Efficiently Learning the Topology and Behavior of Networked Dynamical Systems Via Active Queries

Daniel J. Rosenkrantz,[1,3] Abhijin Adiga,[1] Madhav V. Marathe,[1,2] Zirou Qiu,[1,2]

S. S. Ravi [1,3], Richard E. Stearns [1,3], Anil Vullikanti [1,2]

[1] Biocomplexity Institute and Initiative, University of Virginia. [2] Computer Science Dept., University of Virginia. [3] Computer Science Dept., University at Albany – SUNY.

Acknowledgment: This work was partially supported by University of Virginia Strategic Investment Fund award number SIF160, Virginia Department of Health grant VDH-21-501- 0135-1, NSF Grants OAC-1916805 (CINES), CCF-1918656 (Expeditions), IIS-1931628, IIS-1955797, and NIH grant R01GM109718.

Introduction

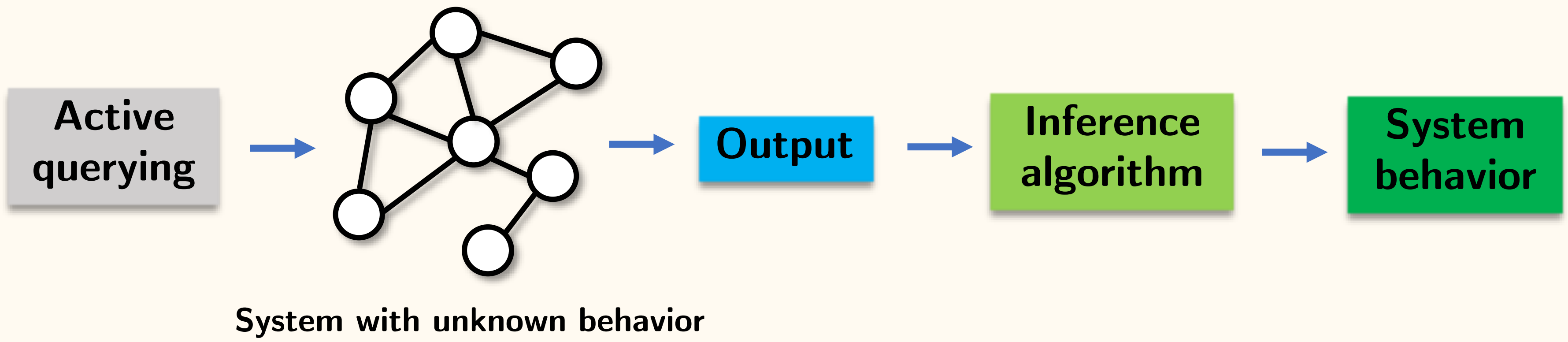
- Inferring **features** of unknown systems is an active research area in ML [1].
- Our focus: Inferring features of **discrete dynamical systems**.
  - Previous work [2,3,4]: Inferring system behavior **given** network topology.
  - Our work: Inferring **both** system behavior and network topology (“full inference”).

Introduction

- Inferring **features** of unknown systems is an active research area in ML [1].
- <u>Our focus</u>: Inferring features of **discrete dynamical systems**.
  - Previous work [2,3,4]: Inferring system behavior **given** network topology.
  - Our work: Inferring **both** system behavior and network topology ("full inference").

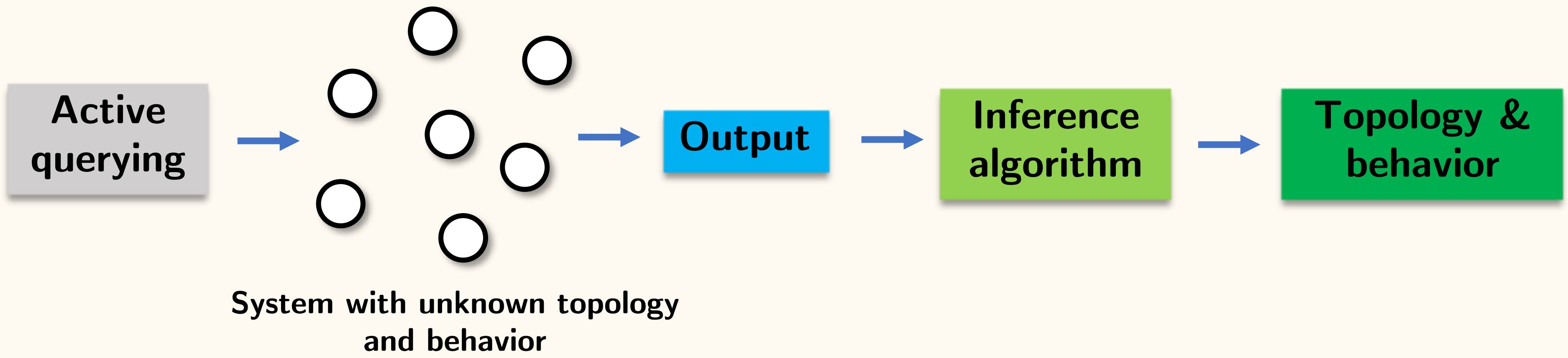
Introduction

- Inferring **features** of unknown systems is an active research area in ML [1].
- <u>Our focus</u>: Inferring features of **discrete dynamical systems**.
  - Previous work [2,3,4]: Inferring system behavior **given** network topology.
  - Our work: Inferring **both** system behavior and network topology (“full inference”).



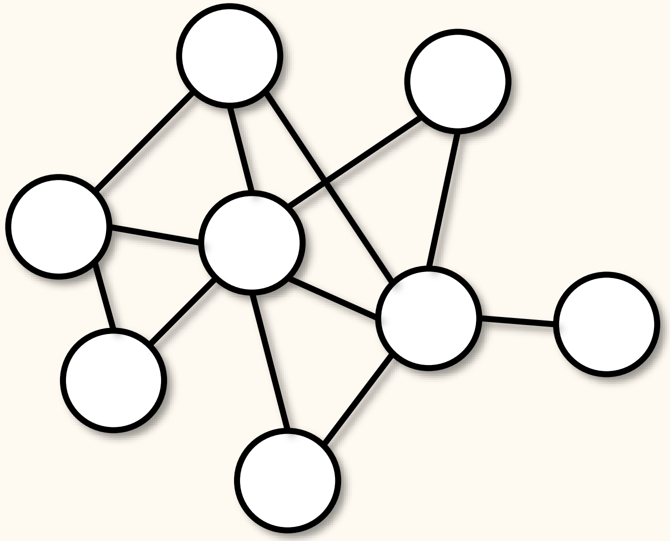
Introduction

- Inferring **features** of unknown systems is an active research area in ML [1].
- Our focus: Inferring features of **discrete dynamical systems**.
  - Previous work [2,3,4]: Inferring system behavior **given** network topology.
  - Our work: Inferring **both** network topology and system behavior ("full inference").



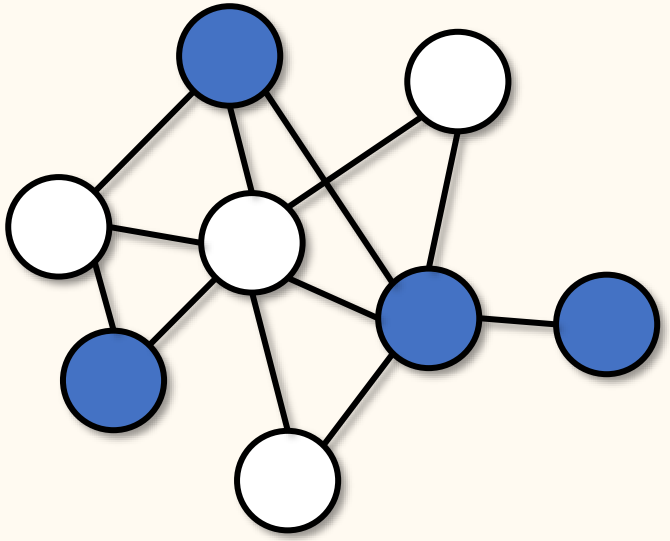
Components of a Dynamical System

- A graph $G = (V, E)$.
- Each node v has:
  - A **binary** state $\{0, 1\}$.
  - A Boolean function f_v (i.e., **behavior**)
  - f_v gives the next state.
- **Synchronous** state update (Notation: SyDS).
- **Symmetric function**: Value decided by the number of 1's in the input.



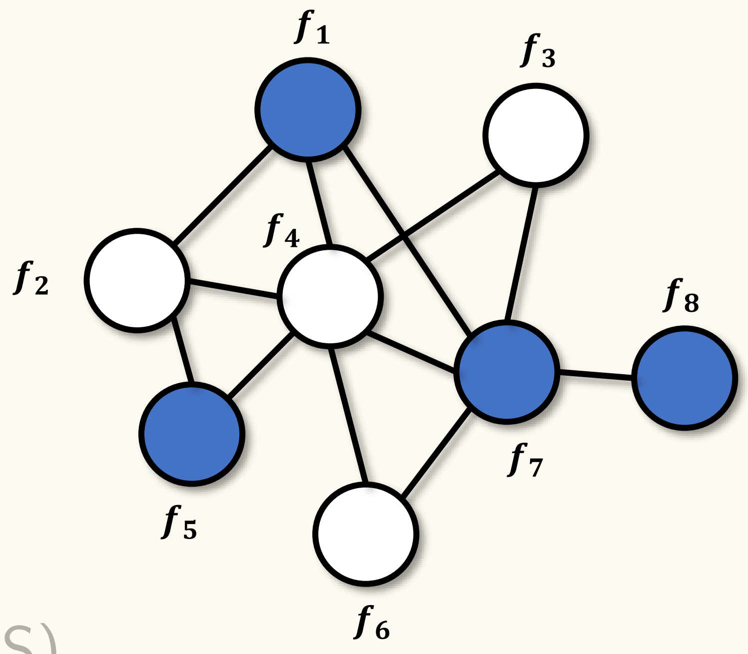
Components of a Dynamical System

- A graph $G = (V, E)$.
- Each node v has:
  - A **binary** state $\{0, 1\}$.
  - A Boolean function f_v (i.e., **behavior**)
  - f_v gives the next state.
- **Synchronous** state update (Notation: SyDS).
- **Symmetric function**: Value decided by the number of 1's in the input.



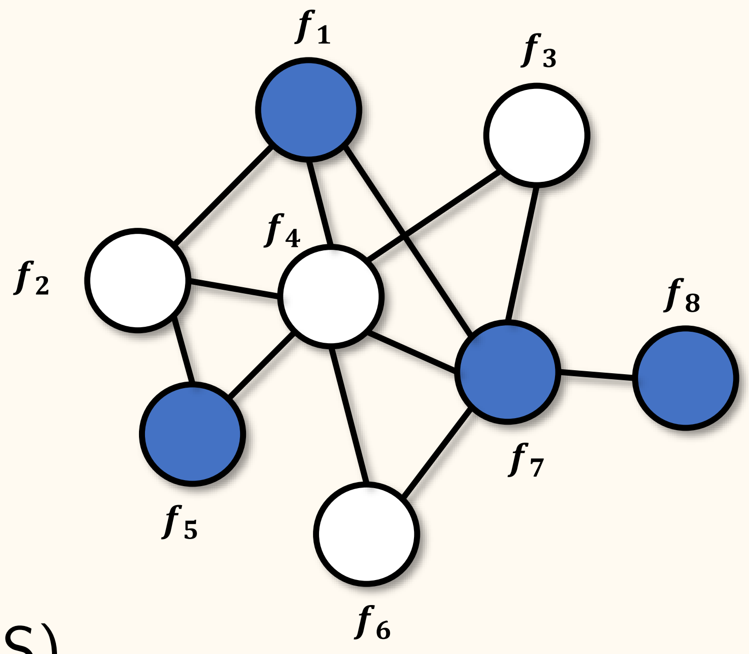
Components of a Dynamical System

- A graph $G = (V, E)$.
- Each node v has:
  - A **binary** state $\{0, 1\}$.
  - A Boolean function f_v (i.e., **behavior**)
  - f_v gives the next state.
- **Synchronous** state update (Notation: SyDS).
- **Symmetric function**: Value decided by the number of 1's in the input.



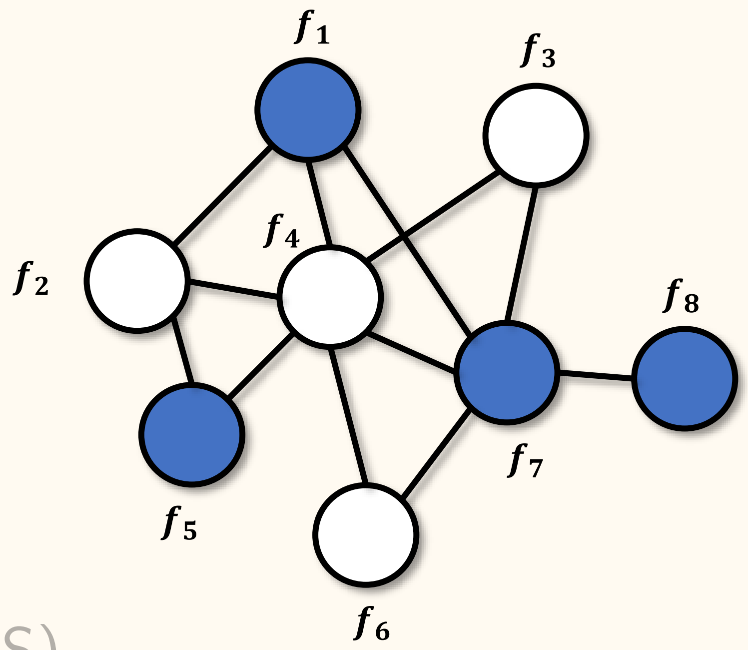
Components of a Dynamical System

- A graph $G = (V, E)$.
- Each node v has:
  - A **binary** state $\{0, 1\}$.
  - A Boolean function f_v (i.e., **behavior**)
  - f_v gives the next state.
- **Synchronous** state update (Notation: SyDS).
- **Symmetric function**: Value decided by the number of 1's in the input.



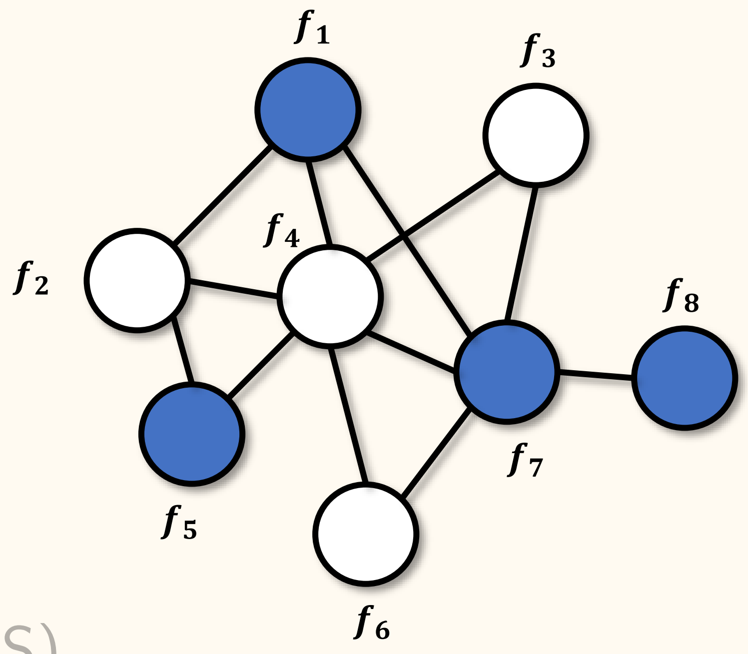
Components of a Dynamical System

- A graph $G = (V, E)$.
- Each node v has:
  - A **binary** state $\{0, 1\}$.
  - A Boolean function f_v (i.e., **behavior**)
  - f_v gives the next state.
- **Synchronous** state update (Notation: SyDS).
- **Symmetric function**: Value decided by the number of 1's in the input.



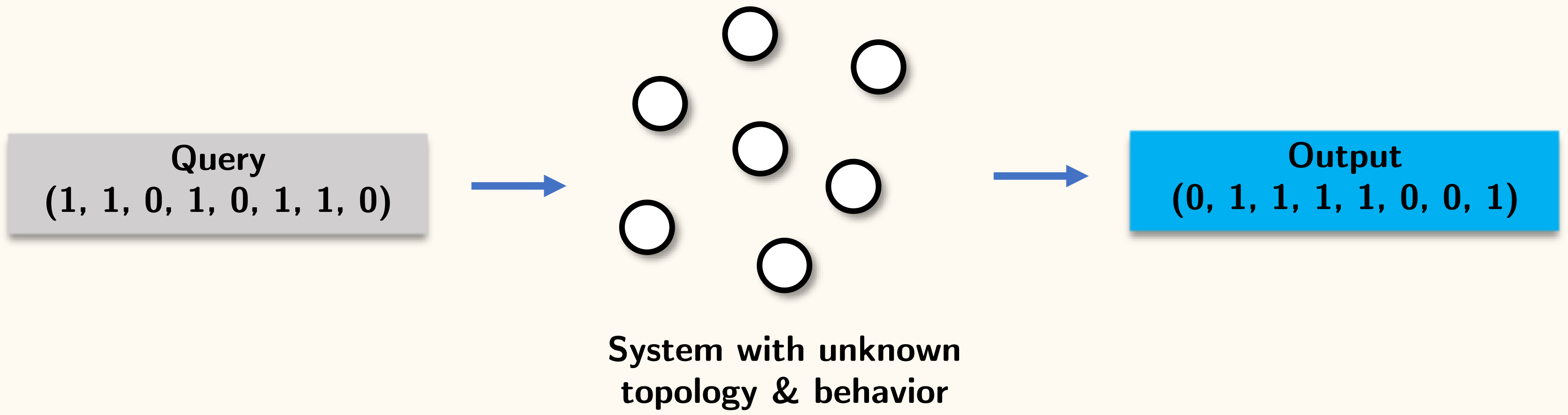
Components of a Dynamical System

- A graph $G = (V, E)$.
- Each node v has:
  - A **binary** state $\{0, 1\}$.
  - A Boolean function f_v (i.e., **behavior**)
  - f_v gives the next state.
- **Synchronous** state update (Notation: SyDS).
- **k-Threshold function**: Value is 1 iff *at least* k inputs are 1.



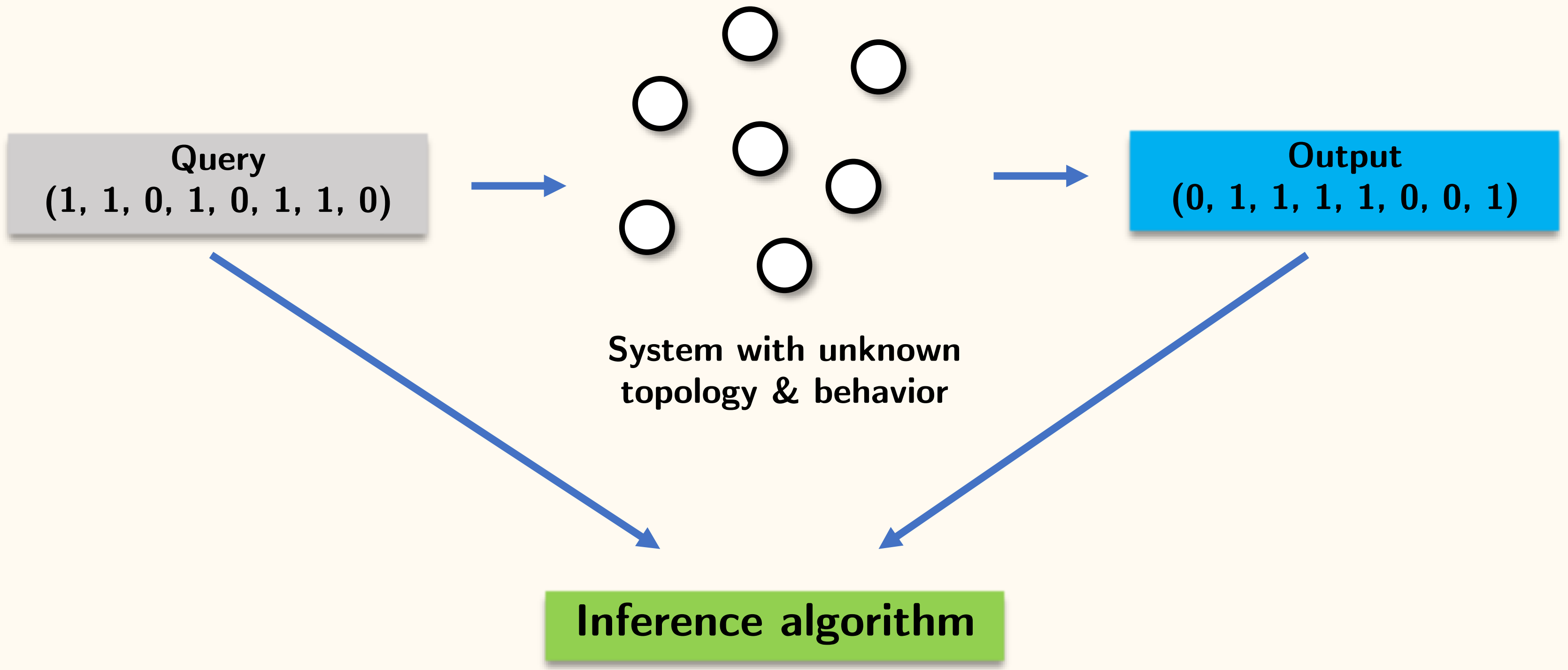
Our Query Model

- Active querying



Our Query Model

- Active querying



Our Query Model

- **Batch mode**
  - All queries are *given together*.
- **Adaptive mode**
  - Queries are specified in *multiple stages*.
  - New queries may be based on the responses to the previous queries.

Our Query Model

- **Batch mode**
  - All queries are *given together*.
- **Adaptive mode**
  - Queries are specified in *multiple stages*.
  - New queries may be based on the responses to the previous queries.

Summary of Results

- To infer **Symmetric-SyDS**
  - $O(n^2)$ queries under the batch mode.
  - $O(n + m \log n)$ queries under the adaptive mode.
- To infer **Threshold-SyDS**
  - $O(n + m \log n)$ queries under the adaptive mode.
  - $O(n \Delta \log n)$ queries under the batch mode w.h.p., where Δ is the maximum degree.
- **Lower bound** for batch mode
  - To infer both topology and behavior, a query set of size $\Omega(n \log n)$ is needed.
- Experimental study of how the number of queries varies with *network structure* and *system parameters*.

Summary of Results

- To infer **Symmetric-SyDS**
  - $O(n^2)$ queries under the batch mode.
  - $O(n + m \log n)$ queries under the adaptive mode.
- To infer **Threshold-SyDS**
  - $O(n + m \log n)$ queries under the adaptive mode.
  - $O(n \Delta \log n)$ queries under the batch mode w.h.p., where Δ is the maximum degree.
- **Lower bound** for batch mode
  - To infer both topology and behavior, a query set of size $\Omega(n \log n)$ is needed.
- Experimental study of how the number of queries varies with *network structure* and *system parameters*.

Summary of Results

- To infer **Symmetric-SyDS**
  - $O(n^2)$ queries under the batch mode.
  - $O(n + m \log n)$ queries under the adaptive mode.
- To infer **Threshold-SyDS**
  - $O(n + m \log n)$ queries under the adaptive mode.
  - $O(n \Delta \log n)$ queries under the batch mode w.h.p., where Δ is the maximum degree.
- **Lower bound** for batch mode
  - To infer both topology and behavior, a query set of size $\Omega(n \log n)$ is needed.
- Experimental study of how the number of queries varies with *network structure* and *system parameters*.

Summary of Results

- To infer **Symmetric-SyDS**
  - $O(n^2)$ queries under the batch mode.
  - $O(n + m \log n)$ queries under the adaptive mode.
- To infer **Threshold-SyDS**
  - $O(n + m \log n)$ queries under the adaptive mode.
  - $O(n \Delta \log n)$ queries under the batch mode w.h.p., where Δ is the maximum degree.
- **Lower bound** for batch mode
  - To infer both topology and behavior, a query set of size $\Omega(n \log n)$ is needed.
- Experimental study of how the number of queries varies with *network structure* and *system parameters*.

References

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

[2] Adiga, A., Kuhlman, C. J., Marathe, M., Ravi, S. S., and Vullikanti, A. *PAC learnability of node functions in net- worked dynamical systems*. In Proc. ICML 2019, pp. 82–91, 2019.

[3] Adiga, A., Kuhlman, C. J., Marathe, M. V., Ravi, S. S., Rosenkrantz, D. J., and Stearns, R. E. *Learning the behavior of a dynamical system via a "20 questions" approach*. In Thirty second AAAI Conference on Artificial Intelligence, pp. 4630–4637, 2018.

[4] He, X., Xu, K., Kempe, D., and Liu, Y. Learning influence functions from incomplete observations. *In Advances in Neural Information Processing Systems*, pp. 2073–2081, 2016.

Note: See the paper for a full list of references.

Questions?

zq5au@virginia.edu, ssravi@virginia.edu.