PAC Learnability of Node Functions in Networked Dynamical Systems

Abhijin Adiga, Chris J. Kuhlman, Madhav V. Marathe *
S. S. Ravi, and Anil K. Vullikanti *

Network Systems Science and Advanced Computing Division
Biocomplexity Institute and Initiative
University of Virginia

*Also with the Computer Science Department, University of Virginia.
Inferring network dynamical systems is a broad and well-studied area.

We consider the problem of inferring the node functions of a networked dynamical system.

- **Observation model:** Probably Approximately Correct (PAC) learning
- **Model class:** Threshold dynamical systems
Motivation and previous work

- PAC learning network dynamical systems:
  - Learning influence functions of nodes in stochastic networked dynamical systems [Narasimhan et al., 2015; He et al., 2016].
  - Extensive research on PAC learning threshold functions, and in general, Boolean functions [Hellerstein & Servedio 2007].

- Practical Use of Threshold models:
  - Widespread application in modeling protests, information diffusion (e.g., word of mouth, social media), adoption of practices (e.g., contraception, innovation), transmission of emotions, etc. (Granovetter 1978).
  - Social science network experiments (Centola 2010).
  - General inference: (González-Bailón et al. 2011; Romero, Meeder, and Kleinberg 2011) present methods to infer thresholds from social media data.
Threshold propagation model

- Closed neighborhood of a vertex $v$: $N[v]$
- Every node is associated with a threshold: $t(v)$

\[
q_{i+1}(v) = \begin{cases} 
1, & \sum_{v' \in N[v]} q_i(v') \geq t(v) \\
0, & \text{otherwise}
\end{cases}
\]

- $t(a) = 1$, $t(b) = 1$, $t(c) = 2$, $t(d) = 2$
Probably Approximately Correct (PAC) learning framework

- Pairs of labeled configurations from a distribution $\mathcal{D}$

+ means $q'$ is the successor of $q$. Otherwise, it is not.

User knows:
- Network (undirected, unweighted)
- Concept class: threshold functions

Questions of interest
- Are threshold dynamical systems efficiently learnable?
- Sample complexity: How many examples (i.e., pairs of configurations) are sufficient to infer the dynamical system?
- Is there an efficient learning algorithm?
- How do these algorithms perform on real-world networks?
Results

Sample complexity

Threshold dynamical systems are PAC learnable.

- Upper bound on sample complexity $\mathcal{M}(\epsilon, \delta)$:
  
  $$\mathcal{M}(\epsilon, \delta) \leq \frac{1}{\epsilon} \left( n \log(d_{\text{avg}} + 3) + \log(1/\delta) \right).$$

  We also extend the bound to other classes of threshold functions.

- Lower bounds on sample complexity:
  - $\Omega(n/\epsilon)$ using Vapnis-Chervonenkis (VC) dimension of the hypothesis space of threshold functions.
  - It is within a factor $O(\log n)$ of the upper bound.
  - Tight example: When the underlying graph is a **clique**, the VC dimension of the hypothesis space is $\leq n + 1$. 
Results

Algorithmic efficiency

Hardness of learning depends on negative examples.
- When there are both positive and negative examples, the hypothesis class of threshold functions is not efficiently PAC learnable, unless the complexity classes NP and RP (Randomized Polynomial time) coincide.

Efficient learning algorithms:
- When there are only positive examples, we present an algorithm which learns in time $O(|\mathcal{E}|n)$, where $\mathcal{E}$ is the set of examples and $n$ is the number of nodes.
- **Exact algorithm**: When a set $\mathcal{E}_N$ of negative examples is also given, we present a dynamic programming algorithm that learns in time $O(2^{|\mathcal{E}_N|}\text{poly}(n))$, which is polynomial when $|\mathcal{E}_N| = O(\log n)$.
- **Approximation algorithm**: Using submodular function maximization under matroid constraints, we present an efficient learner which is consistent with all the positive examples and at least $(1 - 1/e)$ fraction of the negative examples.
### Results

#### Experiments

**Accuracy and sample complexity**
- Effect of graph size
- Effect of graph density
- Effect of distributions for sampling configurations

| Network     | Properties       | \( n \) | \( |E| \) | \( d_{ave} \) | \( d_{max} \) |
|-------------|------------------|--------|-------|---------------|---------------|
| Jazz        |                  | 198    | 2742  | 27.70         | 100           |
| NRV         |                  | 769    | 4551  | 11.84         | 20            |
| euEmall     |                  | 986    | 16064 | 32.58         | 345           |
| Ran Reg\(^a,1\) | \( n d_{avg}/2 \) | 10     | 10    |               |               |
| Scl free\(^a,2\) | \( \sim n d_{avg}/2 \) | 9.5–9.9 | 13–149 |               |               |
| Cliques\(^3\) | \( n_q n_c(n_c-1)/2 \) | \( n_c - 1 \) | \( n_c - 1 \) |               |               |