Heterogeneous Threshold Estimation for Linear Threshold Modeling

Christopher Tran
ctran29@uic.edu
University of Illinois at Chicago

Elena Zheleva
ezheleva@uic.edu
University of Illinois at Chicago

ABSTRACT

Social networks play a central role in the spread of diseases, ideas, and beliefs. The Linear Threshold Model (LTM) is a prominent model which describes the process of diffusion through the network and how nodes become "infected" based on a threshold of number of neighbors who are already "infected." LTM is often used with the assumption that node thresholds are globally unique or randomly distributed. In many cases, however, thresholds can differ between individuals, and knowing individual-level thresholds can lead to better diffusion predictions. In this work, we propose a causal inference approach for estimating node thresholds. We develop a Structural Causal Model to show the identifiability of causal effects in the Linear Threshold Model, and map the threshold estimation problem to heterogeneous treatment effect estimation. Through experimental results on real-world and synthetic datasets, we show that individualized thresholds play an important part in reliable long-term diffusion prediction.

ACM Reference Format:
Christopher Tran and Elena Zheleva. 2020. Heterogeneous Threshold Estimation for Linear Threshold Modeling. In *MLG '20: International Workshop on Mining and Learning with Graphs, Aug 24, 2020, San Diego, CA, US.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The advent of online social networks allowed millions of people to interact with each other. The spread of information and ideas has been studied as information diffusion in many fields such as epidemiology [16], marketing [15], and the social sciences [14]. The Linear Threshold Model (LTM) is one prominent framework of information diffusion in networks. In the LTM process, an individual is influenced to adopt a product or idea if the proportion of that individual's friends who have already adopted that product or idea is above some threshold. LTM has been used in many settings, such as modeling the spread of ideas [29] and the development of influence maximization algorithms [7, 15]. However, node thresholds are typically assumed to be globally unique or randomly distributed [15], despite the fact that in reality, different individuals may have different susceptibility to social influence. In fact, a recent survey on LTMs highlighted this gap in the research literature and specified the need to develop individual-level threshold estimation models [27]. Moreover, existing works do not study LTM through a causal inference lens, even though LTM is essentially a causal concept: "how many friends does it take to buy a product before it *causes* me to buy the same product?" In our work, we seek to address these shortcomings of previous work.

We move away from globally set threshold assumptions and estimate individual-level thresholds from data. Our models reflect real-world scenarios in which some individuals are more easily influenced than others, or are influenced by specific friends differently than other friends. To address the variety of characteristics and behaviors of individuals, we frame the threshold estimation problem as a causal problem: "what is the minimum number of activated neighbors that can cause a node with certain attributes to become activated?" We study a Structural Causal Model (SCM) that encodes *interference* by *contagion* [20, 21]. Interference is defined as the influence an individual's social network has on their own outcomes, where contagion is the process of friends' outcomes influencing an individual's own outcomes. For example, in Fig 1, Angelo might be tempted to buy new trendy sunglasses (outcome) if his friends already bought them (contagion). Recent work has studied the role and specification of SCMs in the presence of interference [5, 20, 26], the estimation of social effects [1, 19, 26], and the use of other graphical models for diffusion prediction [10]. To the best of our knowledge, no Structural Causal Model has been developed for the Linear Threshold Model. Through SCM, we show how the contagion can be identified in LTM, while contrasting our work to prior studies on unidentifiability for social effects [19, 26]. We map the problem of individual threshold estimation to trigger-based heterogeneous treatment effect estimation [28], and propose two methods of estimating the threshold, one based on the concept of meta-learners and the other on causal trees. In our experiments, we explore the problem of diffusion prediction using the Linear Threshold Model. Our experimental results on real-world and synthetic datasets show that individualized threshold estimation plays a crucial role in reliable long-term diffusion prediction.

2 PROBLEM SETUP

We present the problem setup for information diffusion, as well as LTM. We then describe the objective of threshold estimation for LTMs as well as our main objective for predicting thresholds.

2.1 Information diffusion

Let $G = (\boldsymbol{V}, \boldsymbol{E})$ denote an attributed, *social network*, where $\boldsymbol{V}$ is the set of nodes and $\boldsymbol{E}$ is the set of edges between nodes. If two nodes $v, u \in \boldsymbol{V}$ are connected by an edge, then $(v, u) \in \boldsymbol{E}$, denotes an edge from v to u. If G is an undirected graph, then $(v, u) \in \mathbf{E} \implies (u, v) \in \mathbf{E}$ and we assume that diffusion can flow in either direction.

MLG '20, Aug 24, 2020, San Diego, CA, US

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

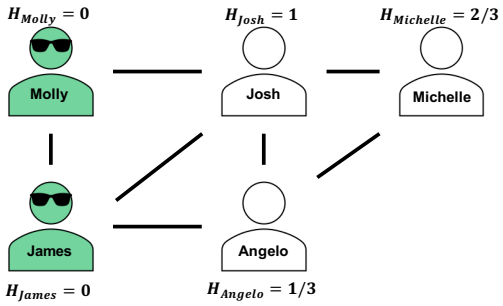


Figure 1: A toy example illustrating the Linear Threshold diffusion process, where Molly and James are activated. In the first time step, Angelo will become activated, since his threshold, $H_{Angelo} = 1/3$, but Josh will not become activated since his threshold is 1, which requires all his friends to be activated first.

Additionally, define the set of neighbors of v to be $N(v) = \{u : u \in \boldsymbol{V}, (v,u) \in \mathbf{E}\}$. Each node $v \in \boldsymbol{V}$ has an m-dimensional vector of attributes, $\mathbf{X_v}$, and outcome of interest $Y_v \in \{0,1\}$, which is a binary indicator of whether the node is activated (e.g. whether an individual has bought new sunglasses). We define the set of all activated nodes at time t to be $\mathcal{D}_t = \{v : Y_v = 1\}$.

2.2 Linear Threshold Model

According to LTM, each node v has a threshold of activation H_v. Given an initial set of activated nodes, $\mathcal{D}_0 \subseteq \mathbf{V}$, diffusion occurs in discrete steps, $t = 1, 2, \ldots, T$. In each time step t, a node $v \in \boldsymbol{V} \setminus \cup_{i=0}^{t} \mathcal{D}_i$ is activated if the *activation influence*, the weighted proportion of its activated neighbors reaches its threshold H_v[1]:

$$\sum_{u \in N(v)} w_{uv} Y_u \geq H_v, \tag{1}$$

where w_{uv} is the normalized influence strength of neighbor u to v. We assume $w_{uv} \in [0,1]$ for $(u,v) \in \boldsymbol{E}$, so that nodes only become activated as the number of activated neighbors increases. Influence weights can be calculated in a variety of ways such as: degree centrality [15], PageRank [15], credit distribution [13], or edge centrality [9].

Figure 1 shows a toy example of a social network with 5 individuals, assuming equal weights. Each node has their own threshold (e.g. $H_v = 1/3$ means v's threshold is 1/3.) The initial set of activations is the set of individuals who have adopted a new product (new sunglasses), which consists of two individuals: $\mathcal{D}_0$ = {Molly, James}. In the first time step, Angelo will buy new sunglasses, since his threshold, $H_{Angelo} = 1/3$, and 1 of his 3 friends (James) has already bought new sunglasses. The set of activations at the first time step is, $\mathcal{D}_1$ = {Molly, James, Angelo}. No one else will buy sunglasses, since Josh's threshold is 1, which requires all his friends to buy sunglasses, and Michelle's threshold is 2/3, which means all her friends need to buy sunglasses, since she only has 2 friends.

[1]We consider the threshold and activation influence to be a weighted proportion. Our model can be applied in both settings.

2.3 Problem statement

Our goal in this work is to estimate individual-level thresholds for all nodes in the network:

Problem 1. *(Heterogeneous threshold estimation for LTM) Given a graph, $G = (\boldsymbol{V}, \boldsymbol{E})$ and a set of individual-thresholds $\boldsymbol{H} = \{H_v | v \in \boldsymbol{V}\}$. The goal is to estimate thresholds, $\hat{\boldsymbol{H}} = \{\hat{H}_v | v \in \boldsymbol{V}\}$, such that the average error is minimized:*

$$\min_{\hat{H}_v} \frac{1}{|\boldsymbol{V}|} \sum_{\boldsymbol{V}} |H_v - \hat{H}_v|. \tag{2}$$

An important application of estimating individual-level thresholds is to predict diffusion [6, 13, 25]. Another application is targeting individuals who are susceptible to neighborhood influence, since individuals with lower thresholds require less activated neighbors.

3 CAUSAL MODEL FOR LTM

First, we provide a brief background to causal inference, particularly the problem of estimating heterogeneous treatment effects. Then, we present a causal model assumption for LTM.

3.1 Causal inference background

For any node v, denote the treatment group assignment as $Z_v \in \{0,1\}$, where $Z_v = 1$ and $Z_v = 0$ indicates that v is in the treatment or control group, respectively. The observed outcome of an individual given a treatment value of z is $Y_v(z)$. If we have both observed outcomes, then we can compute the individual effect as: $Y_v(1) - Y_v(0)$. In practice, we cannot observe both outcomes simultaneously, so the actual observed outcome is:

$$Y_v = \begin{cases} Y_v(0), & \text{if } Z_v = 0, \\ Y_v(1), & \text{if } Z_v = 1, \end{cases} \tag{3}$$

Often times the average treatment effect is of interest, which is the mean of differences in treated and control groups: $E[Y(1) - Y(0)]$.

3.1.1 Heterogeneous treatment effect estimation. In many cases, effects of treatment are different for each individual, known as heterogeneous (or individual) treatment effect expressed through the *conditional average treatment effect (CATE)* [2, 22]. CATE is defined as the expectation of difference in outcome with respect to an individual's features:

$$\tau(\mathbf{x}) \equiv E[Y(1) - Y(0) \mid \mathbf{X} = \mathbf{x}], \tag{4}$$

Since this is the estimated effect with respect to relevant features, we get *individualized* effects, instead of average effects overall.

3.2 Causal model for LTM

The first step in estimating causal effects in networked data is reasoning about causal effect identification using Structural Causal Models (SCMs) [21]. SCM is a graphical model that encodes cause-effect relationships between variables. SCMs typically assume that data is independent and identically distributed (IID), and do not encode interference between individuals. Interference occurs when one individual's outcome may depend on other individuals. Identification in the presence of interference has been studied [1, 5, 20, 26], but not in the context of LTM. Ogburn and VanderWeele presented an extensive discussion of SCMs for interference [20], exploring

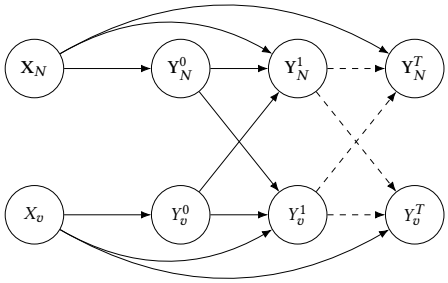


Figure 2: A causal model of contagion for a node v that reflects the LTM process.

three different types of interference for fixed group sizes: direct, contagion, and allocational. In the context of LTM, diffusion is a problem of *contagion*. Contagion is present when an individual's individual outcome depends on the outcome of other individuals, which is analogous to LTM activation. We present our SCM that encodes contagion for an arbitrary size neighborhood.

Figure 2 shows the causal model of diffusion we develop to capture the LTM process. Here, $\mathbf{X}_N$ represents a *set* of features, and $\mathbf{Y}_N^t$ is the set of indicators of activations for the neighbors, N, of node v. An arrow from one variable to another denotes a cause-effect relationship (e.g. a node's features affect its own activation). In this model, the contagion from node v's neighbors are captured through the arrows from $\mathbf{Y}_N^t$ to Y_v^{t+1}. While group effects have been shown to be unidentifiable without strong assumptions [19], Shalizi and Thomas explored the identification of effects after decoupling and considering pairwise interactions [26]. In contrast to the models explored by Shalizi and Thomas, our models consider an arbitrary neighborhood size for the contagion effects.

One issue to consider is that the neighborhood features and outcomes are of varying sizes, so the estimation of effects cannot be estimated directly using standard regression with a fixed number of variables. We follow the approach used by Arbour et al. for summarizing relational features using aggregation [1]. They use sufficient statistics to model the distribution of relational variables for estimating direct interference. We use the weighted mean of activated neighbors as the measurement for activation influence.

In order to *identify* the effect of activation influence (left side of eq (1)) on a node's own activation (i.e. the effect of $\mathbf{Y}_N^t$ on Y_v^{t+1}), we need to find the correct *adjustment set* through the *back-door criterion* [21]. To identify the contagion at an arbitrary time step t, the effect of $\mathbf{Y}_N^{t-1}$ on Y_v^t, we find a set of variables Z such that "no node in Z is a descendant of $\mathbf{Y}_N^1$ and Z blocks every path between $\mathbf{Y}_N^1$ and Y_v^2 that contains an arrow into $\mathbf{Y}_N^1$". Here, a blocked path follows the same criteria as in a traditional directed acyclic graph (DAG). In our example, there are three arrows going into $\mathbf{Y}_N^{t-1}$: Y_v^{t-2}, $\mathbf{Y}_N^{t-2}$ and $\mathbf{X}_N$. Based on this, one sufficient set to block all back-door paths is to condition on the node features X_v, and the previous outcome Y_v^t. This is because the previous outcome blocks all back-door paths from any Y_N and X_v blocks all back-door paths of Y_v^{t-1}. Therefore, to identify the contagion effect for any node $v \in \boldsymbol{V}$, we only need to take into account v's features and its previous outcome. In the CATE formula defined in eq. (4), this is the heterogeneous subgroup $X_v = \mathbf{x}$, The special case is the initial contagion effect of $\mathbf{Y}_N^0$ on Y_v^1, where there are no back-door paths. In this case, we can estimate the effect of $\mathbf{Y}_N^0$ without an adjustment set.

To match the characteristics of LTM, we define a functional form of the outcome of node v. The outcome of node v at time t can be represented as a function of $\left(X_v, Y_v^{t-1}, \mathbf{Y}_n\right)$, which maps to the threshold indicator, ϕ, analogous to eq. 1:

$$\phi(X_v, \mathbf{Y}_N) = \mathbb{1}\left[\sum_{u \in N} w_{uv} * Y_u \geq h(X_v)\right]. \tag{5}$$

Here, we define a threshold function on the node features as $H_v = h(X_v)$, which captures the notion of individual-level thresholds. Finally, we can define the functional form of Y_v^t for $t \geq 1$ as:

$$Y_v^t(X_v, Y_v^{t-1}, \mathbf{Y}_N) = \begin{cases} Y_v^{t-1} & \text{if } Y_v^{t-1} = 1, \\ \phi(X_v, \mathbf{Y}_N) & \text{if } Y_v^{t-1} = 0. \end{cases} \tag{6}$$

This form correctly captures the LTM process: a node stays activated if already activated, otherwise, it activates based on its individual threshold and the activations and influences of neighbors.

4 THRESHOLD ESTIMATION FOR LTM

Now, we map the problem of estimating thresholds to the problem of estimating triggers for heterogeneous effects. Then, we propose two methods for estimating thresholds for LTM.

4.1 HTE triggers for threshold estimation

The treatment in our problem, Z_v, is the activation influence of neighbors (i.e. $\sum w_{uv} * Y_u$), which is not a binary value, but rather a continuous value. In LTM, the continuous treatment value of activation influence turns into a binary treatment through node thresholds: a node is "treated" if its activation influence is above its threshold, and "untreated" (or control) if it is below. Then our goal is transformed into a problem of estimating the correct node threshold that correctly identifies when a node is treated. To do this, we map the threshold estimation problem to the problem of estimating triggers for heterogeneous treatment effects [28].

A *trigger* is defined as the minimum amount of treatment necessary to change an outcome. Some examples of triggers are: (1) the minimum number of days a patient needs to take a medicine to be cured; (2) a minimum discount needed for a customer to buy a product. For the problem of threshold estimation, our causal question is "what is the minimum number of activated neighbors that can cause a node with certain attributes to become activated"? A trigger θ can have two potential outcomes, dependent on whether Z is above and below the trigger: $Y(Z \geq \theta)$ and $Y(Z < \theta)$. Then, the average treatment effect with trigger θ is $E[Y(Z \geq \theta) - Y(Z < \theta)]$. When the threshold is the same for everyone in a population, the trigger is defined at the population level.

In order to estimate individual-level thresholds, we study *heterogeneous (individual) triggers* estimation. Let the individual trigger of node v be θ_v with potential outcomes: $Y_v(Z_v \geq \theta_v)$ and $Y_v(Z_v < \theta_v)$. We can define CATE with a trigger as:

$$\tau(\mathbf{x}) = E[Y(Z_v \geq \theta_v) - Y(Z_v < \theta_v) | \mathbf{X} = \mathbf{x}]. \tag{7}$$

We see that estimating CATE with a trigger translates to estimating the "effect" of the activation influence being above the trigger. The trigger $\theta_v = \hat{h}(X_v)$ is an estimation of the node threshold $h(X_v)$.

We demonstrate through an example why the trigger that maximizes the effect corresponds to finding the correct threshold for a node v. Let $|N(v)| = 8$ and the true threshold be $h(X_v) = 0.5$ for a node v. Assume all neighbor influences are equal ($w_{uv} = 1/8$). Define $A_v^0(z)$ and $A_v^1(z)$ to be the set of potential outcomes when there are less than and greater than z neighbors activated, respectively:

$$A_v^0(z) = \{Y_v(a) \mid 0 \le a < z\},$$
$$A_v^1(z) = \{Y_v(a) \mid z \le a \le 8\}, \tag{8}$$

where $Y_v(a)$ is the outcome with a activated neighbors. For simplicity, we assume that each potential outcome is equally likely. Suppose a model estimates $\hat{h}(X_v) = \theta_v = 1/8$. To find the causal effect of the estimated threshold, we evaluate the expected outcomes above and below the trigger[2]:

$$E[Y_v(Z_v < \theta_v)] = \frac{1}{N_0} \sum_{a=0}^{N_0-1} Y_v(a) = \frac{1}{1}(0 * 1) = 0, \tag{9}$$

$$E[Y_v(Z_v \ge \theta_v)] = \frac{1}{N_1} \sum_{a=N_0}^{N_1} Y_v(a) = \frac{1}{8}(0 * 3 + 1 * 5) = \frac{5}{8}, \tag{10}$$

where $N_0 = |A_v^0(1)| = 1$ and $N_1 = |A_v^1(1)| = 8$. The first quantity is 0 since there is only 1 element in $A_v^0(1)$, namely, $Y_v(0)$, which is 0 since no neighbors are activated. The second set, $A_v^1(1)$, has 3 cases where the node does not activate (1, 2, 3 neighbors activated) and 5 cases where the node does activate (4, 5, 6, 7, 8 neighbors activated). We find the expected outcome is 5/8, and the causal effect is:

$$E[Y_v(Z_v \ge \theta_v)] - E[Y_v(Z_v < \theta_v)] = 5/8 - 0 = 5/8. \tag{11}$$

On the other extreme, imagine a model estimates $\theta_v = 1.0$. The expected outcomes and treatment effect in this case are:

$$E[Y_v(Z_v < \theta_v)] = 4/8, \tag{12}$$
$$E[Y_v(Z_v \ge \theta_v)] = 1, \tag{13}$$
$$E[Y_v(Z_v \ge \theta_V)] - E[Y_V(Z_v < \theta_v)] = 1 - 4/8 = 4/8. \tag{14}$$

Finally, suppose a model estimates the true threshold correctly, $\theta_v = h(X_v) = 0.5$. Then, the expected outcomes are:

$$E[Y_v(Z_v < \theta_v)] = 0, \tag{15}$$
$$E[Y_v(Z_v \ge \theta_v)] = 1, \tag{16}$$

since there will be no cases where node v activates below the trigger and will always activate above the trigger. This results in a treatment effect of 1, the maximum value of CATE for LTM. Therefore, we conclude that estimating the *trigger* that *maximizes* the treatment effect will estimate the node *threshold* that changes the output.

Formally, we define the problem as: find the *triggers*, θ_v, that maximizes CATE τ with a minimum trigger θ:

$$\arg\max_{\theta} \ \tau(\mathbf{x}) = E[Y_v(Z_v \ge \theta_v) - Y_v(Z_v < \theta_v) \mid \mathbf{X} = \mathbf{x}]. \tag{17}$$

For each node, we find a the minimal trigger that maximizes CATE, which results in an estimated individual threshold.

[2]We omit the conditional $\mathbf{X} = \mathbf{x}$ since we only consider node v.

4.2 Heterogeneous trigger estimation

We have discussed how to map the threshold estimation problem to a problem of estimating triggers for heterogeneous treatment effects. Here, we describe two methods for estimating triggers: Trigger-based causal trees (TCT) [28] and ST-Learner. We first describe how to adapt trigger-based causal trees [28] to the problem of LTM trigger estimation. Then, we propose a novel meta-learner, ST-Learner, which can use any base learner for estimating effects.

4.2.1 Causal Trees. Causal trees work similarly to decision trees, in that they greedily split using a partition function. The main difference is that the goal of a causal tree is to estimate CATE for different populations of individuals, rather than to predict a label. The causal trees we use work by finding splits that maximize the differences in CATE as a splting criteria, so that the most heterogeneous effects are found at each split [2]. We use the CTL method by Tran and Zheleva, which use an additional validation set while building the tree to penalize effect estimations that do not generalize estimated effects [28]. In order to learn triggers, an additional search is done at each split to find the trigger that maximizes the effect estimation in the split. Through recursive partitioning, CTL balances heterogeneous discovery of effects with generalization effects on separate validation data.

Using the causal tree, we can extract subpopulations via the leaves of the tree. Each leaf represents a subpopulation based on the attribute path to that leaf node. In each leaf, a trigger that maximizes the difference in effects is given. For example, some leaves may have only a trigger of 0.1, which means nodes that end up in that leaf only need a small amount of influence to trigger a change. Other leaves may have a high trigger (e.g. 0.7) which means that a large majority of their neighborhood needs to be activated, to influence a change. Using these leaves, we can assign an *individualized* threshold for every node in the graph.

4.2.2 ST-Learner. Estimating heterogeneous effects using base or meta learners has been described in detail by Künzel et al [17]. They describe three types of learners: S-, T-, and X- learners. However, none of them consider the problem of trigger-based HTE and are not directly suitable for solving the problem of threshold estimation. Here, we describe a variant of the S-Learner to solve the trigger-based HTE estimation problem, which we call the ST-Learner[3].

The S-Learner is based on a *single* base learner for estimating outcomes from data. A base learner can be any regression or classification method, such as Linear Regression or Random Forest. The base learner is used to predict a unit's outcome given the features and treatment indicator: $E[Y_v|X_v, Z_v]$. Here, the treatment indicator is given no special meaning, so the base learner can choose to not use it in the prediction (e.g. a weight of 0). To estimate CATE, the base learner can be used to estimate the expected outcomes when Z_v is 1 and 0: $E[Y_v(1) - Y_v(0)|X_v] = E[Y_v|X_v, Z_v = 1] - E[Y_v|X_v, Z_v = 0]$, which is the difference in predictions when varying Z_v.

To create the ST-Learner, we need to learn the trigger that maximizes the effect. We first use a single base learner to learn the outcomes given all features concatenated with the treatment variable: $E[Y_v|X_v, Z_v]$. Let $\mathbf{Z} = \{z_1, z_2, \ldots, z_n\}$ be all the possible treatment

[3]The "S" refers to the *single* learner and "T" is for *triggers*, following the same naming scheme.

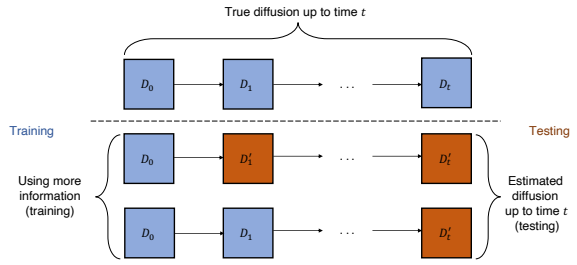


Figure 3: Representation of our experimental setup. Given the true number of activations up to time t (first row), we train on different sets. $\mathcal{D}_i$ represents the set of all activated nodes up to time i and D'_i represents the predicted activations. The second row is using only the initial set of activations, $\mathcal{D}_0$. The third row uses activations up to time 1, $\mathcal{D}_1$.

values of Z_v in the data. Define $\Theta = \{r_1, r_2, \ldots, r_m\}, m \leq n$, to be the subset of triggers we are considering. We use r_i to refer to any potential trigger, while θ_v is a *node's* individual trigger. We can consider all potential values of treatment in the training data, or we can consider a discretization of these values ($m \leq n$).

Using the trained base learner, we can now estimate CATE with different triggers. For any node in our data, we can estimate CATE at any treatment level by imputing the treatment values in $\mathbf{Z}$, so that we have n predictions for any node. Since we have all predictions, we can compute the expectation above and below each trigger. Let $\Theta_i^1 = \{z_k : z_k \geq r_i\}$ and $\Theta_i^0 = \{z_j : z_j < r_i\}$ be the set of treatment values above and below the trigger r_i, respectively. Mapping the right hand side of eq (7) for an arbitrary trigger r_i, we get the analogous objective to the ST-Learner:

$$\arg\max_{r_i} \ \tau(\mathbf{x}) = \frac{1}{|\Theta_i^1|} \sum_{z \geq r_i} E[Y|\mathbf{X} = \mathbf{x}, Z = z] - \frac{1}{|\Theta_i^0|} \sum_{z < r_i} E[Y|\mathbf{X} = \mathbf{x}, Z = z]. \quad (18)$$

We use the base learner to estimate the outcomes for every node above and below the trigger, by taking the average outcomes.

5 EXPERIMENTS

We study three real-world datasets that are natural cases of influence diffusion. We also experiment on four synthetic network generation models. Our models are compared to methods proposed by Talukder et al. [27] and a practical regression baseline.

5.1 Experimental setup

We use two baseline threshold estimation methods by Talukder et al.: *Heuristic Expected* and *Heuristic Individual* [27]. *Heuristic Expected* computes a unique threshold for all the nodes in the network. *Heuristic Individual* estimates a range of values to sample node thresholds from. These baseline methods utilize degree centrality in their threshold selection methods. Since random thresholds are often used in influence maximization problems [15], we also employ a baseline called *Random*, that assigns individual thresholds uniformly random from 0 to 1. In addition, we use a more practical baseline for threshold estimation using *Linear Regression*. To get labels for Linear Regression, we take all currently activated nodes in the network and estimate their threshold by taking the current neighborhood activations. Then, we fit a Linear Regression model on those nodes and estimate thresholds for all nodes in the network.

We compare the baseline methods to the *Causal Tree* and the *ST-Learner* method. For the *ST-Learner*, we use Linear Regression as the base learner. For neighbor influences, we use degree centrality: $w_{uv} = 1/|N(v)|$, where $|N(v)|$ is the degree of node v. However, our models can be applied with any weight estimation method.

5.2 Datasets

We study three real-world datasets for diffusion estimation. Additionally, we use synthetic datasets to explore how our models performs under different network assumptions. For each dataset, we specify the time period for diffusion.

5.2.1 Hateful Users. The Hateful Users dataset was collected by Ribeiro et al. [24]. In this dataset, the authors collected a sample retweet network from Twitter, with 200 most recent tweets for each user. A sample of users were selected to be annotated as *hateful* or *not hateful*. We employ their methodology of identifying hateful users, and the rest can be predicted based on the history of tweets.

We estimate node triggers for how "hatefulness" spreads through the network, where being activated means you change from *not hateful* to *hateful*. For node attributes, we extract *Empath* categories from user tweets [12], and average across all user tweets. Empath captures a wide variety of topics that are relevant. Some examples are: violence, fear, and warmth. We consider a month to month diffusion: how "hatefulness" diffuses on a month to month basis. We look at two time periods: Jan 2016 to Dec 2016 and Jan 2017 to Oct 2017. In the first time period, there are 18,520 users, with an average clustering coefficient of 0.0521. In the second period, there are 345,693 users with an average clustering coefficient of 0.0071.

5.2.2 Cannabis. The Cannabis dataset is a follower network, for the spread of cannabis tweets. The dataset is originally collected on all users who tweet about the ecigarette Juul. From the users who tweet about Juul, we identify those users who also tweet about cannabis or marijuana. All tweets are collected and Empath categories [12] are used as attributes. We estimate node thresholds for how cannabis related tweets diffuse through the network. We consider the period between Jan 2017 to Dec 2017. There are 2,976,033 users in this dataset. The average clustering coefficient is 0.0369.

5.2.3 Higgs Boson. This dataset is based on the announcement of the Higgs-boson like particle at CERN on July 4, 2012. The dataset was collected by De Domenico et al. between July 1 and July 8 of 2012 and is follower network on Twitter [8]. We focus on the spread of mentions for the Higgs-boson discovery with the announcement on July 4 until July 8. There are no explicit node attributes (e.g. tweet text) like the other two datasets, so we construct features based on the graph and use those for our method. We use degree centrality, both in-degree and out-degree, as well as counts of user and neighborhood tweets. We consider hourly diffusions. The average clustering coefficient in this dataset is 0.1280.

Method	Hateful 2016	Hateful 2017	Cannabis	Higgs	Erdos-Renyi	Pref. Attachment	Small World	Forest Fire
Random	0.2450	0.2262	0.2033	0.6287	0.7242	0.8132	0.8039	0.8005
Heuristic Expected	0.2608	0.2300	0.2433	0.6429	0.6488	0.7551	0.7470	0.7376
Heuristic Individual	0.2617	0.2301	0.2691	0.6436	0.6580	0.7655	0.7566	0.7396
Linear Regression	0.1522	0.2297	0.3674	0.6449	0.9693	0.9459	0.8520	0.8429
Causal Tree	**0.7207**	**0.5793**	**0.7830**	0.9427	0.9470	0.9274	0.8461	0.8267
ST-Learner	0.6877	0.5785	0.7564	**0.9734**	**0.9783**	**0.9611**	**0.8871**	**0.8737**

Table 1: Average Jaccard index across all datasets (higher is better). Our models have the highest accuracy across all datasets.

5.2.4 Synthetic datasets. We generate synthetic datasets using four graph generation models: Erdos-Renyi [11], preferential attachment [3], small world [30], and forest fire [18]. We set the final number of nodes to 1000, and for each node, we randomly generate 100 node attributes from a Gaussian, $N(0, 1)$. To generate thresholds, we use a random linear regression model with 10 attributes using scikit-learn's *make regression* function [23]. The normalized (between 0 and 1) outputs of the linear model are used as thresholds for each node. 50 nodes are randomly set to activated and diffusion events are generated based on LTM for 8 time steps. In Erdos-Renyi, we vary the probability of edge creation p from 0.05 to 0.5. In preferential attachment, we vary the parameter k, the number of new attachments, from 1 to 50. For the small world networks, we fix the probability of rewiring an edge to 0.1, and vary the parameter of k nearest neighbors from 1 to 50. For forest fire networks, we fix the backward probability of an edge to 0.1 and vary the forward probability of an edge f from 0.05 to 0.5. For each network generation model, we run 10 simulations and report the average accuracy.

5.3 Evaluation

For evaluation, we simulate diffusion using estimated node thresholds from each model. The goal for evaluation is to determine if our node thresholds are capable of capturing the true number of activations. The better the threshold estimation, the closer the estimation of activated nodes at any given time point, which we call *reach*.

To compare results across different methods, we start with a given *snapshot* of the network (e.g. Jan 2016), and estimate the node thresholds at that snapshot. Here, a snapshot is the current structure and activations of the network at time t. The proportion of activated neighbors for any activated node is considered the threshold. For unactivated nodes, we do not consider them for any of the learning models (Linear Regression, Causal Tree, ST-Learner), while they are considered for the Heuristic baselines. Using those thresholds, we simulate a diffusion process until a target time (e.g. Dec 2016). We show results at different starting time points, since we may have more information with different snapshots. For evaluation, we show the real and simulated number of activated nodes based on estimated node thresholds. Figure 3 shows our experimental setup.

In addition to the comparisons on the *global reach* of each method, we also estimate the error of the specific nodes which were predicted as activated nodes using Jaccard index. Let $\mathcal{D}_t$ be the set of activated nodes and $\mathcal{D}'_t$ be the set of predicted activated nodes up to time t. We compute the average Jaccard index over all sets as:

$$J = \frac{1}{T}\sum_{t=0}^{T}\frac{|\mathcal{D}_t \cap \mathcal{D}'_t|}{|\mathcal{D}_t \cup \mathcal{D}'_t|}. \tag{19}$$

We will also reference the average Jaccard index as the accuracy of prediction. While the *reach* estimations compares the number of activated nodes, the average Jaccard index shows how accurate each model is in predicting specific activated nodes.

5.4 Real-world data results

We present the results on the real-world datasets. In each dataset, we show the true and simulated reach curves. Overall, our models are able to get better reach estimates, and predict a curve closer to the true reach, compared to baseline methods. Our models are able to achieve more accurate activation predictions, shown in Table 1.

5.4.1 Hateful Users Dataset. Figures 4a-4d show the diffusion predictions for the Hateful Users dataset from Jan 2016 to Dec 2016, with varying starting points (Jan, Mar, May, Jul). We show diffusion simulations at four snapshots before Dec 2016. The first thing we notice is that our models initially overestimates the reach the diffusion prediction, but is able to predict close to the final diffusion amount. Additionally, we see that as we get more time steps, our models obtains more accurate reach estimates. For example, in Figures 4a and 4b, where we start with information in January and March, our models overestimates the prediction in the beginning. With more information, the reach estimates are better, as in Figures 4c (starting in May), and 4d (starting in July). In the 2017 dataset, we notice the same trends as in the 2016 dataset: our models overestimates the reach prediction in the beginning, but predicts close to the final true amount, so we omit the plots.

5.4.2 Cannabis Dataset. Figures 4e-4h show results for threshold estimation on the Cannabis dataset from Jan 2017 to Dec 2017. Contrast to the Hateful Users dataset, all models predict diffusion that saturate after a number of time steps. A reason for this could be the sparsity of edges. Additionally, the network is a follower network, so different parts of the network do not necessarily interact with each other, which results in disjoint subgraphs. About 77% of users in this follower network only have 1 follower. The diffusion may get "blocked" when there is no connection to different parts of the graph, so the simulated diffusion will not progress throughout the network. Another reason is the possibility of exogenous variables that affect how users tweet about cannabis, and how many users tweet about cannabis. New users may join Twitter, or laws may be passed that increase the activity of Cannabis related hashtags. The methods presented are not able to account for exogenous variables.

5.4.3 Higgs Dataset. For the Higgs dataset, we show diffusion predictions from the beginning of July 4 until the end of July 8. Figures 4i-4l shows the diffusion process for four different starting time steps. Specifically, we start at noon, July 4th and increase the starting time steps by 12 hours each time. Here, we see that the

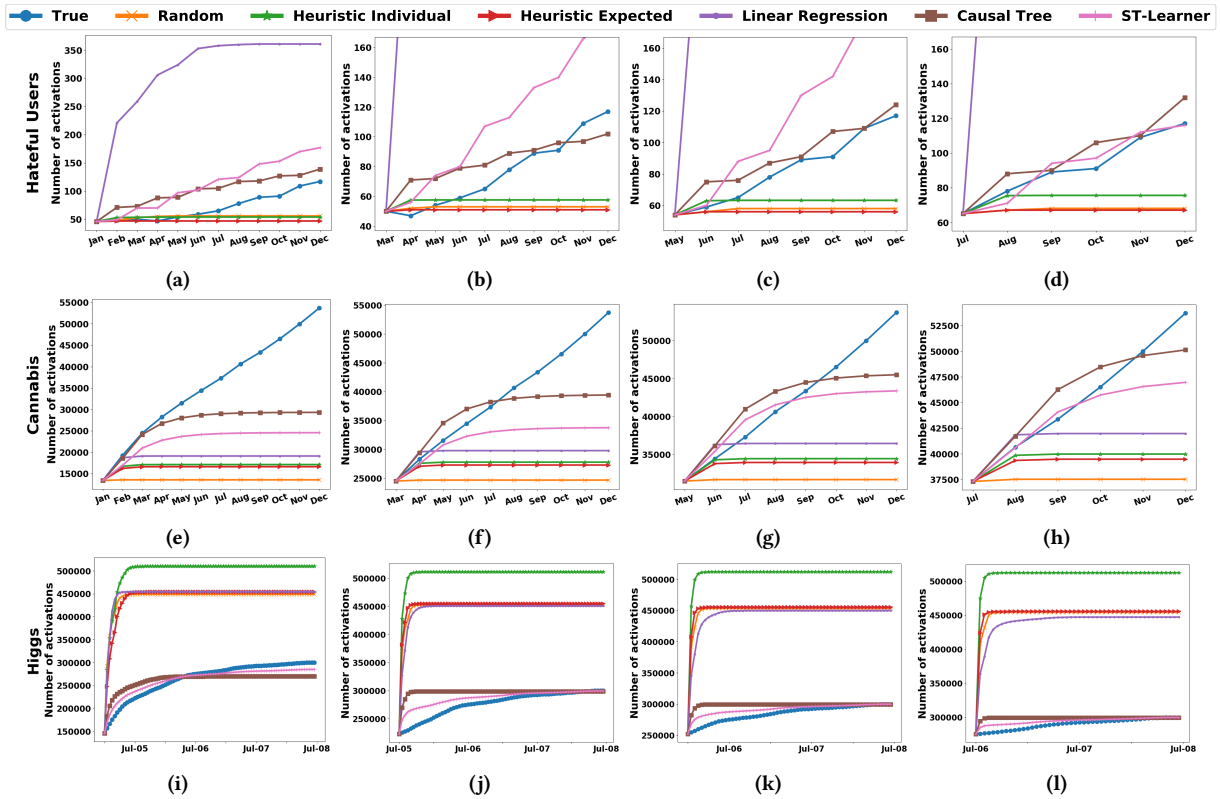


Figure 4: Comparison of diffusion size prediction on three real world datasets. Our models have the closest estimation of reach over longer time periods whereas the baselines incorrectly predict diffusion saturation in the early stages.

Method	Hateful Users	Cannabis	Higgs
Heuristic Expected	0.9523	0.012	0.5840
Heuristic Individual	[0.5, 1.0]	[0, 0.1]	[0.3. 0.9]
Linear Regression	0.0063	0.0369	0.5298
Causal Tree	0.0202	0.0020	0.7127
ST-Learner	0.0486	0.0051	0.5435

Table 2: Average estimated thresholds for all models.

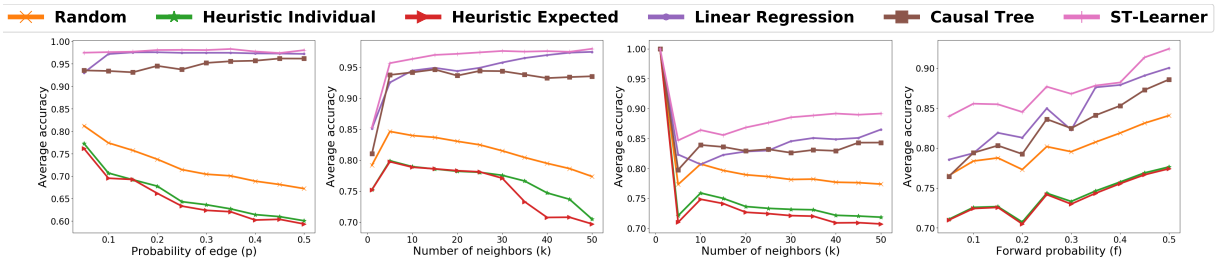
ST-Learner performs better than the causal trees. Additionally, the baseline models significantly overestimate the reach predictions across all snapshots of the network.

5.4.4 Discussion. Interestingly, the baseline methods predict diffusion saturation in all real-world datasets, but may overestimate (Higgs dataset) or underestimate reach (Hateful Users and Cannabis datasets). To explain this behavior, we look at the thresholds produced by each model, shown in Table 2. In Hateful Users, our models produce significantly lower thresholds compared to the Heuristic baseline methods (0.0202 and 0.0486 v.s. [0.5, 1.0] and 0.9523). From the results in Figures 4a-4d, we saw the baseline methods diffusion predictions plateau early on, which suggests the thresholds estimated were too high and resulted in saturation. This also occurs in the Cannabis dataset, where our model's thresholds are significantly lower than the baseline methods.

Compared to our models, Linear Regression estimates larger thresholds in the Hateful Users, which results in overestimating reach. In the Higgs dataset, our thresholds are on average higher (0.7127 and 0.5435 v.s. [0.3, 0.9] and 0.5840), which explains why the baselines significantly overestimate true reach estimations (e.g. Figure 4j). From these results, we show that having individualized and accurate thresholds are important for reliable diffusion prediction.

5.5 Synthetic data results

We explore how the network structure affects a model's ability to estimate thresholds correctly. Here, we show the average Jaccard index of all models run over 10 simulations. Figure 5 shows the average Jaccard index of all models across variations of parameters for each network structure. We can see that the Linear Regression based models perform the best, with the ST-Learner performing the best overall. The most likely reason is that those threshold estimation models are closest to the underlying process of threshold generation, which was based on a linear regression. Linear Regression can exploit the imprecise thresholds to learn the correct thresholds and predict thresholds more accurately. The Causal Tree performs worse out of the three learning methods, which may also be due to the linear model used for threshold generation.



(a) Erdos-Renyi (b) Pref. attachment (c) Small world (d) Forest fire

Figure 5: Average Jaccard index across variations for several network structures. All models' vary in accuracy across all networks depending on the parameters.

6 CONCLUSION

In this work, we proposed a causal inference approach to individual-level threshold estimations in the Linear Threshold Model. We have shown that under our causal model assumption, contagion can be identified, and the heterogeneous treatment effects can be estimated in the social network. We adapt trigger-based causal trees and develop a new meta-learner, the ST-Learner, for estimating triggers for heterogeneous effects to solve the threshold estimation problem. Our results show that learning individualized thresholds from data can provide better and more reliable estimates of diffusion in networks, compared to state-of-the-art methods which ignore individual variations. A fruitful avenue of research would be to develop models that combine edge diffusion probability estimation with threshold estimation, as well as explore learning from data over time to combat upwardly biased thresholds and diffusion predictions, known as the opacity problem [4].

REFERENCES

[1] David Arbour, Dan Garant, and David Jensen. 2016. Inferring network effects from observational data. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 715–724.

[2] Susan Athey and Guido Imbens. 2016. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences* 113, 27 (2016), 7353–7360.

[3] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[4] George Berry, Christopher J Cameron, Patrick Park, and Michael Macy. 2019. The opacity problem in social contagion. *Social Networks* 56 (2019), 93–101.

[5] Rohit Bhattacharya, Daniel Malinsky, and Ilya Shpitser. 2019. Causal Inference Under Interference And Network Uncertainty. In *Uncertainty in artificial intelligence: proceedings of the... conference. Conference on Uncertainty in Artificial Intelligence*, Vol. 2019. NIH Public Access.

[6] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. 2016. Representation Learning for Information Diffusion Through Social Networks: An Embedded Cascade Model. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (San Francisco, California, USA) *(WSDM '16)*. ACM, New York, NY, USA, 573–582. https://doi.org/10.1145/2835776.2835817

[7] Tianyu Cao, Xindong Wu, Tony Xiaohua Hu, and Song Wang. 2011. Active learning of model parameters for influence maximization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 280–295.

[8] Manlio De Domenico, Antonio Lima, Paul Mougel, and Mirco Musolesi. 2013. The anatomy of a scientific rumor. *Scientific reports* 3 (2013), 2980.

[9] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Angela Ricciardello. 2012. A novel measure of edge centrality in social networks. *Knowledge-based systems* 30 (2012), 136–150.

[10] Quang Duong, Michael P Wellman, and Satinder Singh. 2011. Modeling information diffusion in networks with unobserved links. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE, 362–369.

[11] Paul Erdős and Alfréd Rényi. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5, 1 (1960), 17–60.

[12] Ethan Fast, Binbin Chen, and Michael S Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 4647–4657.

[13] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 241–250.

[14] Mark Granovetter. 1978. Threshold models of collective behavior. *American journal of sociology* 83, 6 (1978), 1420–1443.

[15] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 137–146.

[16] Abdelmajid Khelil, Christian Becker, Jing Tian, and Kurt Rothermel. 2002. An epidemic model for information diffusion in MANETs. In *Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*. 54–60.

[17] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. 2019. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences* 116, 10 (2019), 4156–4165.

[18] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 177–187.

[19] Charles F Manski. 1993. Identification of endogenous social effects: The reflection problem. *The review of economic studies* 60, 3 (1993), 531–542.

[20] Elizabeth L Ogburn, Tyler J VanderWeele, et al. 2014. Causal diagrams for interference. *Statistical science* 29, 4 (2014), 559–578.

[21] Judea Pearl. 2009. *Causality*. Cambridge university press.

[22] Judea Pearl. 2017. Detecting latent heterogeneity. *Sociological Methods & Research* 46, 3 (2017), 370–389.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[24] Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. 2018. Characterizing and detecting hateful users on twitter. In *Twelfth International AAAI Conference on Web and Social Media*.

[25] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. 2008. Prediction of information diffusion probabilities for independent cascade model. In *International conference on knowledge-based and intelligent information and engineering systems*. Springer, 67–75.

[26] Cosma Rohilla Shalizi and Andrew C Thomas. 2011. Homophily and contagion are generically confounded in observational social network studies. *Sociological methods & research* 40, 2 (2011), 211–239.

[27] Ashis Talukder, Md Golam Rabiul Alam, Nguyen H Tran, Dusit Niyato, Gwan Hoon Park, and Choong Seon Hong. 2019. Threshold Estimation Models for Linear Threshold-Based Influential User Mining in Social Networks. *IEEE Access* 7 (2019), 105441–105461.

[28] Christopher Tran and Elena Zheleva. 2019. Learning Triggers for Heterogeneous Treatment Effects. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[29] Thomas W Valente. 1996. Network models of the diffusion of innovations. *Computational & Mathematical Organization Theory* 2, 2 (1996), 163–164.

[30] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world'networks. *nature* 393, 6684 (1998), 440.