Contamination-Aware Experiments on Networks

Mine Su Erturk
Graduate School of Business
Stanford University
mserturk@stanford.edu

Eray Turkel
Graduate School of Business
Stanford University
eturkel@stanford.edu

1 Introduction

We study the problem of a decision maker (DM) conducting experiments over time, on a network environment. An ‘experiment' in the context of our model is treating a node in the network. Each node responds to the treatment differently, depending on their (observable) characteristics. We also assume that treating a node creates spillovers on neighboring nodes. The DM's goal is learning the optimal treatment allocation over the network by sequentially conducting local experiments on different nodes.

We assume that there are multiple analysts conducting experiments on the network, as is the case in many online platforms. The DM's experiment creates negative externalities on other ongoing experiments, or the deployed treatments from previously run experiments. We model this as a cost for contaminating the treatment regime of other experimenters. Therefore, the DM has to strike a balance between learning the treatment response function efficiently, and staying within acceptable levels of contamination. We analyze the DM's problem as a sequential decision making problem, which puts our framework closer to the literature on online learning and bandits, instead of using a randomization based inference setting, as is common in the network experimentation literature.

2 Model

There are N units connected on an undirected, unweighted graph. Denote each unit on the graph by $i \in \{1, \dots, N\}$. The units have observable characteristics, $\{x_1, \dots, x_N\}$ with $x_i \in \mathbb{R}^k$. The DM's objective depends on the outcomes of the units, denoted $\{Y_1, \dots, Y_N\}$ with $Y \in \mathbb{R}$, which respond to a treatment regime that will be determined. The decision maker can run experiments sequentially, at time periods $t \in \{1, \dots, T\}$. Let w_{it} denote the treatment assignment for node i during the period of experimentation t, with $w_{it} \in \{0, 1\}$, and denote the vector of treatment assignments at time t by $w_t = [w_{1t}, \dots, w_{Nt}] \in \{0, 1\}^N$. We assume that the experimentation is local, in the sense that every period, only one node can be treated. Thus for all time periods t, $\sum_{i \leqslant N} w_{it} = 1$.

Each node's outcome depends on their characteristics, the treatment they receive and the exposure to the treatment through their neighbors. Define the exposure for node i at period t as e_{it}, which is assumed to be:

$$e_{it} = \frac{\sum_{j \leqslant N} \mathbf{1}(j \in nhbd(i)) w_{jt}}{\sum_{j \leqslant N} \mathbf{1}(j \in nhbd(i))}$$

Y is assumed to be a linear function of the unit characteristics and the exposure to the treatment:

$$Y_{it}(w_{it}, e_{it}) = \beta^T x_i + \Gamma^T x_i e_{it} + \epsilon_i,$$

where $\beta, \Gamma \in \mathbb{R}^k$, and for all nodes i, $\epsilon_i \sim N(0, \sigma_\epsilon^2)$.

The DM's goal is to choose a treatment assignment policy over the network based on the value of Γ. Denote the final treatment assignment for node i by w_i, dropping the time subscript. We denote a policy by $\pi(\Gamma)$, where $\pi(\Gamma) = [w_1, \dots, w_N] \in \{0, 1\}^N$. We assume that there is an upper limit on the number of nodes that can be treated in the final deployment, which we call D, i.e., $\sum_{i \leqslant N} w_i \leqslant D$.

At every period t, the DM's experimentation creates a cost, because the experiment interacts with a previous experimenter's treatment allocation, contaminating their results. We assume that the cost is proportional to the number of units exposed to the treatment, and once a unit is exposed, future exposures of that unit do not increase the cost any further. Define the total cost of experimentation at period t by:

$$c_t = c \sum_{i \leqslant N} \mathbf{1} \left(e_{it} > 0 \text{ and } e_{is} = 0, \forall s < t \right).$$

Throughout, we will let $c = 1$ for simplicity. Hence, the total cost incurred through experimentation by the DM can be written as $C = \sum_{t \leqslant T} c_t$. We assume that the DM has to stay below a 'contamination budget' of at most $\bar{C}$ during experimentation. Then, as a function of the contamination budget $\bar{C}$, we define the value of a policy as follows:

$$V(\pi, \bar{C}) = \sum_{i \leqslant N} \Gamma^T x_i e_i(\pi),$$

where $e_i(\pi)$ denotes the exposure function determined at the last period for final deployment, induced by the treatment allocation under policy π, decided after an experimentation period with the contamination budget $\bar{C}$. The DM does not observe the true vector Γ but has an imperfect estimate of it learned through sequential experimentation, which we will call $\hat{\Gamma}$. The DM maximizes the empirical value function using its estimate at the end of the experimentation period:

$$Max_{\hat{\pi}} : \sum_{i \leqslant N} \hat{\Gamma}^T x_i e_i(\hat{\pi}) \quad \text{s.t. } 1^T \hat{\pi} \leqslant D. \tag{1}$$

Let $X_{K \times N}$ denote the matrix of covariates for all the nodes on the graph, and let X_s denote the analogous matrix only for nodes that were exposed to the treatment during the experimentation period. Let $\hat{\Gamma}_{1 \times K}$ be the vector of estimated coefficients.

Define the vector of exposure values at time t as $\bar{e}_t = [\mathbf{1}(e_{1t} > 0), \ldots, \mathbf{1}(e_{Nt} > 0)]_{1 \times N}$ and let c_t denote the cost vector at time t. We can recursively write: $c_0 = 0_{1 \times N}, c_1 = \bar{e}_1$, and generally, $c_n = (c_{n-1} + \bar{e}_n) - (Diag(c_{n-1})\bar{e}_n)$. $Diag(c_n)$ denotes the diagonal matrix with the entries of c_n on its diagonal entries. This recursive formulation captures the notion that the cost of second and future exposures to the experiment are zero.

We can also define the final exposure vector e in terms of the adjacency matrix of the network, $A_{N \times N}$ (with self-edges), and the final treatment allocation vector $\pi \in \{0, 1\}^N$. Representing e as a vector, we have: $Diag(A\mathbf{1}_N)^{-1} A\pi = e$, where $\mathbf{1}_N$ is an $(N \times 1)$ vector of 1's. Therefore, we can rewrite the maximization problem (1) as a convenient linear program, where we relax π_i and allow it to take values in $[0, 1]$, interpreting the resulting policy as a probabilistic treatment assignment rule:

$$Max_{\hat{\pi}} : \left(\hat{\Gamma}^T X Diag(A\mathbf{1}_N)^{-1} A \right) \hat{\pi}, \textit{ subject to: } 1^T \hat{\pi} \leqslant D, \forall i, \hat{\pi}_i \in [0, 1] \tag{2}$$

3 Regret Analysis

Let $\hat{\pi}$ denote the optimal policy maximizing the optimization problem (2) given that the DM has estimated the coefficient vector as $\hat{\Gamma}$, which achieves the value $V(\hat{\pi}, \bar{C})$. Let $V(\pi^*)$ be the optimal value achievable by the oracle who solves the optimization problem (2) with the true coefficient vector Γ. Thus, we can define the regret as follows:

$$R = V(\pi^*) - V(\hat{\pi}, \bar{C}).$$

To analyze the regret of the empirical value maximizing policy $\hat{\pi}$, let us introduce some notation. Let $\sigma_j(A)$ denote the j^{th} eigenvalue of the matrix A, with σ_1 being the largest. Suppose that the covariate matrices X, X_s are standardized, so the matrix XX^T is the symmetric correlation matrix.

Theorem 1 *Fix $\tau \geqslant 0$. Then, regret is bounded above by K with the following probability:*

$$\mathbb{P}(R \leqslant K) \geqslant 1 - \exp\left(-\frac{1}{2}(\tau-1)^2\right), \tag{3}$$

where

$$K = \frac{\delta \max\{\sqrt{n+D^2}, ||\Gamma||\sigma_1(XX^T)\}}{\sqrt{n+1}} \cdot \frac{\sqrt{n+D^2}}{\sqrt{n+1}-\delta}, \tag{4}$$

and $\delta = \tau\sigma_\epsilon \sum_{j=1}^{k} \sigma_j((X_sX_s^T)^{-1})$.

The proof is based on a perturbation analysis of the LP in (2) with respect to the objective function coefficients.

3.1 Proposed Algorithm and Simulations

We propose an algorithm based on information-directed sampling and the literature on linear bandits, which adaptively chooses experimental subjects that will maximize information gain. Since our setting has an additional component which is the contamination cost due to externalities on other experiments, we use a knapsack-based heuristic $c - \mu$ rule. The full procedure that determines the path of experimentation is given in Algorithm 1.

In simple terms, our algorithm chooses the most valuable nodes to experiment on greedily at every period, taking into account the cost of contaminating other nodes. We measure the information gain from experimenting on a given node by the change in the trace of the matrix $(X_sX_s^T)^{-1}$ after adding all the unexposed neighbors of the target node to the set of exposed nodes. We calculate the cost of experimenting on a node by calculating the number of new exposures resulting from treating the target node. We get the value of experimenting on a node by taking the ratio of the information gain to the cost, and determine the path of experimentation by sequentially choosing the nodes with the highest value until we reach our contamination budget, $\bar{C}$. Using this algorithm to determine which nodes to experiment on every period, the DM updates their estimate of the coefficient vector $\hat{\Gamma}$ after every experiment, solving the maximization problem in (2) in the last period to decide on a final treatment regime that will be deployed.

We run simulation experiments to compare the performance of our proposed algorithm against the policy chosen by an oracle who knows the true response function. For each simulation, we draw an Erdos-Renyi random graph with 100 nodes and different connectivity parameters $p \in \{0.025, 0.05, 0.1\}$, and create our random covariate matrix $X_{10\times100}$ using draws from independent standard normal random variables, and generate our true coefficient vector Γ drawing from a 10-dimensional normal distribution with mean zero and covariance matrix I_{10}. Our results show that the proposed algorithm performs reasonably well in this simulated setting, with performance improving in sparser graphs. In sparse graphs, since there is less contamination caused by experimenting on a given node, the algorithm can learn better before hitting the contamination cap, hence the average percentage gap to the oracle value falls below 5% with a contamination budget $\bar{C}$ only a third of the graph size. In denser graphs, approaching the oracle value while limiting contamination becomes more difficult, and given a $\bar{C}$ half of the graph size, the average percentage gap to the oracle value is around 20% on graphs with $p = 0.05$ and 80% for graphs with $p = 0.1$. However, this problem is much less severe in large graphs, as can be seen from our simulations on random graphs with N=1000, $p = 0.1$, and a 10-dimensional covariate vector Γ. In this setting, the average gap to oracle value is below 5%, even with a $\bar{C}$ value one-eighth the graph size.

The learning problem of the DM will likely be more difficult if we had homophily in the covariates of our nodes (i.e., the covariate vectors of neighboring nodes are correlated)

Algorithm 1 Contamination-aware sequential experimentation

```
function CALCULATE VALUE(Target (Integer), X (Matrix), Treated Nodes (Set), Exposed Nodes (Set), Current Cost (Integer), $\bar{C}$ (Integer))
    $Cost \leftarrow |Neighbors(Target) \backslash ExposedNodes|$
    if ($Target \in TreatedNodes$) or ($Cost + CurrentCost > \bar{C}$) then
        return 0
    else
        $NewExposedNodes \leftarrow Neighbors(Target) \cup ExposedNodes$
        $X_E \leftarrow X[ExposedNodes]$
        $X_N \leftarrow X[NewExposedNodes]$
        $InformationGain \leftarrow tr((X_E X_E^T)^{-1}) - tr((X_N X_N^T)^{-1})$
        $Value \leftarrow InformationGain/Cost$
        return $Value$
    end if
end function
```

```
function GET SAMPLING PATH(A (Matrix), $\bar{C}$ (Integer), X (Matrix))
    $Path \leftarrow [\ ]$ (Vector)
    $CurentCost \leftarrow 0$
    $TreatedNodes \leftarrow \varnothing$
    $ExposedNodes \leftarrow \varnothing$
    while $CurentCost < \bar{C}$ do
        $BestNode \leftarrow ArgMax_{i \in \{1,\ldots,N\}} CalculateValue(i, X, TreatedNodes, ExposedNodes, CurentCost, \bar{C})$
        $OptimalValue \leftarrow CalculateValue(BestNode, X, TreatedNodes, ExposedNodes, CurentCost, \bar{C})$
        if $OptimalValue > 0$ then
            $CurentCost \leftarrow CurentCost + |Neighbors(BestNode) \backslash ExposedNodes|$
            $TreatedNodes \leftarrow TreatedNodes \cup \{BestNode\}$
            $ExposedNodes \leftarrow ExposedNodes \cup Neighbors(BestNode)$
            $Path \leftarrow Append(Path, BestNode)$
        else
            break
        end if
    end while
    return $Path$
end function
```

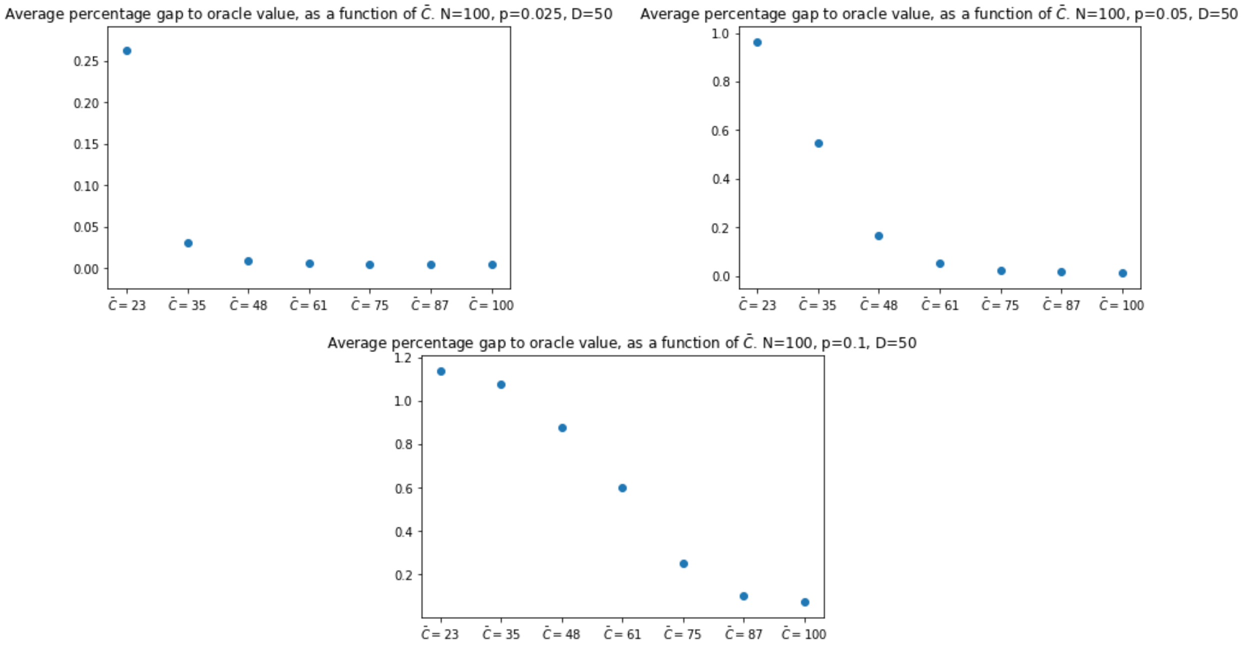

Figure 1: Percentage gap to oracle value ($\frac{V(\pi*)-V(\hat{\pi},\bar{C})}{V(\pi*)}$), for a set of different contamination budgets in simulation studies on Erdos-Renyi random graphs with $p \in \{0.025, 0.05, 0.1\}$, and N=100, with a random 10-dimensional unknown parameter vector Γ. 1000 simulations were run for every parameter combination.

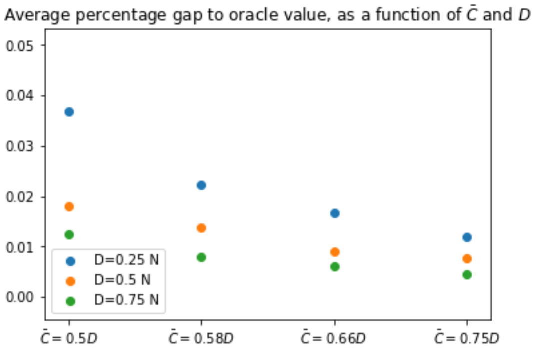

Figure 2: Percentage gap to oracle value ($\frac{V(\pi*)-V(\hat{\pi},\bar{C})}{V(\pi*)}$), for a set of different contamination budgets in simulation studies on Erdos-Renyi random graphs with $p = 0.1$, and N=1000, with a random 10-dimensional unknown parameter vector Γ. 1000 simulations were run for every parameter combination.