Heterogeneous Peer Effects in the Linear Threshold Model

Christopher Tran, Elena Zheleva

Department of Computer Science, University of Illinois at Chicago
Chicago, IL
{ctran29, ezheleva}@uic.edu

Abstract

The Linear Threshold Model is a widely used model that describes how information diffuses through a social network. According to this model, an individual adopts an idea or product after the proportion of their neighbors who have adopted it reaches a certain threshold. Typical applications of the Linear Threshold Model assume that thresholds are either the same for all network nodes or randomly distributed, even though some people may be more susceptible to peer pressure than others. To address individual-level differences, we propose causal inference methods for estimating individual thresholds that can more accurately predict whether and when individuals will be affected by their peers. We introduce the concept of heterogeneous peer effects and develop a Structural Causal Model which corresponds to the Linear Threshold Model and supports heterogeneous peer effect identification and estimation. We develop two algorithms for individual threshold estimation, one based on causal trees and one based on causal meta-learners. Our experimental results on synthetic and real-world datasets show that our proposed models can better predict individual-level thresholds in the Linear Threshold Model and thus more precisely predict which nodes will get activated over time.

Introduction

Social networks play a vital role in spreading information, ideas, and behaviors. For example, medical and agricultural innovations can spread through the world (Rogers 2010), and new products can spread via word of mouth or viral marketing (Kempe, Kleinberg, and Tardos 2003). Emotions such as happiness (Christakis and Fowler 2007) and hatefulness (Ribeiro et al. 2018) have also been observed to spread through social networks. These processes, known as *information diffusion*, have traditionally been studied in the social sciences (Granovetter 1978), but more recently have motivated applications in viral marketing (Kempe, Kleinberg, and Tardos 2003) and recommender systems (Nikolakopoulos et al. 2019). Many models exist that capture the diffusion process, including epidemic models (Kermack and McKendrick 1927), voter models (Clifford and Sudbury 1973), the Independent Cascade (Kempe, Kleinberg, and Tardos 2003), and the Linear Threshold Model (LTM) (Granovetter 1978). In this work, we focus on LTM.

According to LTM, an individual is influenced to adopt a product or idea if the proportion of their friends who have already adopted that product or idea is above some threshold. LTM is a popular model used in many settings, such as modeling the spread of ideas (Rogers 2010), predicting diffusion (Soni, Ramirez, and Eisenstein 2019), and the development of many prominent influence maximization algorithms (Kempe, Kleinberg, and Tardos 2003; Chen, Yuan, and Zhang 2010; Goyal, Lu, and Lakshmanan 2011; Li et al. 2018). However, node thresholds are typically assumed to either be the same for all nodes or randomly distributed. Moreover, there are no methods for estimating individual thresholds (Talukder et al. 2019), even though different individuals may have different susceptibility to social influence. Moreover, LTM has not been studied through a causal inference lens, even though LTM captures a causal concept: "how many friends does it take to buy a product before they *cause* an individual to buy the same product?" In our work, we seek to address these shortcomings.

We develop two models for estimating individual-level thresholds from data. Our models reflect real-world scenarios in which some individuals are more easily influenced than others or differently by specific friends. To address the variety of characteristics and behaviors of individuals, we propose a new concept, *heterogeneous peer effects* (HPEs) in networks, and contrast it with heterogeneous treatment effects for IID data. While recent work has developed methods for estimating heterogeneous treatment effects in networks, they only consider the network structure as a confounder or as a proxy to latent confounders for individual-level effects and do not estimate heterogeneous peer effects (Veitch, Wang, and Blei 2019; Guo, Li, and Liu 2020).

To facilitate the estimation of peer effects, we develop a Structural Causal Model (SCM) that is specific to LTM and encodes *interference* by *contagion*. Interference is the influence of treatments or outcomes of peers on an individual, and contagion is the process of friends' outcomes influencing an individual's outcome (Ogburn and VanderWeele 2014). For example, a user (e.g., Angelo in Figure 1) might buy sunglasses (outcome) if their friends already bought them (contagion). Prior work has studied the role of SCMs in the presence of interference (Ogburn and VanderWeele 2014; Bhattacharya, Malinsky, and Shpitser 2020) and the estimation of average peer effects (Arbour,

Garant, and Jensen 2016) but not peer effect heterogeneity. To demonstrate the value of our node threshold prediction algorithms, we evaluate them on three tasks, node threshold estimation, activated node prediction, and diffusion size prediction, using both synthetic and real-world data.

Related work

Diffusion models have been studied for decades in epidemiology (Kermack and McKendrick 1927) and opinion dynamics (Holley, Liggett et al. 1975). Two popular models for social networks are the Linear Threshold Model (LTM) (Granovetter 1978) and Independent Cascade (IC) Model (Goldenberg, Libai, and Muller 2001). These models have been used in diffusion prediction (Soni, Ramirez, and Eisenstein 2019) and influence maximization (Kempe, Kleinberg, and Tardos 2003). While some work has focused on estimating parameters of IC models (Saito, Nakano, and Kimura 2008; Bourigault, Lamprier, and Gallinari 2016; Kalimeris et al. 2018), not much work has explored threshold estimation in the Linear Threshold Model.

A recent survey on influence maximization techniques using LTM highlighted the lack of threshold estimation models (Talukder et al. 2019). Goyal et al. learn edge diffusion probabilities but do not learn individual thresholds (Goyal, Bonchi, and Lakshmanan 2010). Recent work identified a problem in threshold estimation called the "opacity problem" (Berry et al. 2019). The opacity problem states that the thresholds estimated from data are upwardly biased and propose only to use nodes whose thresholds are *precisely measured*. They use those thresholds in a regression model to estimate thresholds for all nodes. In contrast, we estimate individual thresholds on "snapshots" of networks.

Our work focuses on defining and identifying heterogeneous effects of peers through the use of a Structural Causal Model (SCM) (Pearl 2009) to estimate individual thresholds in the LTM. Shalizi and Thomas explore causal inference under the presence of homophily and contagion (Shalizi and Thomas 2011). They address the problem of identification of social effects when influence is decoupled from group effects (Manski 1993). Ogburn and VanderWeele presented an extensive discussion of SCMs for interference (Ogburn and VanderWeele 2014). Arbour et al. utilize abstract ground graphs for estimating average effects in social networks (Arbour, Garant, and Jensen 2016). To the best of our knowledge, there have been no SCMs developed for the LTM. In addition, recent work has introduced methods for estimating heterogeneous *treatment* effects in networks but do not estimate effects of peers (Guo, Li, and Liu 2020).

Problem Description

To define the problems of individual threshold estimation and conditional average peer effect estimation, we first present the data, linear threshold model, and define heterogeneous peer effects.

Data Model

Let $G = (\mathbf{V}, \mathbf{E})$ denote an attributed social network, where $\mathbf{V}$ is the set of nodes and $\mathbf{E}$ is the set of edges between nodes. If two nodes $v, u \in \mathbf{V}$ are connected by an edge, then $(v, u) \in \mathbf{E}$, denotes an edge from v to u. Define the set of neighbors of v to be $N(v) = \{u : u \in \mathbf{V}, (u, v) \in \mathbf{E}\}$. Each node $v \in \mathbf{V}$ has an observed m-dimensional vector of attributes, $\mathbf{X}_v$, unobserved characteristics, $\mathbf{U}_v$, and an 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 activated nodes at time t to be $\mathcal{D}_t = \{v : Y_v = 1\}$.

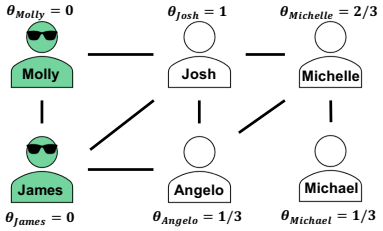


Figure 1: A toy example illustrating the Linear Threshold Model diffusion process, where Molly and James are activated. In the first time step, Angelo will become activated, since his threshold, $\theta_{Angelo} = 1/3$.

Linear Threshold Model

According to the Linear Threshold Model (LTM), each node v has an activation threshold θ_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 $v \in \mathbf{V} \setminus \mathcal{D}_i$ is activated if the *activation influence*, the weighted proportion of its activated neighbors, reaches the node's threshold θ_v:

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

where w_{uv} is the normalized influence weight of neighbor u on v. According to LTM, nodes can only become activated as activated neighbors increase. In practice, node thresholds are implemented by considering random or uniform thresholds (Talukder et al. 2019) even though the propensity to be influenced can vary from individual to individual. Our goal in this work is to estimate thresholds for all nodes.

Problem 1. *(Individual node threshold estimation for LTM) Let $G = (\mathbf{V}, \mathbf{E})$ be a graph and $\boldsymbol{\theta} = \{\theta_v \mid v \in \boldsymbol{V}\}$ be a set of node thresholds. The goal is to estimate the thresholds for all nodes, $\hat{\boldsymbol{\theta}} = \{\hat{\theta}_v \mid v \in \mathbf{V}\}$, such that the average mean squared error between θ and $\hat{\theta}$ is minimized:*

$$\underset{\hat{\theta}_v}{\operatorname{argmin}} \ \mathit{MSE} = \frac{1}{|V|} \sum_{v}^{|V|} \left(\theta_v - \hat{\theta}_v\right)^2. \tag{2}$$

Figure 1 shows a toy example of a social network with five individuals. Each node has their own threshold (e.g., $\theta_{\text{Angelo}} = 1/3$ means Angelo's threshold is $1/3$). The initial set of activations are the individuals who have adopted a product (sunglasses), which consists of two individuals: $\mathcal{D}_0 = \{\text{Molly, James}\}$. Assuming equal weights, Angelo will buy new sunglasses in the first time step since one of his three friends (James) has already bought them. No one else

will buy sunglasses in subsequent steps since Josh's threshold is 1 and Michelle's threshold is $2/3$.

Causal Inference in LTM

Here, we connect causal inference with the Linear Threshold Model under the potential outcomes framework and define heterogeneous peer effects and their estimation.

Causal inference in networks A common assumption in estimating causal effects is the stable unit treatment value assumption (SUTVA) (Rubin 1978), which is the assumption that the outcome of an individual is independent of the treatment assignment of other individuals: $Y_v(Z_v) \perp Z_u, \forall u \neq v \in \boldsymbol{V}$. However, this assumption is violated in network data because of the interaction between individuals - known as *interference* (Ogburn and VanderWeele 2014) - which can lead to biased causal effect estimations. Interference occurs in the LTM, where activation depends on the activation of neighbors. Thus, we need to account for node features *and* network interference when estimating effects.

Define $\mathbf{A} = \{Z_v \mid v \in \mathbf{V}\}$ to be the set of treatment variables for all nodes in the network. Then, the outcome of a node is dependent on two variables: the individual treatment and the set of treatments in the network: $Y_v(Z_v, \mathbf{A})$. We can define the average treatment effect (ATE) and the average peer effect (APE) (Arbour, Garant, and Jensen 2016; Fatemi and Zheleva 2020) as:

$$\text{ATE} = E[Y(Z=1) - Y(Z=0) \mid \mathbf{A} = \mathbf{a}], \quad (3)$$

$$\text{APE} = E[Y(\mathbf{A} = \mathbf{a}) - Y(\mathbf{A} = \mathbf{a}') \mid Z = z]. \quad (4)$$

ATE estimates the effect of treatment on a node, keeping other treatment assignments the same. APE estimates the effect when keeping a node's treatment fixed while changing treatments of other nodes.

Contagion is a case of *interference*, where peer outcomes in the previous time step affect individual outcomes. In the LTM, the outcome of peers in the previous time step are captured by $\mathbf{A} = \{Y_v^{t-1} \mid v \in \mathbf{V}\}$, which is the treatment variable of interest. Neighborhood-level variables (e.g. $\mathbf{A}$) are *relational* and individual-level variables (e.g. $\mathbf{X}, Y, Z$) are *propositional* variables.

Heterogeneous peer effects Heterogeneous treatment effects (HTEs) occur when subpopulations react differently to treatment. HTE is expressed through the conditional average treatment effect (CATE). CATE is defined as the expected difference in potential outcomes, conditional on an individual's features: $\tau(\mathbf{x}) \equiv E[Y(z) - Y(z') \mid \mathbf{X} = \mathbf{x}]$, which results in personalized effect estimations. In LTM, this is the expected change in activation if the node was activated vs. not activated in the previous time point with features $\mathbf{X}_v = \mathbf{x}$. So far, we have ignored the influence of neighbors. Next, we describe how to incorporate it into the HTE estimation. First, we define CATE under interference as:

$$\tau(\mathbf{x}) = E[Y(Z=1) - Y(Z=0) | \mathbf{X} = \mathbf{x}, \mathbf{A} = \mathbf{a}]. \quad (5)$$

Different individuals may have different peer effects based on personal characteristics, such as personality and susceptibility to influence. To capture this idea, we define *conditional average peer effect (CAPE)*:

$$\rho(\mathbf{x}) = E[Y(\mathbf{A} = \mathbf{a}) - Y(\mathbf{A} = \mathbf{a}') \mid \mathbf{X} = \mathbf{x}, Z = z] \quad (6)$$

Heterogeneous peer effects (HPEs) occur when there are at least two possible assignments of $\mathbf{X}$, $\mathbf{x}_i$ and $\mathbf{x}_j$, for which $\rho(\mathbf{x}_i) \neq \rho(\mathbf{x}_j)$. Since not all nodes in the network affect every other node, we use neighbors of nodes for contagion. We can define $\mathbf{A}_v = \{Y_u^{t-1} \mid u \in N(v)\}$, the set of outcomes of neighbors of v in the previous time step, which affects the outcome and APE in the current time step.

Problem 2. *(Conditional average peer effect estimation) Let $G = (\mathbf{V}, \mathbf{E})$ be a graph. Let $\hat{\rho}(\mathbf{X})$ be an estimate for the CAPE. The goal is to minimize the expected error between the estimate and its true value:*

$$\min_{\hat{\rho}} E[(\rho(\mathbf{X}) - \hat{\rho}(\mathbf{X}))^2]. \quad (7)$$

In general, it is not trivial to estimate CAPE since relational variables are sets, and $\mathbf{A}_v$ for $v \in \mathbf{V}$ are different sizes. To circumvent this problem, we model relational variables through aggregate functions, such as the mean or variance (Arbour, Garant, and Jensen 2016). Next, we discuss the link between CAPE estimation and threshold estimation.

HPE and Threshold Estimation

We begin by linking CAPE estimation to threshold estimation. Second, we introduce a causal model to identify the CAPE. Then, we map the CAPE estimation to a problem of estimating triggers for heterogeneous effects, which allows the estimation of node thresholds for LTM.

Causal model for LTM

In LTM, we define the relational variable at time t as the *activation influence* on the left side of Equation 1:

$$I_v^t = \sum_{u \in N(v)} w_{uv} Y_u^{t-1}. \quad (8)$$

We define the treatment variable, Z_v, as the previous outcome: $Z_v = Y_v^{t-1}$ and the CAPE is defined with the aggregation function in (8):

$$\rho(\mathbf{x}) = E[Y(I^t = i^t) - Y(I^t = i^{t'}) \mid \mathbf{X} = \mathbf{x}, Z = z], \quad (9)$$

where i^t and $i^{t'}$ correspond to two different values of $\mathbf{A}$. In order to identify this effect, we need to account for correct *adjustment variables*. Hence, we now introduce a causal model for the LTM process to identify CAPE for LTM.

To estimate CAPE in LTM, we develop a Structural Causal Model (SCM) of diffusion that corresponds to the LTM process. SCMs are graphical models that encode cause-effect relationships between variables and allow for the identification of effects (Pearl 2009).

Figure 2 shows the causal model of diffusion we develop to capture the LTM process. Here, $\mathbf{X}_N$ represents a *set* of neighbor features, and I_v^t is the random variable of *activation influence* on the outcomes of v's neighbors. An arrow

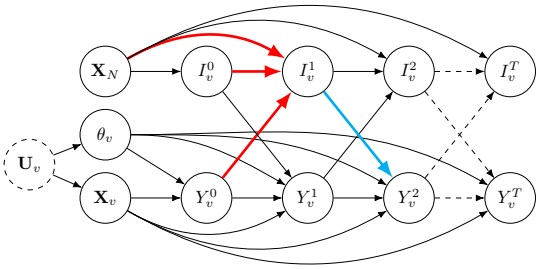


Figure 2: A causal model of peer effects for a node v that reflects the LTM process.

from one variable to another denotes a cause-effect relationship. Contagion from node v's neighbors are captured through the arrows from I_v^t to Y_v^{t+1}.

In order to *identify* the effect of activation influence on a node's own activation (e.g., the effect of I_v^1 on Y_v^2, marked in blue in Figure 2), we need to find the correct *adjustment set* that blocks all back-door paths, based on the back-door criterion (Pearl 2009; Arbour, Garant, and Jensen 2016). In our example, three arrows are going into the treatment I_v^1: Y_v^0, I_v^0 and $\mathbf{X}_N$, opening potential backdoor paths. Based on this, a sufficient set to block all backdoor paths is the set of $\mathbf{X}_v$, the previous outcome Y_v^1, and the threshold θ_v. In the CAPE formula defined in Equation (6), this is the heterogeneous subgroup $\mathbf{X} = \mathbf{x}$ that we need to consider. The special case is the initial peer effect of I_v^0 on Y_v^1, where there are no backdoor paths from I_v^0 to Y_v^1. In this case, we can estimate the effect of I_v^0 without adjustment.

To match the characteristics of LTM, we define a functional form of the outcome of node v. The outcome Y_v^{t+1} can be represented as a function of $\big(\mathbf{X}_v, Y_v^t, I_v^t, \theta_v\big)$. We can define the functional form of Y_v^{t+1} for $t \geq 1$ as:

$$Y_v^{t+1}(\mathbf{X}_v, Y_v^t, I_v^t, \theta_v) = \begin{cases} Y_v^t & \text{if } Y_v^t = 1, \\ \mathbb{1}\big[I_v^t \geq \theta_v\big] & \text{if } Y_v^t = 0. \end{cases}$$

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

Individual threshold estimation for LTM

To estimate CAPE in LTM, we estimate the threshold θ assuming that I^t is known. We first map the problem of estimating thresholds to the problem of estimating triggers (Tran and Zheleva 2019) in the context of heterogeneous peer effects. This mapping allows us to adapt causal trees to the problem of estimating thresholds and opens avenues for further algorithm development.

Since the peer influence, I_v^t, is a continuous value, it cannot be modeled using traditional causal inference methods which consider binary treatment values. We reformulate the peer effect as an effect of a binary treatment: the expected difference when a node's activation influence, I_v^t, is above and below its threshold, θ_v. Then our goal is transformed into estimating the correct node threshold that correctly identifies when a node is above and below its actual threshold. To do this, we map the threshold estimation problem to the problem of estimating triggers for heterogeneous effects (Tran and Zheleva 2019).

A *trigger* is defined as the minimum amount of treatment necessary to change an outcome (Tran and Zheleva 2019). Some examples of triggers are the minimum number of days a patient needs to take a medicine to be cured or 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"? Define a trigger, $\hat{\theta}$, with sets of potential outcomes above and below the trigger:

$$\mathcal{Y}^t(I^t \geq \hat{\theta}) = \{Y_v^t \mid I_v^t \geq \hat{\theta}\} \tag{10}$$

$$\mathcal{Y}^t(I^t < \hat{\theta}) = \{Y_v^t \mid I_v^t < \hat{\theta}\} \tag{11}$$

Then, we can define the average peer effect with a global trigger, $\hat{\theta}$ as: $E[\mathcal{Y}^t(I^t \geq \hat{\theta}) - \mathcal{Y}^t(I^t < \hat{\theta})]$, which is the case when a global threshold is defined for LTM.

Since we are interested in individual node thresholds, we estimate *heterogeneous triggers* rather than global triggers. We can now define CAPE with a trigger:

$$\mathcal{P}(\mathbf{x}) = E[\mathcal{Y}(I^t \geq \hat{\theta}) - \mathcal{Y}(I^t < \hat{\theta}) \mid \mathbf{X} = \mathbf{x}, Z = z]. \tag{12}$$

Estimating CAPE with a trigger translates to estimating the effect of influence crossing the trigger. The implication of mapping the node threshold estimation to a trigger-based heterogeneous treatment effect estimation problem is that an accurate trigger is the best estimate of the node threshold.

Theorem 1. *For any node $v \in \boldsymbol{V}$, let θ_v be v's true threshold. Then $\hat{\theta}_v$ that maximizes the CAPE with a trigger*

$$\underset{\hat{\theta}_v}{\text{argmax}} \; E[\mathcal{Y}(I_v^t \geq \hat{\theta}_v) - \mathcal{Y}(I_v^t < \hat{\theta}_v) \mid \mathbf{X}_v, Z_v], \tag{13}$$

provides the best estimate of the node threshold, θ_v.

The proof is in the Appendix[1].

Trigger-based Causal Trees Adapting trigger-based causal tree methods to find individual thresholds is straightforward. Causal trees work similarly to decision trees in that they greedily split using a partition function. The main difference between causal trees and decision trees is that the goal of a causal tree is to estimate CATE for different populations of individuals rather than to predict a label. Causal trees work by maximizing a partition function based on the difference of mean in outcomes while keeping low variance outcomes (Athey and Imbens 2016). The intuition is that finding splits that maximize the difference in means finds the heterogeneity in effect.

Tran and Zheleva developed a trigger-based causal tree method called CT-HV (Tran and Zheleva 2019). CT-HV works by introducing a validation set for generalizing CATE estimations and reducing variance in outcomes. In order to learn triggers, an additional search is done at each tree split to find the trigger that maximizes the effect estimation for

[1]https://github.com/edgeslab/hpe-ltm

that split. We can utilize CT-HV for threshold estimation through CAPE with a trigger. Instead of an individual-level treatment, we use the activation influence, I^t, to estimate triggers. Details of their algorithm are in the Appendix.

ST-Learner Now we present a novel causal *meta-learner* for estimating triggers for heterogeneous effects. A causal meta-learner uses the outputs of base learners for estimating effects. A base learner can be any regression or classification method, such as Linear Regression or Decision Trees. Künzel et al. name two commonly used meta-learners in causal inference literature, S- and T-learners, and develop a meta-learner called X-Learner (Künzel et al. 2019). However, none of them consider the problem of trigger-based HTE and are not directly suitable for solving the problem of threshold estimation. Following the idea of causal meta-learners, we propose a novel algorithm, *ST-Learner*[2].

The goal of our ST-Learner is to learn the trigger that maximizes the effect. A base learner is trained to predict the outcome (Y_v) given all node features ($\mathbf{X}_v$) and the activation influence (I_v): $E[Y_v|\mathbf{X}_v, I_v]$. Once a learner has been built on the training data (i.e., known activations), the next step is to estimate the trigger for each node. The estimation considers different possible values of I_v and consists of two steps: outcome prediction and trigger estimation.

Let $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_n\}$ be all the possible treatment values (i.e., activation influence levels I_v) in the data. Define $\Theta = \{r_1, r_2, \ldots, r_m\}, m \leq n$, to be a set of triggers. We use r_i to refer to any potential trigger, while θ_v is a *node's* individual trigger. In practice, we can consider all potential values of treatment in the training data to be possible triggers, or we can consider a discretization of these values ($m \leq n$). For any node, we can compute n predictions for all possible treatment values $\beta_k \in \boldsymbol{\beta}$: $E[Y_v|\mathbf{X}_v, I = \beta_k]$.

With the predictions based on different activation influence levels, we can estimate CAPE with a trigger. Let $\Theta_i^1 = \{\beta_k : \beta_k \geq r_i\}$ and $\Theta_i^0 = \{\beta_j : \beta_j < r_i\}$ be the set of treatment values above and below some trigger r_i. For an arbitrary trigger r_i, we get the trigger estimation for ST-Learner:

$$\underset{r_i}{\operatorname{argmax}}\ \hat{\mathcal{P}}_s(\mathbf{x}) = \frac{1}{|\Theta_i^1|} \sum_{\beta_k \geq r_i} E[Y_v|\mathbf{X}_v = \mathbf{x}, I = \beta_k] - \frac{1}{|\Theta_i^0|} \sum_{\beta_j < r_i} E[Y_v|\mathbf{X}_v = \mathbf{x}, I = \beta_j].$$

We use the base learner to estimate the outcomes above and below the trigger by taking average outcomes. This results in a time complexity of $O(|B| \cdot L)$ where $O(L)$ is the complexity of prediction for a base learner. Pseudocode for ST-Learner is available in the Appendix.

Experiments

We study the effectiveness of our proposed methods using four synthetic network generation models and three real-world datasets. We consider three tasks: threshold prediction, activated node prediction, and diffusion size prediction.

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

Experimental setup

Our main goal in this work is to estimate node thresholds (task 1) and understand their role in downstream tasks, such as activated node prediction (task 2) and diffusion size prediction (task 3). Comparing LTM to other diffusion models (e.g., cascade models) is outside the scope of this work. To compare the effectiveness of threshold prediction on these three tasks, we estimate the node thresholds given the data at a snapshot (e.g., end of Jan 2016). A snapshot is the current structure and activations at network time t. Each model predicts thresholds based on individual snapshots of data. For example, models learn thresholds using the network snapshot at the end of Jan 2016 ($\mathcal{D}_0$) and estimate node thresholds for all inactivated nodes in the test data. In this way, we also assess how having snapshots with more activations, and thus more training data, impacts predictions.

We use four baseline threshold estimation methods, two of which are based on a recent LTM survey (Talukder et al. 2019). *Heuristic Expected* computes the same threshold for all nodes in the network. *Heuristic Individual* estimates a range of values and samples individual node thresholds randomly from that range. We also employ a baseline called *Random*, that assign node 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 activated nodes in the network and estimate their threshold by computing the proportion of activated neighbors. All baseline methods focus on node threshold prediction for LTM.

We compare *Causal Tree* and *ST-Learner* to the baseline methods. For *ST-Learner*, we use two base learners, Linear Regression (ST-LR) and Decision Trees (ST-DT). For neighbor influences, we use degree centrality: $w_{uv} = 1/|N(v)|$, where $|N(v)|$ is the degree of node v.

Task 1: node threshold prediction For the first task, the goal is to accurately predict individual node thresholds. Given a snapshot of the data at time t, we estimate a threshold for each node. To evaluate the effectiveness of each model, we use the average mean squared error of predicted thresholds across all snapshots.

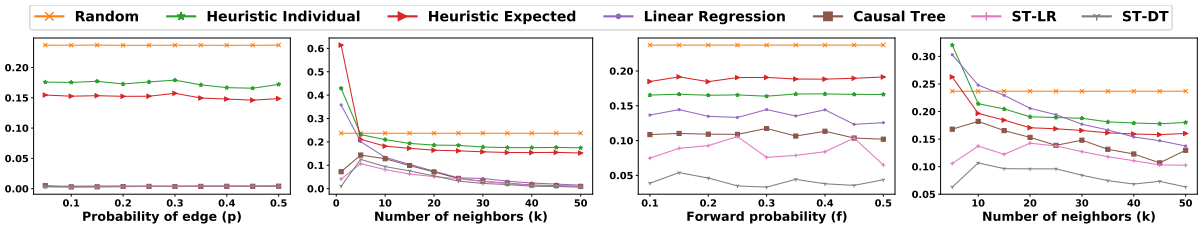
$$\text{MSE} = \frac{1}{T \cdot |V|} \sum_{t=1}^{T} \sum_{v}^{|V|} \left(\theta_v - \hat{\theta}_v^{(t)}\right)^2. \tag{14}$$

This task is evaluated based on synthetic data because the true thresholds are unknown in real-world data.

Task 2: activated node prediction In the second task, we are interested in how well each model can predict which specific nodes activate at each time step. To do this, we use the predicted node thresholds and simulate a diffusion according to LTM. Let $\mathcal{D}_t$ be the set of activated nodes at time t and $\hat{\mathcal{D}}_t$ be the set of predicted activated nodes at time t. We compute the average Jaccard index on all sets as:

$$J = \frac{1}{T} \sum_{t=0}^{T} \frac{|\mathcal{D}_t \cap \hat{\mathcal{D}}_t|}{|\mathcal{D}_t \cup \hat{\mathcal{D}}_t|}. \tag{15}$$

The Jaccard index shows how accurate a model's estimated thresholds are for predicting activated nodes.



(a) Erdos-Renyi (b) Pref. attachment (c) Forest fire (d) Watts-Strogatz

Figure 3: MSE of threshold prediction for the *Linear* setups over different parameters.

Task 3: diffusion size prediction The final task is diffusion size prediction. Using the simulated diffusion, we plot the number of predicted activations and compare them to the true number of activations at each time step. This task shows how node threshold prediction impacts diffusion size prediction. We refer to the diffusion size as the *reach*.

Datasets

We evaluate task 1 under multiple synthetic network scenarios and tasks 2 and 3 with three real-world datasets.

Synthetic datasets. We study four graph generation models: Erdos-Renyi (Erdős and Rényi 1960), preferential attachment (Barabási and Albert 1999), forest fire (Leskovec, Kleinberg, and Faloutsos 2005), and Watts-Strogatz (Watts and Strogatz 1998) and two threshold generation models. For each set of {network generation model, network parameter, threshold generation model}, we run 10 simulations and report average results. We set the number of nodes to 1000, and for each node, we randomly generate 100 node attributes from a Gaussian, $N(0, 1)$.

In Erdos-Renyi, we vary the probability of edge creation p from 0.05 to 0.5. In preferential attachment, the number of new attachments k 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 the Watts-Strogatz networks, we fix the probability of rewiring an edge to 0.1, and vary nearest neighbors k from 1 to 50.

In the first threshold generation, we use a random linear regression model with 10 of 100 attributes and normalize the output to be the user threshold, called the *Linear* setup. In the second threshold generation, we use 2 features to separate thresholds into four quadrants, where each quadrant receives a single threshold uniformly from $U(0, 1)$, called the *Quadrant* setup. 50 nodes are randomly activated, and diffusion events are generated based on LTM for 8 time steps.

Hateful Users. The Hateful Users dataset is a retweet network from Twitter, with 200 most recent tweets for each user (Ribeiro et al. 2018). Each user is represented by the average *Empath* category based on their tweets (Fast, Chen, and Bernstein 2016). Empath captures a wide variety of topics such as violence, fear, and warmth. A sample of users were selected to be annotated as *hateful* or *not hateful* and the rest were predicted based on the history of tweets using the methodology in (Ribeiro et al. 2018).

We estimate node thresholds for how "hatefulness" spreads through the network, where being activated means you change from *not hateful* to *hateful*. 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.

Cannabis. The Cannabis dataset is a follower network. The dataset covers all users who tweet about both cannabis and the e-cigarette Juul. From the users who tweet about Juul, we identify those users who also tweet about cannabis or marijuana. Empath categories are used as attributes. We estimate thresholds for how cannabis tweets spread. Activation means an individual tweets a cannabis related tweet. We consider the period between Jan 2017 to Dec 2017.

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 between July 1 and July 8 of 2012 and is a follower network on Twitter (De Domenico et al. 2013). We estimate node thresholds for the mention of the Higgs-boson discovery. Activation means a user tweets about the Higgs-boson discovery. There are no node attributes, so we construct features based on the graph. We use degree centrality, both in-degree and out-degree, and counts of user and neighborhood tweets. We consider hourly diffusions from July 4, 12:00am until July 8, 12:00am.

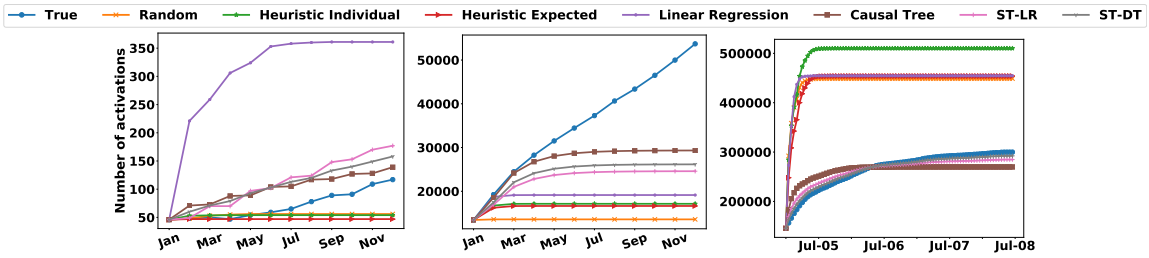
Task 1 results: node threshold prediction

Since real-world datasets do not have true node thresholds, we present results on the synthetic datasets for this task. Figure 3 shows the MSE for each threshold estimation method varied across network generation models for the *Linear* threshold generation method. *Quadrant* dataset results are available in the Appendix and show ST-DT performing the best overall. We see that the node threshold estimators perform better across setups compared to baseline methods that do not consider node threshold prediction. Our methods also perform noticeably better than the Linear Regression baseline in forest fire, and Watts-Strogatz. All individualized threshold prediction models perform noticeably better than models that do not consider node threshold prediction.

From the results, ST-DT performs the best overall on the *Linear* setup. Just estimating thresholds using a regression method, like Linear Regression, does not always perform better than other baselines. For example, Linear Regression has very high error in Watts-Strogatz networks for smaller number of neighbors. ST-DT also performs the best overall on the nonlinear threshold in the *Quadrant* setup, followed by Causal Tree.

Table 1: Our models predict the specific nodes that will activate most accurately based on highest average Jaccard index.

Threshold prediction method	Linear setup				Quadrant setup				Real-world data			
	Erdos-Renyi	Pref. Attach.	Forest Fire	Watts-Strogatz	Erdos-Renyi	Pref. Attach.	Forest Fire	Watts-Strogatz	Hateful 2016	Hateful 2017	Cannabis	Higgs
Random	0.6981	0.7986	0.9094	0.8082	0.9830	0.8319	0.9187	0.7877	0.2450	0.2262	0.2033	0.6287
Heuristic Expected	0.6268	0.7415	0.9649	0.7557	0.9783	0.7736	0.9630	0.7082	0.2608	0.2300	0.2433	0.6429
Heuristic Individual	0.9127	0.8728	0.9103	0.8398	0.9785	0.7801	0.9692	0.7196	0.2617	0.2301	0.2691	0.6436
Linear Regression	0.9082	0.8589	0.9454	0.8306	0.9891	0.8532	0.9494	0.7940	0.1522	0.2297	0.3674	0.6449
Causal Tree	0.9599	0.9357	0.9509	0.8383	0.9978	0.9559	0.9638	0.8542	**0.7207**	0.5793	**0.7830**	0.9427
ST-Learner	0.9658	0.9471	0.9622	0.8618	0.9972	0.9508	0.9526	0.8059	0.6877	0.5785	0.7564	0.9457
ST-Learner (Tree)	**0.9813**	**0.9777**	**0.9636**	**0.8669**	**0.9983**	**0.9801**	**0.9663**	**0.8672**	0.7128	**0.5988**	0.7811	**0.9734**



(a) Hateful Users (b) Cannabis (c) Higgs

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.

Task 2 results: activated node prediction

For task 2, we show results for both synthetic and real-world datasets. Table 1 shows the average Jaccard index for all datasets. Overall, our models are able to achieve the best average Jaccard index across all datasets which is especially pronounced in the real-world datasets. ST-DT achieves the highest average Jaccard index in all synthetic datasets. In the real-world datasets, we see that our models are able to achieve significantly higher average Jaccard index compared to the baseline methods. One potential reason for the discrepancy between the synthetic and real-world datasets is that synthetic datasets have generated features and a specific linear or quadrant threshold function. On the other hand, our formulation and estimation using individual level features can better capture correct node thresholds for LTM for more accurate prediction in real-world scenarios.

Task 3 results: diffusion size prediction

We show results on diffusion size prediction on one network snapshot for real-world datasets in Figure 4. A good model should have a curve that is close to the true diffusion curve (in blue). Other snapshots can be found in the Appendix and are consistent with the results here.

Hateful Users dataset. Figure 4a shows the diffusion size predictions for the Hateful Users dataset from Jan 2016 to Dec 2016. Our models initially overestimate reach, but are able to predict close to the final reach. With more training data they estimate reach more accurately. In this dataset, Linear Regression significantly overestimates diffusion prediction in all snapshots. In the 2017 dataset we notice the same trends as in the 2016 dataset, so we omit the plots.

Cannabis dataset. Figure 4b shows diffusion size predictions on the Cannabis dataset from Jan 2017 to Dec 2017. In contrast to the Hateful Users dataset, all models predict reach that saturates after a number of time steps. One reason for this could be the sparsity of edges. Additionally, different parts of the network may not necessarily interact with each other, which results in disjoint subgraphs.

Higgs dataset. Figure 4c shows the diffusion size predictions for the Higgs dataset. Specifically, we start at noon, July 4th and increase the starting snapshots by 12 hours each time. We see that the ST-DT performs the best overall among our methods, and our methods perform significantly better than baselines. Additionally, the baselines significantly overestimate the reach predictions.

Conclusion

In this work, we proposed a causal inference approach for estimating node thresholds in the Linear Threshold Model (LTM). We defined a new concept of heterogeneous peer effect estimation, and developed a structural causal model for LTM to identify and estimate peer effects. We developed a new meta-learner, the ST-Learner, and adapted trigger-based causal trees to solve the threshold estimation problem through heterogeneous peer effects. Our results on synthetic and real-world datasets showed our models are able to estimate individualized thresholds from data better than baseline methods, and can produce more accurate sets of activated nodes and diffusion size predictions in the context of LTM, especially for real-world data. A fruitful avenue of research would be to develop models that estimate edge influence weights, or to jointly learn edge influence weights and thresholds through a causal inference lens.

Acknowledgments

This material is based on research sponsored in part by the Defense Advanced Research Projects Agency (DAPRA) under contract numbers HR00111990114 and HR001121C0168 and the National Science Foundation under grant No. 2047899. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

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

Athey, S.; and Imbens, G. 2016. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27).

Barabási, A.-L.; and Albert, R. 1999. Emergence of scaling in random networks. *science*, 286(5439): 509–512.

Berry, G.; Cameron, C. J.; Park, P.; and Macy, M. 2019. The opacity problem in social contagion. *Social Networks*, 56: 93–101.

Bhattacharya, R.; Malinsky, D.; and Shpitser, I. 2020. Causal inference under interference and network uncertainty. In *Uncertainty in Artificial Intelligence*, 1028–1038. PMLR.

Bourigault, S.; Lamprier, S.; and Gallinari, P. 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*, WSDM '16, 573–582. New York, NY, USA: ACM. ISBN 978-1-4503-3716-8.

Chen, W.; Yuan, Y.; and Zhang, L. 2010. Scalable influence maximization in social networks under the linear threshold model. In *2010 IEEE international conference on data mining*, 88–97. IEEE, IEEE.

Christakis, N. A.; and Fowler, J. H. 2007. The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4).

Clifford, P.; and Sudbury, A. 1973. A model for spatial conflict. Oxford University Press.

De Domenico, M.; Lima, A.; Mougel, P.; and Musolesi, M. 2013. The anatomy of a scientific rumor. *Scientific reports*, 3: 2980.

Erdős, P.; and Rényi, A. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1): 17–60.

Fast, E.; Chen, B.; and Bernstein, M. S. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 4647–4657. ACM.

Fatemi, Z.; and Zheleva, E. 2020. Minimizing Interference and Selection Bias in Network Experiment Design. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, 176–186.

Goldenberg, J.; Libai, B.; and Muller, E. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3): 211–223.

Goyal, A.; Bonchi, F.; and Lakshmanan, L. V. 2010. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, 241–250. ACM.

Goyal, A.; Lu, W.; and Lakshmanan, L. V. 2011. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *2011 IEEE 11th international conference on data mining*, 211–220. IEEE.

Granovetter, M. 1978. Threshold models of collective behavior. *American journal of sociology*, 83(6): 1420–1443.

Guo, R.; Li, J.; and Liu, H. 2020. Learning individual causal effects from networked observational data. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 232–240.

Holley, R. A.; Liggett, T. M.; et al. 1975. Ergodic theorems for weakly interacting infinite systems and the voter model. *The annals of probability*, 3(4): 643–663.

Kalimeris, D.; Singer, Y.; Subbian, K.; and Weinsberg, U. 2018. Learning diffusion using hyperparameters. In *International Conference on Machine Learning*, 2425–2433.

Kempe, D.; Kleinberg, J.; and Tardos, É. 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*, 137–146. ACM.

Kermack, W. O.; and McKendrick, A. G. 1927. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*.

Künzel, S. R.; Sekhon, J. S.; Bickel, P. J.; and Yu, B. 2019. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10): 4156–4165.

Leskovec, J.; Kleinberg, J.; and Faloutsos, C. 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*, 177–187. ACM.

Li, Y.; Fan, J.; Wang, Y.; and Tan, K.-L. 2018. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10): 1852–1872.

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

Nikolakopoulos, A. N.; Berberidis, D.; Karypis, G.; and Giannakis, G. B. 2019. Personalized Diffusions for Top-n Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM.

Ogburn, E. L.; and VanderWeele, T. J. 2014. Causal diagrams for interference. *Statistical science*, 29(4): 559–578.

Pearl, J. 2009. *Causality*. Cambridge university press.

Ribeiro, M. H.; Calais, P. H.; Santos, Y. A.; Almeida, V. A.; and Meira Jr, W. 2018. Characterizing and detecting hateful users on twitter. In *Twelfth International AAAI Conference on Web and Social Media*.

Rogers, E. M. 2010. *Diffusion of innovations*. Simon and Schuster.

Rubin, D. B. 1978. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, 34–58.

Saito, K.; Nakano, R.; and Kimura, M. 2008. Prediction of information diffusion probabilities for independent cascade model. In *International conference on knowledge-based and intelligent information and engineering systems*, 67–75. Springer.

Shalizi, C. R.; and Thomas, A. C. 2011. Homophily and contagion are generically confounded in observational social network studies. *Sociological methods & research*, 40(2): 211–239.

Soni, S.; Ramirez, S. L.; and Eisenstein, J. J. 2019. Detecting Social Influence in Event Cascades by Comparing Discriminative Rankers. In *The 2019 ACM SIGKDD Workshop on Causal Discovery*, 78–99.

Talukder, A.; Alam, M. G. R.; Tran, N. H.; Niyato, D.; Park, G. H.; and Hong, C. S. 2019. Threshold Estimation Models for Linear Threshold-Based Influential User Mining in Social Networks. *IEEE Access*, 7: 105441–105461.

Tran, C.; and Zheleva, E. 2019. Learning Triggers for Heterogeneous Treatment Effects. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Veitch, V.; Wang, Y.; and Blei, D. 2019. Using Embeddings to Correct for Unobserved Confounding in Networks. In *Advances in Neural Information Processing Systems 32*, 13792–13802.

Watts, D. J.; and Strogatz, S. H. 1998. Collective dynamics of 'small-world'networks. *nature*, 393(6684): 440.

Appendix

Individual threshold estimation for LTM

The implication of mapping the node threshold estimation to a trigger-based heterogeneous treatment effect estimation problem is that an accurate trigger correctly estimates the node threshold.

Theorem 1. *For any node $v \in \boldsymbol{V}$, let θ_v be v's true threshold. Then $\hat{\theta}_v$ that maximizes the CAPE with a trigger in eq* (12)*:*

$$\underset{\hat{\theta}_v}{\operatorname{argmax}} \; E[\mathcal{Y}(I_v^t \geq \hat{\theta}_v) - \mathcal{Y}(I_v^t < \hat{\theta}_v) \mid \mathbf{X}_v, Z_v], \quad (16)$$

provides the best estimate of the node threshold, θ_v.

Proof. Suppose $|N(v)| = n$ and w_{uv} be an arbitrary weight such that $\sum_u w_{uv} = 1$. Let the true threshold of node v be θ_v. Define $\mathbf{A_v} = \mathbf{a_i}$ to be an assignment of activations of neighbors of v and $W_v(\alpha_i)$ to be the outcome for the activated set $\mathbf{a_i}$:

$$W_v(\alpha) = \begin{cases} 0 & \text{if } I_v(\mathbf{a_i}) < \theta_v, \\ 1 & \text{if } I_v(\mathbf{a_i}) \geq \theta_v. \end{cases} \quad (17)$$

Where I_v is the activation influence. Let the set of potential outcomes, $\mathcal{Y}_v$, below and above the estimated trigger $\hat{\theta}_v$ be:

$$\mathcal{Y}_v(I_v < \hat{\theta}_v) = \{W_v(\mathbf{a_i}) \mid I_v(\mathbf{a_i}) < \hat{\theta_v}\} \quad (18)$$

$$\mathcal{Y}_v(I_v \geq \hat{\theta}_v) = \{W_v(\mathbf{a_i}) \mid I_v(\mathbf{a_i}) \geq \hat{\theta}_v\}. \quad (19)$$

Let N_0 and N_1 be the size of the sets $\mathcal{Y}_v(I_v < \hat{\theta}_v)$ and $\mathcal{Y}_v(I_v \geq \hat{\theta}_v)$, respectively. Define the set of outcomes when node v is not activated and activated with the true threshold as $\mathcal{Z}_v(0)$ and $\mathcal{Z}_v(1)$:

$$\mathcal{Z}_v(0) = \{W_v(\mathbf{a_i}) \mid I_v(\mathbf{a_i}) < \theta_v\}, \quad (20)$$

$$\mathcal{Z}_v(1) = \{W_v(\mathbf{a_i}) \mid I_v(\mathbf{a_i}) \geq \theta_v\}. \quad (21)$$

Let $N_{\mathcal{Z}_0}$ and $N_{\mathcal{Z}_1}$ be the cardinalities of set $\mathcal{Z}_v(0)$ and $\mathcal{Z}_v(1)$, respectively. The difference between $\mathcal{Y}_v$ and $\mathcal{Z}_v$ is that $\mathcal{Y}_v$ contains the true potential activations based on the *estimated* threshold and $\mathcal{Z}_v$ contains the true potential activations based on the *true* threshold. We define expectation over the sets $\mathcal{Y}_v$ and $\mathcal{Z}_v$ as the mean over the set. There are three cases for the estimated threshold, $\hat{\theta}_v$:

Case 1 ($\hat{\theta}_v < \theta_v$): We compute the expected outcomes below and above the estimated trigger $\hat{\theta}_v$:

$$\begin{aligned} E[\mathcal{Y}_v(I_v < \hat{\theta}_v)] &= E[\{W_v(\mathbf{a_i}) \mid I_v(\mathbf{a_i}) < \hat{\theta}_v\}] \\ &= \frac{1}{N_0} \sum_i W_v(\mathbf{a_i}) = 0, \end{aligned} \quad (22)$$

since there are no times when W_v will be 1 (activated) since all outcomes below the estimated threshold $\hat{\theta}_v$ are also below the true threshold θ_v. The expected value above the estimated threshold is:

$$\begin{aligned} E[\mathcal{Y}_v(I_v \geq \hat{\theta}_v)] &= E[\{W_v(\mathbf{a_i}) \mid I_v(\mathbf{a_i}) \geq \hat{\theta}_v\}] \\ = \frac{1}{N_1} \sum_i W_v(\mathbf{a_i}) = \frac{1}{N_1} \sum_i W_v(\mathbf{a_i}) &+ \frac{1}{N_1} \sum_j W_v(\mathbf{a_j}) \\ = \frac{1}{N_1} \cdot N_{\mathcal{Z}_1} &= \frac{N_{\mathcal{Z}_1}}{N_1}. \end{aligned} \quad (23)$$

The effect is the difference above and below the estimated trigger:

$$E[\mathcal{Y}_v(I_v \geq \hat{\theta}_v) - \mathcal{Y}_v(I_v < \hat{\theta}_v)] = \frac{N_{\mathcal{Z}_1}}{N_1} - 0 = \frac{N_{\mathcal{Z}_1}}{N_1} < 1. \quad (24)$$

Note that $N_{\mathcal{Z}_1} < N_1$ since N_1 has cases above the estimated threshold, which is smaller than the true threshold.

Case 2 ($\hat{\theta}_v > \theta_v$):

$$\begin{aligned} E[\mathcal{Y}_v(I_v < \hat{\theta}_v)] &= E[\{W_v(\mathbf{a_i}) \mid I_v(\mathbf{a_i}) < \hat{\theta}_v\}] \\ &= \frac{1}{N_0}\sum_i W_v(\mathbf{a_i}) + \frac{1}{N_0}\sum_j W_v(\mathbf{a_i}) \\ &= 0 + \frac{1}{N_0}\cdot(N_0 - N_{\mathcal{Z}_0}) \\ &= \frac{N_0 - N_{\mathcal{Z}_0}}{N_0}. \end{aligned} \tag{25}$$

$$\begin{aligned} E[\mathcal{Y}_v(I_v \geq \hat{\theta}_v)] &= E[\{W_v(\mathbf{a_i}) \mid I_v(\mathbf{a_i}) \geq \hat{\theta}_v\}] \\ &= \frac{1}{N_1}\cdot N_1 = 1. \end{aligned} \tag{26}$$

$$E[\mathcal{Y}_v(I_v \geq \hat{\theta}_v) - \mathcal{Y}_v(I_v < \hat{\theta}_v)] = 1 - \frac{N_0 - N_{\mathcal{Z}_0}}{N_0} < 1. \tag{27}$$

Case 3 ($\hat{\theta}_v = \theta_v$):

$$\begin{aligned} E[\mathcal{Y}_v(I_v < \hat{\theta}_v)] &= E[\{W_v(\mathbf{a_i}) \mid I_v(\mathbf{a_i}) < \hat{\theta}_v\}] \\ &= \frac{1}{N_0}\sum_i W_v(\mathbf{a_i}) = 0. \end{aligned} \tag{28}$$

$$\begin{aligned} E[\mathcal{Y}_v(I_v \geq \hat{\theta}_v)] &= E[\{W_v(\mathbf{a_i}) \mid I_v(\mathbf{a_i}) \geq \hat{\theta}_v\}] \\ &= \frac{1}{N_1}\sum_i W_v(\mathbf{a_i}) = 1. \end{aligned} \tag{29}$$

$$E[\mathcal{Y}_v(I_v \geq \hat{\theta}_v) - \mathcal{Y}_v(I_v < \hat{\theta}_v)] = 1 - 0 = 1. \tag{30}$$

The maximum causal effect only occurs when $\hat{\theta}_v = \theta_v$. Therefore if we can estimate the trigger that maximizes CAPE we find the true node threshold. As a result, we can use any trigger-based HTE estimation method to estimate node thresholds. □

Threshold estimation algorithms

Trigger-based causal trees Here we describe the Causal Tree (CT-HV) algorithm proposed in (Tran and Zheleva 2019). Let $\mathbf{X}^\ell$ be the features in partition ℓ, which represents a node in the tree which contains N_ℓ samples, and let $\hat{\mu}_1(\ell)$ and $\hat{\mu}_0(\ell)$ be the mean of outcomes when treated and non-treated. The estimate of CATE of any partition is $\hat{\tau}_c(\mathbf{X}^\ell) = \hat{\mu}_1(\ell) - \hat{\mu}_0(\ell)$. Given a partition ℓ that needs to be partitioned further into two children ℓ_1, ℓ_2, the causal tree algorithm finds the split on features that maximizes the weighted CATE in each child:

$$\max_{\ell_1,\ell_2} N_{\ell_1}\cdot\hat{\tau}_c(\mathbf{X}^{\ell_1}) + N_{\ell_2}\cdot\hat{\tau}_c(\mathbf{X}^{\ell_2}). \tag{31}$$

We refer to the two quantities $N_{\ell_1}\cdot\hat{\tau}_c(\mathbf{X}^{\ell_1})$ and $N_{\ell_2}\cdot\hat{\tau}_c(\mathbf{X}^{\ell_2})$ as partition measures for ℓ_1 and ℓ_2. In our work, we adapt the CT-HV algorithm (Tran and Zheleva 2019) for the problem of node threshold estimation.

In order to learn triggers, an additional search is done at each split to find the trigger that maximizes the effect estimation in each split. Let $F(\ell)$ represent the partition measure for CT-HV and let $\hat{\theta_v}$ be some estimated trigger for partition ℓ. Define $M_1(\ell, \hat{\theta_v})$ and $M_0(\ell, \hat{\theta_v})$ to be the expected mean outcomes when above and below the trigger $\hat{\theta_v}$, and $F(\ell, \hat{\theta_v})$ be the partition measure with trigger $\hat{\theta_v}$: $F(\ell, \hat{\theta_v}) = M_1(\ell, \hat{\theta_v}) - M_0(\ell, \hat{\theta_v}) = E[\mathcal{Y}_\ell(I \geq \hat{\theta_v}) - \mathcal{Y}_\ell(I < \hat{\theta_v})]$. Then to find a trigger in partition ℓ, we find the trigger that maximizes the partition measure: $F(\ell, \hat{\theta_v})$. This results in a unique trigger in each partition.

Algorithm 1 ST-Learner

Input: Training set $(\mathbf{X}, Y, I)$, test example $\mathbf{x}^{te}$, base learner f, possible treatments $\boldsymbol{\beta}$, possible triggers Θ
Output: The causal effect estimate for $\mathbf{x}^{te}$

function TRAIN($\mathbf{X}, Y, I$)
  Fit f to predict Y, f: $(\mathbf{X}, I) \rightarrow Y$
  $\implies f(\mathbf{X}, I) = E[Y \mid \mathbf{X}, I]$
function PREDICT($\mathbf{x}^{te}$)
  for each β_i in $\boldsymbol{\beta}$ **do**
    Compute and store $f(\mathbf{x}^{te}, \beta_i)$
  max_trigger $= 0$, max_effect $= 0$
  for each r_i in Θ **do**
    Compute $t_1 = \frac{1}{|\Theta_i^1|}\sum_{\beta_k \geq r_i} f(\mathbf{x}^{te}, \beta_k)$
    Compute $t_0 = \frac{1}{|\Theta_i^0|}\sum_{\beta_j < r_i} f(\mathbf{x}^{te}, \beta_j)$
    $e = t_1 - t_0$
    if $e >$ max_effect
      max_trigger $= r_i$, max_effect $= e$
  return max_trigger, max_effect

ST-Learner

Pseudocode is provided for ST-Learner in Algorithm **??**. Both the training and prediction subroutine are included. In the train subroutine, we train a base learner, f to predict the outcome Y using node features $\mathbf{X}$ and the activation influence I. To predict a trigger for a new test example, we compute and score the predicted outcome for all treatment values found in the dataset. Then, for each potential trigger to consider, we compute an average above and below that trigger, and find the trigger that maximizes the difference in outcomes. The additional complexity added by the search for triggers is on the order of potential treatment values $|B|$. For example, suppose a base learner of Linear Regression requires $O(d)$ time for prediction, then ST-Learner with Linear Regression will take $O(d \cdot |B|)$ time.

Additional results

Task 1: node threshold prediction Figure **??** shows the change in MSE in the *Linear* and *Quadrant* setup. From the results, we see that Causal Tree in general performs better on the nonlinear threshold in the *Quadrant* dataset, compared to the linearly generated threshold.

Task 3: diffusion size prediction

Figure **??** shows additional figures for the real-world datasets. Each figure starts at a different network snapshot, where a snapshot is the current structure and activations at time t. For example, Figure **??** learns on data starting at the

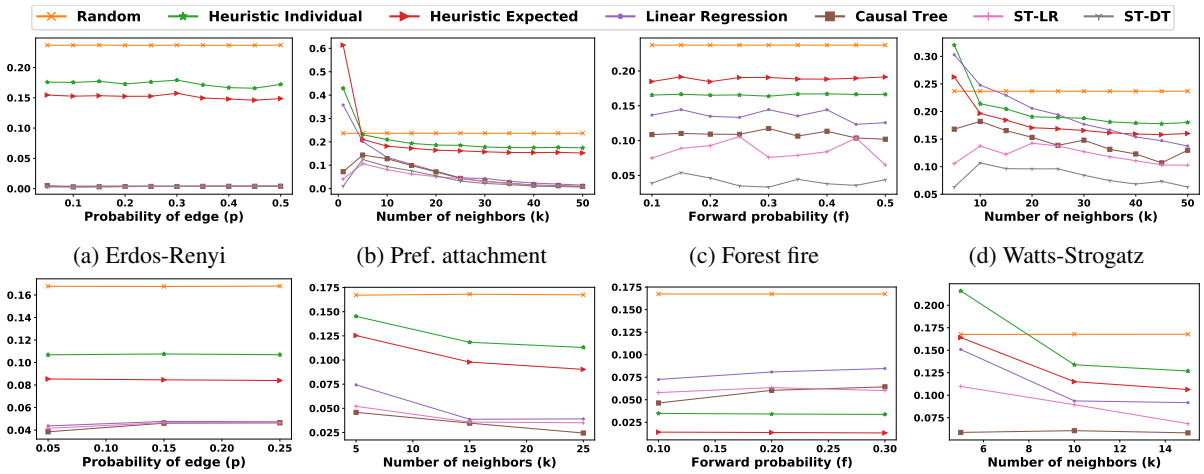


Figure 5: MSE of threshold prediction for the *Linear* (top row) and *Quadrant* (bottom row) setups over different parameters.

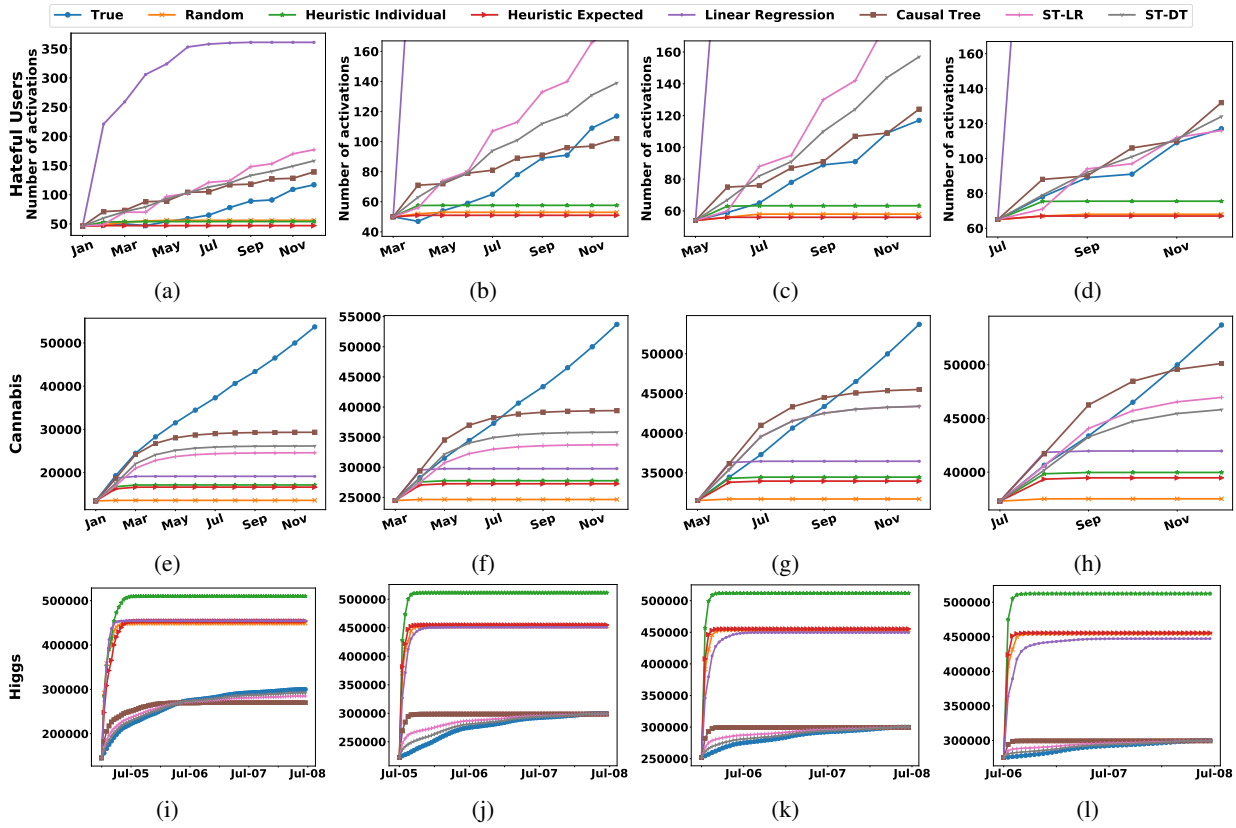


Figure 6: Comparison of diffusion size prediction on three real world datasets. Our models have the closest estimation of reach over longer time periods whereas the baselines incorrectly predict diffusion saturation in the early stages.

snapshot at the end of March 2016. A later snapshot means there are more activations in the dataset, and thus more training data. As the training data size increases, our methods can achieve better diffusion size prediction.

Hateful Users dataset. Figures **??-??** 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. Our models initially overestimate the diffusion size prediction The first thing we notice is that our models ini-

tially 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 **??** and **??**, 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 **??** (starting in May), and **??** (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.

Cannabis dataset. Figures **??**-**??** 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. Our models predict saturation closer to the final diffusion amount with more training data.

Higgs dataset. Figure **??** shows the diffusion size predictions for the Higgs dataset. Specifically, we start at noon, July 4th and increase the starting snapshots by 12 hours each time. We see that the ST-Learner performs slightly better than Causal Tree and both outperform the baselines by a significant margin. Additionally, the baselines significantly overestimate the reach predictions.