Inferring Graphs from Cascades: A Sparse Recovery Framework

Jean Pouget-Abadie
Harvard University
jeanpougetabadie@g.harvard.edu

Thibaut Horel
Harvard University
thorel@seas.harvard.edu

ABSTRACT

In the Graph Inference problem, one seeks to recover the edges of an unknown graph from the observations of cascades propagating over this graph. We approach this problem from the sparse recovery perspective. We introduce a general model of cascades, including the voter model and the independent cascade model, for which we provide the first algorithm which recovers the graph's edges with high probability and $\mathcal{O}(s \log m)$ measurements where s is the maximum degree of the graph and m is the number of nodes. Furthermore, we show that our algorithm also recovers the edge weights (the parameters of the diffusion process) and is robust in the context of approximate sparsity. Finally we validate our approach empirically on synthetic graphs.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Parameter Learning*

1. INTRODUCTION

A recent line of research has focused on the Graph Inference Problem: recovering the directed edges of an unknown graph from the observations of a diffusion process propagating on this graph [1, 2, 5]. For example, the Independent Cascade Model, formalised in [3], is a famous diffusion process where each "infected" node has a weighted probability to "infect" its neighbors in the graph. If we are only able to observe the time step at which nodes are infected over several diffusion processes, can we recover the edges and the edge weights of the graph?

Here, we propose a sparse recovery framework to not only solve the Graph Inference Problem, but also recover the unknown weights of the diffusion process, for a large class of discrete time diffusion processes. Recall that for every instance of the diffusion process, the only thing known to the observer are the time steps at which the vertices in the graph become "infected" by the diffusion process. The parallel with sparse recovery problems is as follows: for a given vertex, the (unknown) "influence" of its parents in the graph is a *signal*, that we observe through a series of *measurements*, which are the instances of the diffusion process. The two main challenges to apply sparse recovery tools to this problem are: (1) contrary to a very common assumption, the measurements given by a diffusion process are correlated (2) most diffusion processes lead to non-linear sparse recovery problems.

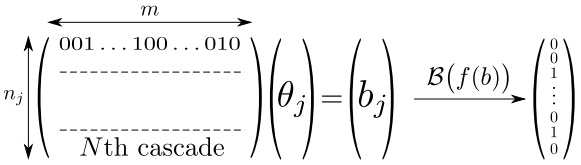

Figure 1: Illustration of the sparse recovery framework. θ_j is the unknown weight vector, b_j is the result of the sparse product $\theta_j \cdot X^t$. We observe bernoulli variables $\mathbf{B}(f(b))$.

In what follows, we first present a general class of discrete-time diffusion processes which encompasses the famous Influence Cascade Model and the Voter Model. For this class of diffusion processes, despite the aforementioned challenges, we show how to recover the unknown parameters with a convergence rate on par with rates observed in the sparse recovery literature. Finally, we validate our approach experimentally, by comparing its performance to prior algorithms on synthetic data.

2. MODEL

Let consider a graph $G = (V, E)$ with $|V| = m$ and where the set of edges E is unknown. For a given vertex j, the cascade model is parameterized by a vector $\theta_j \in \mathbb{R}^m$ where the i-th coordinate of θ_j captures the "influence" of vertex i on j. This influence is 0 if i is not a parent of j.

A cascade is characterized by the propagation of a "contagious" state in discrete time steps. Initially, each vertex is contagious with probability p_{init}. Let us denote by X^0 the indicator vector of the initially contagious vertices. Denoting by X^t the indicator vector of the set of vertices which are contagious at time step t, the probability that j will be contagious at time $t+1$ is given by:

$$\mathbb{P}(X_j^{t+1} = 1 | X^t) = f(\theta_j \cdot X^t) \tag{1}$$

where $f : \mathbb{R} \to [0, 1]$. Conditioned on X^t, this evolution happens independently for each vertex at time $t+1$. We show below that both the independent cascade model and the voter model can be cast in this framework.

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2744107.

Independent Cascade Model Considering the discrete-time IC model, the probability that a susceptible node j becomes infected at the next time step is given by:

$$\mathbb{P}\left[X_j^{t+1} = 1 \,|\, X^t\right] = 1 - \prod_{i=1}^{m} (1 - p_{i,j})^{X_i^t}.$$

Defining $\Theta_{i,j} \equiv \log(1 - p_{i,j})$, this can be rewritten as:

$$\mathbb{P}\left[X_j^{t+1} = 1 \,|\, X^t\right] = 1 - \prod_{i=1}^{m} e^{\Theta_{i,j} X_i^t} = 1 - e^{\theta_j \cdot X^t} \qquad \text{(IC)}$$

Therefore, the independent cascade model fits into the previous framework with $f : z \mapsto 1 - e^z$.

Voter Model Here, nodes can be either *red* or *blue*. Each round, every node j independently chooses one of its neighbors with probability $\Theta_{i,j}$ and adopts their color. Without loss of generality, we can suppose that being *blue* is the contagious state. The cascades stops at a fixed horizon time T or if all nodes are of the same color. If we denote by X^t the indicator variable of the set of blue nodes at time step t, then we have:

$$\mathbb{P}\left[X_j^{t+1} = 1 | X^t\right] = \sum_{i=1}^{m} \Theta_{i,j} X_i^t = \Theta_j \cdot X^t \qquad \text{(V)}$$

Thus, the linear voter model fits into the previous framework with $f : z \mapsto z$.

3. RESULTS

For a given vertex i, we are given a set of measurements, $(X^t, X_i^{t+1})_{t \in \mathcal{T}_i}$ generated from (1). We estimate θ_i via ℓ_1-regularized maximum likelihood estimation:

$$\hat{\theta}_i \in \operatorname*{argmax}_{\theta} \mathcal{L}_i(\theta_i \,|\, x^1, \ldots, x^n) - \lambda \|\theta_i\|_1 \qquad (2)$$

where $\mathcal{L}_i$ is the log-likelihood of θ_i given the observations. We will need the following assumptions:

1. $\log f$ and $\log(1 - f)$ are concave functions.
2. $\log f$ and $\log(1 - f)$ have derivatives bounded in absolute value by $\frac{1}{\alpha}$ for some $\alpha > 0$.
3. denoting by S the support of the true vector of parameters θ_i^*, define $\mathcal{C}(S) \equiv \{X \in \mathbb{R}^m : \|X\|_1 \leq 1 \text{ and } \|X_{S^c}\|_1 \leq 3\|X_S\|_1\}$. We assume that:

$$\forall X \in \mathcal{C}(\mathcal{S}), X^T \nabla^2 \mathcal{L}_i(\theta_i^*) X \geq \gamma \|X\|_2^2$$

for some $\gamma > 0$. γ is called the *restricted eigenvalue*.

Adapting a result from [4], we obtain the following finite-sample guarantee:

THEOREM 1. *Assume 1. to 3. above and define* $n \equiv |\mathcal{T}_i|$. *For any* $\delta \in (0, 1)$, *let* $\hat{\theta}_i$ *be the solution of* (2) *with* $\lambda \equiv 2\sqrt{\frac{\log m}{\alpha n^{1-\delta}}}$, *then:*

$$\|\hat{\theta}_i - \theta_i^*\|_2 \leq \frac{6}{\gamma}\sqrt{\frac{s \log m}{\alpha n^{1-\delta}}} \quad w.p.\ 1 - \frac{1}{e^{n^\delta \log m}} \qquad (3)$$

Assumption 3. above can be replaced by the following data-independent assumption:

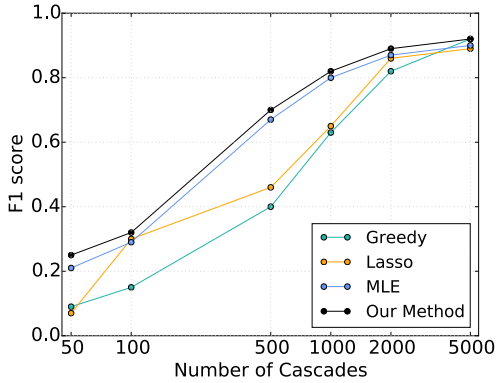

Figure 2: F1 score as a function of the number of observed cascades for a Watts-Strogatz graph, for the Greedy and MLE algorithm from [5], a Lasso algorithm which approximates (2)**, and the penalized log-likelihood program** (2)**.**

3. $\log f$ and $\log(1 - f)$ are ε-concave and the expected gram matrix $\lim_{n\to\infty} \frac{1}{n} X^T X$ has a smallest "restricted" eigenvalue bounded from below by $\gamma > 0$, where X is the $n \times m$ design matrix whose k-th row is X^k.

provided that either $\Omega(s^2 \log m)$ cumulative time steps are observed or $\Omega(s \log m \log^3(s \log m))$ distinct instances of the diffusion process (cascades) are observed.

4. EXPERIMENTS

We compared the performance of Algorithm (2) to prior algorithms for the Graph Inference problem. Given our estimate $\tilde{\Theta}$ of the edge weights, we recover the edges of the graph by simple thresholding: $E = \cup_{j \in V}\{(i, j) : \tilde{\Theta}_{ij} > \eta\}$, for varying values of η. We used the F1-score as a measure of performance: $\text{F1} = 2\text{precision} \cdot \text{recall}/(\text{precision} + \text{recall})$.

The algorithms were tested on several synthetic networks generated from standard social networks model. The results are shown in Figure 4 for the Watts-Strogatz model. The full version of the paper contains more comprehensive experiments.

5. REFERENCES

[1] B. D. Abrahao, F. Chierichetti, R. Kleinberg, and A. Panconesi. Trace complexity of network inference. In *KDD*, pages 491–499, 2013.

[2] H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Schölkopf. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *ICML*, pages 793–801, 2014.

[3] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[4] S. N. Negahban, P. Ravikumar, M. J. Wrainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, December 2012.

[5] P. Netrapalli and S. Sanghavi. Learning the graph of epidemic cascades. *SIGMETRICS Perform. Eval. Rev.*, 40(1):211–222, June 2012.