parametrized); (3) the governing rules are unknown; and (4) we are limited to observing isolated system *snapshots* with no information on the temporal evolution. Hence, we propose INSA (Network *In*ference with *S*ensitivity *A*nalysis). Our approach (cf. Figure 1) involves two steps: (1) We reframe the problem as a prediction task, particularly as an *unsupervised masked reconstruction* objective. That is, we mask the measured *state* of each agent and predict (*reconstruct*) it based on the measured states of the remaining agents. We pursue a 'model-free' approach, utilizing a neural network with minimal parametric constraints. Each agent is equipped with a dedicated neural network to reconstruct its state. (2) Sensitivity analysis is employed on each neural network to infer which agents are most relevant for an agent's predicted state. The resulting *importance scores* are then transformed into binary format and symmetrized to derive an interaction graph.

Related Work. The endeavor to infer interactions finds relevance across a multitude of scientific domains, resulting in an extensive body of literature too vast to review comprehensively here. A significant portion of these techniques serve specific domains, such as epidemics [1], genetics [1], or neuroscience [6]. More generalized methods often presuppose the availability of time-series data [5,4,12,7]. Of particular relevance to our work is GINA [2], which addresses a similar problem setting; however, it involves homogeneous agents and sidesteps the need for thresholding. Another noteworthy approach is the *neural Granger causality* [10], which measures impact scores by assessing how effectively one agent can predict another's state. The abundant literature on neural sensitivity analysis often intersects with studies on feature importance and the field of explainable AI. In our work, we utilize saliency values (specifically, the gradient on the input data). Similar methodologies are elaborated upon in [9,8]. We also reference techniques that evaluate how output quality diminishes upon modifying or omitting input features; for a comprehensive discussion on this, we refer to [11].

Code Availability. Code is made available at `github.com/GerritGr/Insa`.

2 Method

Problem Setting. We aim to reconstruct the latent interaction graph of a complex system, represented by a symmetric and binary adjacency matrix $\mathbf{A}$, with n agents/nodes. We are provided with a set of m snapshots. Each snapshot k identifies each agent i with a state d-dimensional feature vector (resp. state) $\mathbf{x}_{ik} \in \mathbb{R}^d$. Here, we focus on binary states, that is, $d = 1$ and $\mathbf{x}_{ik} \in \{0,1\}$. Importantly, we impose no constraints or assumptions regarding the relationships between snapshots, such as their sequence or the specific timepoints of their measurement. E.g., snapshots may stem from different experiments or could be temporally decoupled due to large observation intervals.

Unsupervised Masked Reconstruction. For each agent i, we train a neural network $\mathrm{NN}_i : \mathbb{R}^{(n-1)\times d} \to \mathbb{R}^d$, that takes as input the values of all other agents and predicts the value of i within a snapshot. We train all n neural networks on a subset (training set) of the total m snapshots.

Sensitivity Analysis. We compute the impact agent i has on agent j for all $i \neq j$. Therefore, we quantify the effect of i on the output of $\mathrm{NN}_j(\cdot)$ by asking: On average, by how much does the modification of $\mathbf{x}_{ik}$ change the output of $\mathrm{NN}_j(\mathbf{x}_{ik})$ (where k ranges over the remaining validation set of snapshots)? We test two modification strategies: (1) Masking the values of $\mathbf{x}_{ik}$ ($\forall k$) (e.g., set it to a dummy value); and (2) Randomly altering it to an incorrect value. The influence is quantified by the relative drop in accuracy between the original output and the output after modification. In addition, we compute *saliency values* by determining the (absolute value of the) gradient on $\mathbf{x}_{ik}$. Note that a significant conceptual distinction exists between the two approaches: While the former measures how modifications to an agent affect the *accuracy* of the prediction, the latter assesses the degree of change in the prediction, without considering whether it results in improved or diminished accuracy. We collect the impact scores in a matrix $\mathbf{S} \in \mathbb{R}^{n \times n}_{\geq 0}$, where s_{ij} is the impact from agent i to j (diagonal entries are dummy values).

Thresholding and Symmetry. After assembling the pairwise impact scores, our objective is to transform $\mathbf{S}$ into a symmetric, binary adjacency matrix $\mathbf{A}$. Several methods were evaluated for this purpose. A shared preliminary step across all methods is row normalization of $\mathbf{S}$. Subsequent to this normalization, we attempted to binarize each row in a non-parametric manner, leveraging clustering algorithms such as hierarchical bottom-up clustering and max-margin clustering. Empirical evaluations suggested that iterative normalization of both rows and columns, followed by clustering of the entire matrix $\mathbf{S}$, was the most effective. This process yielded a binary (but non-symmetrical) matrix $\widehat{\mathbf{S}} \in \{0,1\}^{n \times n}$. Subsequently, an edge was posited to exist if and only if both directions indicated its presence, expressed as $a_{ij} = \min(s_{ij}, s_{ji})$. This approach consistently outperformed alternative strategies, such as matrix symmetrization before binarization, employing the max function in place of min, or constraining the clustering algorithm to produce identical clusters for both s_{ij} and s_{ji}.

Results. We tested three different networks (Erdős-Rényi, Watts–Strogatz, 2d-grid) with 50 nodes each using four binary dynamical models *Cascade*, *Majority Vote*, *Opinion*, and *SIS* based on continuous-time Markov chain semantics. In the opinion model, each agent is randomly assigned a type that determines if it aims to maximize or minimize its agreement with neighboring agents. Otherwise, the individual parameters specify the reactivity of the agents (e.g., how infectious they are). We first sample unique parameters for each agent, then we sample 30 thousand snapshots from random initial configurations and stop the simulation after a random time duration (in quasi-equilibrium). For the baseline, we use correlation (Corr), mutual information (MI), and partial correlation (ParCorr) methods based on the Python package *netrd* [3]. We report the *edit distance, that is the* distance of the (upper triangular) ground truth adjacency matrix with the predicted adjacency matrix (cf. Table 1). Details are provided in the repository.

3 Conclusions and Future Work

We introduce a 'model-free' approach for reconstructing networks, particularly in the face of heterogeneous agents and potentially uncorrelated measurements. Our work validates the underlying hypothesis that sensitivity analysis can differentiate between neighbors and non-neighbors. Looking ahead, we perceive the primary challenge to test our method in real-world situations, complemented by comprehensive benchmarking, potentially alongside causal inference techniques. Another promising avenue for future research is the incorporation of more robust regularization in predictions, such as through weight sharing or conditional predictions. This could enable prediction networks to learn agent similarities rather than reconstructing agent states in isolation.

Table 1. Results: [Smaller is better] Results of different graph inference methods.

Model	Graph	INSA (our method)	Corr	MI	ParCorr
Cascade	ER	**480** ± 41.8	697 ± 363.4	858 ± 482.1	717 ± 84.3
	Grid	**94** ± 10.3	149 ± 71.8	98 ± 11.3	169 ± 33.7
	WS	**125** ± 7.5	485 ± 240.4	325 ± 209.2	559 ± 117.1
Majority Vote	ER	755 ± 47.6	722 ± 80.3	**703** ± 78.2	721 ± 62.3
	Grid	**3** ± 3.9	44 ± 6.6	39 ± 7.3	10 ± 4.7
	WS	**31** ± 5.6	455 ± 112.1	455 ± 112.1	44 ± 5.9
Opinion	ER	**184** ± 10.2	684 ± 108.2	249 ± 18.3	761 ± 107.1
	Grid	**52** ± 20.5	158 ± 57.6	64 ± 11.9	155 ± 36.1
	WS	117 ± 18.7	653 ± 129.0	**127** ± 16.2	646 ± 152.9
SIS	ER	270 ± 41.4	830 ± 168.4	**225** ± 32.3	674 ± 211.3
	Grid	**18** ± 6.7	69 ± 61.2	27 ± 9.3	160 ± 70.8
	WS	78 ± 15.2	408 ± 341.0	**58** ± 15.7	563 ± 168.5

References

1. Barman, S., Kwon, Y.K.: A boolean network inference from time-series gene expression data using a genetic algorithm. Bioinformatics 34(17), i927–i933 (2018)
2. Großmann, G., Zimmerlin, J., Backenköhler, M., Wolf, V.: Unsupervised relational inference using masked reconstruction. Applied Network Science 8(1), 1–21 (2023)
3. Hartle, H., Klein, B., McCabe, S., Daniels, A., St-Onge, G., Murphy, C., Hébert-Dufresne, L.: Network comparison and the within-ensemble graph distance. Proceedings of the Royal Society A 476(2243), 20190744 (2020)
4. Ke, N.R., Bilaniuk, O., Goyal, A., Bauer, S., Larochelle, H., Schölkopf, B., Mozer, M.C., Pal, C., Bengio, Y.: Learning neural causal models from unknown interventions. arXiv preprint arXiv:1910.01075 (2019)
5. Kipf, T., Fetaya, E., Wang, K.C., Welling, M., Zemel, R.: Neural relational inference for interacting systems. In: International conference on machine learning. pp. 2688–2697. PMLR (2018)
6. Korhonen, O., Zanin, M., Papo, D.: Principles and open questions in functional brain network reconstruction. Human Brain Mapping 42(11), 3680–3711 (2021)
7. Löwe, S., Madras, D., Zemel, R., Welling, M.: Amortized causal discovery: Learning to infer causal graphs from time-series data. In: Conference on Causal Learning and Reasoning. pp. 509–525. PMLR (2022)
8. Pizarroso Gonzalo, J., Portela González, J., Muñoz San Roque, A.: Neuralsens: sensitivity analysis of neural networks (2022)
9. Rebuffi, S.A., Fong, R., Ji, X., Vedaldi, A.: There and back again: Revisiting backpropagation saliency methods. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8839–8848 (2020)
10. Tank, A., Covert, I., Foti, N., Shojaie, A., Fox, E.B.: Neural granger causality. IEEE Transactions on Pattern Analysis and Machine Intelligence 44(8), 4267–4279 (2021)
11. Yang, J.B., Shen, K.Q., Ong, C.J., Li, X.P.: Feature selection for mlp neural network: The use of random permutation of probabilistic outputs. IEEE Transactions on Neural Networks 20(12), 1911–1922 (2009)
12. Zhang, Z., Zhao, Y., Liu, J., Wang, S., Tao, R., Xin, R., Zhang, J.: A general deep learning framework for network reconstruction and dynamics learning. Applied Network Science 4(1), 1–17 (2019)