Interpretable State and Time Dependent Multi-Touch Attribution

Jad Soucar
Daniel J. Epstein Department of Industrial & Systems Engineering
University of Southern California
Los Angeles, CA 91214
soucar@usc.edu

Andres Gomez
Daniel J. Epstein Department of Industrial & Systems Engineering
University of Southern California
Los Angeles, CA 91214
gomezand@usc.edu

Johannes O. Royset
Daniel J. Epstein Department of Industrial & Systems Engineering
University of Southern California
Los Angeles, CA 91214
royset@usc.edu

Kaland Mishra
Capital One
McLean, VA 22012
kalanand.mishra@capitalone.com

Swapnil Shinde
Capital One
McLean, VA 22012
swapnil.shinde2@capitalone.com

Pranab Mohanty
Capital One
McLean, VA 22012
pranab.mohanty@capitalone.com

Abstract

In marketing contexts multi-touch attribution (MTA) aims to assign credit to a sequence of observed advertisements influencing a customer's decision to make a purchase. Existing state-of-the-art models often rely on opaque black-box predictors with post-hoc attribution (e.g., approximate Shapley values), which can be difficult to interpret and operationalize. We propose STDA, a novel interpretable **S**tate and **T**ime **D**ependent Multi-Touch **A**ttribution framework that explicitly models how advertising exposures accumulate and decay in a customer's latent purchase propensity. To efficiently solve the resulting optimization problem, we propose a multi-block penalty algorithm that employs a dynamic programming based splitting scheme and a knowledge distillation step, enabling computational tractability at scale. On synthetic data with known ground truth, the proposed algorithm is robust to noise and recovers accurate purchase patterns. On a large real-world dataset provided by a leading financial services provider, the proposed approach matches or outperforms black-box methods from the literature, while preserving white-box attribution.

Preprint.

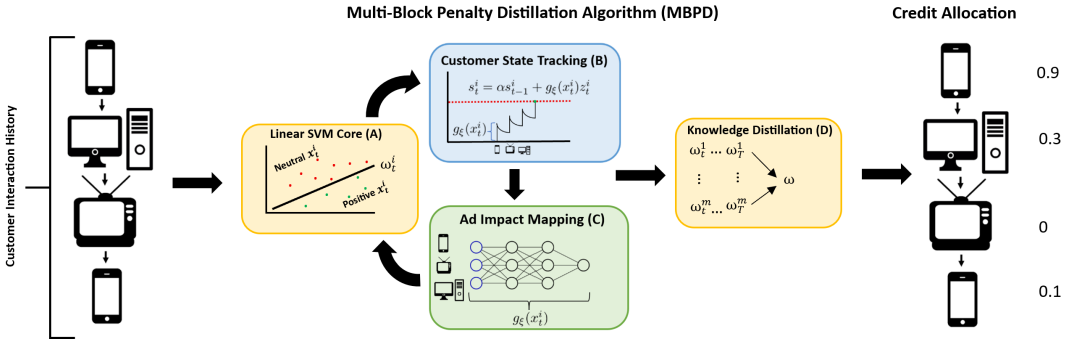


Figure 1: Modeling schematics of the proposed MBPD algorithm. The MBPD algorithm iteratively identifies periods with positive effects on a customer's purchase propensity through a learned SVM in feature space (A), incorporates the positive effects into the customer's purchase propensity state (B), then tunes a learnable advertisement impact function $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ (C). Each learned decision boundary is brought together through a knowledge distillation step (D), which encourages consistency across time and yields a single stabilized decision boundary. The credit allocations are then directly recovered from the learned decision boundary and period valences.

1 Introduction

With the growth of digital advertising campaigns has come an increase in customer-level tracking of responses and behaviors. A typical customer in this scenario is targeted and influenced by multiple advertisements before they make a decision. In this context an advertisement is defined as a deliberate time-indexed, customer-specific marketing exposure that is recordable as an observable event, such as a web pop-up, email, or mobile notification. In this setting the problem of multi-touch attribution (MTA) involves *(i)* assigning credit to these advertisements based on the influence they have on a customer's decision to make a purchase and *(ii)* predicting whether or not a customer will make a purchase given the types of advertisements they have observed. The attribution problem is challenging because a customer's purchase is observed only after a potentially long sequence of exposures with no direct information about which advertisements were critical. Additionally, advertisements typically interact non-linearly, have delayed effects on customers, and involve large, imbalanced datasets.

1.1 Related work & contributions

Previous works in the predictive MTA literature focuses on building statistical attribution models to measure the impact of marketing efforts on customer behavior. Methods developed over the past decade can be broadly categorized into white and black-box models. Popular white-box models include Li and Kannan [1], who introduces a customized Markov chain model to explicitly model purchase transition probabilities, and Zhang et al. [2], who uses a hazard-based survival model that takes into account time decay in ad-exposure response. The predominant approach, however, are black-box schemes that seek machine learned response models to allocate credit across advertisements. Examples include Shao and Li [3], who uses a bagged logistic regression model, Ren et al. [4] who uses dual-attention Recurrent Neural Networks (RNN), Du et al. [5] who also uses an RNN response model with an approximate Shapeley additve explanations (SHAP) value credit allocation scheme, and Yang et al. [6] who uses a long short-term memory (LSTM) response model with approximate SHAP credit. We acknowledge that there is a class of MTA approaches that do not exclusively use predictive models to assign attribution weights, but instead incorporate causality by combining randomized controlled trials with attention-based response models. However causal models typically require strict assumptions on the underlying data generation process such as ignorability (i.e no unobserved-confounders) [7, 8, 9, 10]. Our work does not impose such assumptions and instead contributes the body of predictive, not causal, MTA literature.

The growing trend towards difficult-to-interpret multi-touch attribution models can hamper the use of these models because their insights are not easily translated into operational marketing strategies. Much of this limitation stems from modern MTA models' failing to explicitly model how and when advertisements affect customer purchase propensity, despite the demonstrated effectiveness of such structures in longitudinal advertising studies [11, 12, 13, 14, 15]. To that end, we propose an

interpretable mixed-integer framework that explicitly models temporal purchase propensity through structured state evolution which extends the recursive adstock models of [11, 12] by coupling such dynamics across many customers and allowing for context-dependent jumps in purchase propensity. The proposed framework also learns advertisement importance weights through a joint linear support vector machine (SVM). We offer technical contributions in the form of a novel multi-block penalty distillation (MBPD) algorithm for MTA, which solves problems with multi-agent coupled state dynamics by shifting the computational burden to parallel shortest path subproblems and incorporates a teacher-student distillation step. The modeling schema is summarized in Figure 1.

1.2 Outline

The remainder of the paper is organized as follows. Section 2 develops STDA, a novel modeling framework for interpretable state- and time- dependent MTA relying on mixed-integer programming. Section 3 presents the dynamic programming and knowledge distillation driven MBPD algorithm, summarized in Figure 1, to heuristically solve the mixed-integer program at scale. Section 4 analyzes the algorithm's performance on synthetic datasets. We demonstrate that the proposed algorithm is robust to noise, recovers accurate customer purchase patterns, and outperforms ADMM approaches where dual ascent interacts poorly with integer constraints resulting in unstable integer iterates. Relying on a real-world dataset from a large financial services provider, Section 5 shows that the proposed framework matches or outperforms logistic regression, gradient-boosted trees, and LSTMs while preserving white-box attribution.

2 Problem formulation

We assume access to a dataset detailing interactions of m customers across T time periods. For each customer $i \in \{1, \dots, m\} = [m]$ and time period $t \in [T]$, we observe a tuple $(\boldsymbol{x}_t^i, y_t^i)$, where $\boldsymbol{x}_t^i$ are the features (interaction and customer context) and $y_t^i \in \{0, 1\}$ is a binary response variable with 1 corresponding to customer i making a purchase at time t. The interaction data for customer i is structured as $\boldsymbol{x}^i = \{\boldsymbol{x}_1^i, \dots, \boldsymbol{x}_T^i\}$, where

$$\boldsymbol{x}_t^i = \begin{pmatrix} x_t^{1,i} & .. & x_t^{n_a,i} & c_t^{1,i} & .. & c_t^{n_p,i} \end{pmatrix} \in \mathbb{R}^d. \tag{1}$$

The constant n_a is the total number of advertisement types and n_p is the total number of customer context features. The elements $x_t^{k,i}$ denote the number of advertisements of type k shown to customer i during time period t and $c_t^{k,i}$ denotes customer i's k^{th} context feature during time period t.

Given such data, we seek a model that predicts if and when a customer with particular feature data $\{\boldsymbol{x}_t^i \in \mathbb{R}^d, t = 1, \dots, T\}$ will commit to a sale and which portion of that data affected the decision the most. To keep track of the i^{th} customer's purchase propensity (willingness to purchase) at time t we introduce the variable s_t^i. We assume that s_t^i builds incrementally toward a purchase threshold of $s_t^i = 1$ each time a customer has a positive interaction with an advertisement. However the effects of those positive interactions diminish over time. To formalize this we let α be the discount factor that determines the degree to which interactions are retained by a customer, and $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ be a learnable function with parameter vector $\boldsymbol{\xi}$ that returns an impact of each positive interaction on s_t^i. We assume that $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) \leq 1$ which encodes that a single interaction cannot, by itself, exceed the purchase threshold. Finally we treat z_t^i as a period valence. If $z_t^i = 1$ then the advertisements viewed by customer i at time t had a positive impact on the customer's purchase propensity and s_t^i jumps by the value $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$. Putting all components together yields the recursion $s_t^i = \alpha s_{t-1}^i + g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) z_t^i$, which closely resembles adstock or Koyck distributed lag models of advertising [11, 12]. Unlike most traditional adstock models, we assume that there exists a linear decision boundary described by $\boldsymbol{\omega} \in \mathbb{R}^d, \omega_0 \in \mathbb{R}$ such that if $\langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + w_0 > 0$, then the interaction captured by $\boldsymbol{x}_t^i$ is positive ($z_t^i = 1$), otherwise it is neutral ($z_t^i = 0$).

Due to the structure of $\boldsymbol{x}_t^i$, the weight vector $\boldsymbol{\omega}$ directly encodes the importance of each advertisement type or context feature while also serving as a period valence decision boundary. In particular the first n_a weights of the decision boundary corresponding to advertisement exposures admits a natural interpretation as attribution scores. They quantify each advertisement type's marginal contribution to the positivity of an interaction. This means that the proposed optimization problem directly incorporates marginal importance values of interactions into the model, which reduces the reliance

on post-hoc approximate Shapley techniques. To find the decision boundary $\boldsymbol{\omega}$ and parameter vector $\boldsymbol{\xi}$, we solve the mixed-integer problem

$$\begin{aligned} \min_{\boldsymbol{\omega},\omega_0,\boldsymbol{s},\boldsymbol{z},\boldsymbol{\xi}} \; & \sum_{i=1}^{m}\sum_{t=1}^{T} \ell(y_t^i, s_t^i) + \mu\|\boldsymbol{\omega}\|_2^2 + \beta\|\boldsymbol{z}\|_1 + \gamma\|\boldsymbol{\omega}\|_0 + \lambda\|\boldsymbol{\xi}\|_2 \\ \text{s.t.} \quad & s_t^i = \alpha s_{t-1}^i + g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) z_t^i, \quad M z_t^i - 1 \geq \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \geq 1 - M(1 - z_t^i) \\ & z_t^i \in \{0,1\}, \quad \forall i \in [m], t \in [T], \end{aligned} \tag{STDA}$$

where M is a sufficiently large number and $\ell(y,s) = \lambda_1 \mathbf{1}\{y = 1,\ s < 1\} + \lambda_2 \mathbf{1}\{y = 0,\ s > 1\}$, where $\mathbf{1}$ is the indicator function. We refer to the mixed-integer optimization problem as the state- and time- dependent attribution (STDA) problem. The L_0- and L_2-norm regularization induce sparsity in $\boldsymbol{\omega}$ and reduce decision boundary overfitting. We use L_1-regularization of $\boldsymbol{z}$ to induce sparse period valences and L_2-regularizer on $\boldsymbol{\xi}$ to reduce overfitting of $g_{\boldsymbol{\xi}}$. Intuitively as $\mu, \gamma \to \infty$ the model will settle at the trivial decision boundary of $\boldsymbol{\omega} = \mathbf{0}$, and as $\beta \to \infty$ the model will select fewer positive periods. The (STDA) optimization problem can be specialized to other domains; see Appendix E.

3 Proposed algorithm

The large-scale multi-agent state dynamic coupling induced by the shared decision boundary vector $\boldsymbol{\omega}$, creates global dependencies across customers and time that make solving (STDA) computationally challenging at scale. In this section we develop a multi-block splitting scheme to develop a multi-block penalty algorithm with knowledge distillation, to solve (STDA). We focus on the case where $g_{\boldsymbol{\xi}}$ is trainable, and conclude with pseudo-code and implementation details for the proposed multi-block penalty distillation algorithm (MBPD) for solving (STDA).

3.1 Consensus based splitting scheme for MBPD

To split (STDA) into several algorithmically manageable subproblems we begin by introducing consensus copies of $\boldsymbol{\omega}$ and ω_0, yielding the following formulation:

$$\min_{\boldsymbol{s},\boldsymbol{z},\boldsymbol{u},\boldsymbol{v},\boldsymbol{\Omega},\boldsymbol{\sigma},\boldsymbol{\xi}} \quad \sum_{i=1}^{m}\sum_{t=1}^{T} \left(\ell(y_t^i, s_t^i) + \beta|z_t^i|\right) + \sum_{t=1}^{T} \mu\|\boldsymbol{\Omega}_t\|_2^2 + \gamma \sum_{t=1}^{T} \|\boldsymbol{\sigma}_t\|_0 + \lambda\|\boldsymbol{\xi}\|_2$$

$$\text{s.t. } s_t^i = \alpha s_{t-1}^i + u_t^i \tag{2a}$$

$$M z_t^i - 1 \;\geq\; \langle \boldsymbol{\omega}_t^i, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \;\geq\; 1 - M(1 - z_t^i) \tag{2b}$$

$$u_t^i = z_t^i v_t^i, \quad v_t^i = g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i), \quad \boldsymbol{\omega}_t^i = \boldsymbol{\Omega}_t, \quad \omega_{0,t}^i = \Omega_{0,t} \tag{2c}$$

$$\boldsymbol{\Omega}_t = \boldsymbol{\Omega}_{t-1}, \quad \Omega_{0,t} = \Omega_{0,t-1}, \quad \boldsymbol{\Omega}_t = \boldsymbol{\sigma}_t, \quad \Omega_{0,t} = \sigma_{0,t} \tag{2d}$$

$$z_t^i \in \{0,1\}, \quad \forall i \in [m], t \in [T]. \tag{2e}$$

For the purpose of brevity, we write $\tilde{\boldsymbol{\omega}}_t^i = (\boldsymbol{\omega}_t^i, \omega_{0,t}^i)$, $\tilde{\boldsymbol{\Omega}}_t = (\boldsymbol{\Omega}_t, \Omega_{0,t})$, and $\tilde{\boldsymbol{\sigma}}_t = (\boldsymbol{\sigma}_t, \sigma_{0,t})$. Using the abridged notation, we relax the problem by constructing an augmented objective function with quadratic penalties for constraints (2c) and (2d). We leave constraints (2a) and (2b) as hard constraints. We refer to the augmented objective as $\mathcal{L}(\boldsymbol{p})$ with parameters $\boldsymbol{p} = (\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{\Omega}}, \tilde{\boldsymbol{\sigma}}, \boldsymbol{\xi})$ and penalty coefficients $\rho_u, \rho_c, \rho_g, \rho_s, \rho_0 > 0$. The augmented objective $\mathcal{L}(\boldsymbol{p})$ is constructed using the standard quadratic penalty construction, which for an arbitrary problem $\min_x f(x) \quad \text{s.t} \quad g(x) = h(x)$ is $\mathcal{L}(x) = f(x) + (\rho/2)\|g(x) - h(x)\|_2^2$. The objective is described in full within Appendix A.

We apply a five block splitting scheme to iteratively solve $\min_{\boldsymbol{p}} \mathcal{L}(\boldsymbol{p})$ subject to constraints (2a) and (2b). The splitting scheme we propose can be thought of as solving the first block to determine where jumps in the customer's purchase propensity should be. The second block determines the magnitude of those jumps. The third and fourth block bring the copy variables together. The first four blocks are summarized as component (A) and (B) of Figure 1. While the final block fits the $g_{\boldsymbol{\xi}}$ to the optimal jump magnitudes and is captured by component (C) of Figure 1.

The first block is solved over $(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \tilde{\boldsymbol{\omega}})$. When all other variables are "frozen" the problem becomes separable over customers. Each customer subproblem can be recast into a dynamic program, where

at each time period a valence z_t^i is chosen. When $z_t^i = 1$ then the magnitude of the resulting jump in s_t^i is additionally selected. Each dynamic program is

$$\begin{aligned} V_t^i(s) &= \min_{z\in\{0,1\},\, u\in\mathbb{R}} \Big\{ R_t^i(z,u,s) + V_{t+1}^i(\alpha s + u) \Big\}, \\ R_t^i(z,u,s) &= \ell\big(y_t^i, \alpha s + u\big) + \beta|z| + \tfrac{\rho_u}{2}\big\|u - z v_t^i\big\|_2^2 + \psi_t^i(z) \\ \psi_t^i(z) &= \min_{\tilde{\boldsymbol{\omega}}\in\mathbb{R}^{d+1}} \tfrac{\rho_c}{2}\big\|\tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\Omega}}_t\big\|_2^2 \quad \text{s.t} \quad \begin{cases} \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i\rangle + \omega_0 \geq 1 & z = 1 \\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i\rangle + \omega_0 \leq -1 & z = 0. \end{cases} \end{aligned} \tag{3}$$

We describe the solution method for the dynamic program in Section 3.3. The second block is solved over $\boldsymbol{v}$. The resulting $\boldsymbol{v}$-block update separates over (i,t) and admits the closed form solution

$$\min_{v_t^i} \ \frac{\rho_u}{2}\big\|u_t^i - z_t^i v_t^i\big\|_2^2 + \frac{\rho_g}{2}\big\|v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)\big\|_2^2 \implies v_t^i = \frac{\rho_u z_t^i u_t^i + \rho_g g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)}{\rho_u (z_t^i)^2 + \rho_g}, \quad \forall i, t \in [m], [T]. \tag{4}$$

The third block subproblem is over $\tilde{\boldsymbol{\sigma}}$, which separates in time and over the elements of $\tilde{\boldsymbol{\sigma}}$, leaving us with $T(d+1)$ one-dimensional problems that admit closed form solution $\sigma_{0,t} = \Omega_{0,t}$ for all $t \in [T]$ and a hard thresholding closed form solution

$$\min_{\tilde{\sigma}_{t,j}} \gamma\|\sigma_{t,j}\|_0 + \frac{\rho_0}{2}(\tilde{\sigma}_{t,j} - \tilde{\Omega}_{t,j})^2 \implies \sigma_{t,j} = \begin{cases} 0, & \text{if } |\Omega_{t,j}| \leq \sqrt{2\gamma/\rho_0} \\ \Omega_{t,j} & \text{otherwise,} \end{cases} \quad \forall t, j \in [T], [d]. \tag{5}$$

The fourth block is solved over $\tilde{\boldsymbol{\Omega}}$ which yields the quadratic optimization problem

$$\min_{\tilde{\boldsymbol{\Omega}}} \quad \sum_{t=1}^{T} \Big(\mu\|\boldsymbol{\Omega}_t\|_2^2 + \frac{\rho_c}{2}\sum_{i=1}^{m}\big\|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\big\|_2^2 + \frac{\rho_s}{2}\big\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1}\big\|_2^2 + \frac{\rho_0}{2}\big\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t\big\|_2^2 \Big). \tag{6}$$

The fifth block is solved over $\boldsymbol{\xi}$. The resulting problem is

$$\min_{\boldsymbol{\xi}} \quad \frac{\rho_g}{2}\sum_{i=1}^{m}\sum_{t=1}^{T}\big\|v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)\big\|_2^2 + \lambda\,\|\boldsymbol{\xi}\|_2^2. \tag{7}$$

Algorithmically these five-block are solved iteratively with the penalty coefficients $\rho_u, \rho_c, \rho_g, \rho_s, \rho_0$ increasing after each iteration. The MBPD algorithm terminates when some stopping criteria is met, which we discuss in the next section.

3.2 Stopping criteria and knowledge distillation step

We terminate the proposed MBPD algorithm if the binary variables $\boldsymbol{z} \in \{0,1\}^{mT}$ remain constant for K_z iterations. The stopping criteria is intuitive since $\boldsymbol{z}$ encodes the time period valences for all customers, which is precisely what the SVM decision boundary is designed to identify. Using the $\boldsymbol{z}$ variables found through the multi-block penalty solver, we train an SVM which takes as input the customer interaction data $\boldsymbol{X}$ and treats $\boldsymbol{z} = \{\boldsymbol{z}^i\}_{i=1}^m$ as the labels to be predicted. The process effectively collapses the consensus variables $\{\tilde{\boldsymbol{\omega}}_i^t\}_{i,t=1}^{m,T}$ back to a single decision boundary $\tilde{\boldsymbol{\omega}} = (\boldsymbol{\omega}, \omega_0)$ and is captured by component (D) of Figure 1. This step requires solving

$$\min_{\boldsymbol{w}\in\mathbb{R}^d,\ w_0\in\mathbb{R}} \quad \frac{\nu_2}{2}\,\|\boldsymbol{w}\|_2^2 \ + \ \nu_0\,\|\boldsymbol{w}\|_0 + C\sum_{i=1}^{m}\sum_{t=1}^{T}\max\big\{0,\ 1 - z_i^t\,(\langle \boldsymbol{w}, \boldsymbol{x}_i^t\rangle + \omega_0)\big\}, \tag{8}$$

where ν_0, ν_2 are regularization coefficients and C is a penalty parameter. To retain the same structure as the original (STDA) problem ν_0, ν_2 and C are set to γ, μ and 1 respectively.

3.3 MBPD implementation

With the knowledge distillation step defined we can compactly describe the multi-block penalty distillation algorithm (MBPD) as algorithm 1.

Algorithm 1 MBPD Algorithm

Input: data ($\boldsymbol{X} \in \mathbb{R}^{mTd}, \boldsymbol{y} \in \{0,1\}^{mT}$), coefficients $\beta, \mu, \gamma, \lambda, \lambda_1, \lambda_2, C, \nu_0, \nu_2 > 0$, penalty parameters $\rho_u, \rho_g, \rho_c, \rho_s, \rho_0 > 0$, $K_{\max}$, K_z, $\delta, \epsilon > 1$.
repeat
  Block 1:
  for $i = 1, \ldots, m$ **do**
    $(\boldsymbol{s}^{i,k+1}, \boldsymbol{z}^{i,k+1}, \boldsymbol{u}^{i,k+1}, \tilde{\boldsymbol{\omega}}^{i,k+1}) \leftarrow$ sol of (3)
  end for
  Block 2: $\boldsymbol{v}^{k+1} \leftarrow$ closed-form update (4)
  Block 3: $\tilde{\boldsymbol{\sigma}}^{k+1} \leftarrow$ hard-thresholding update (5)
  Block 4: $\tilde{\boldsymbol{\Omega}}^{k+1} \leftarrow$ sol of (6)
  Block 5: $\boldsymbol{\xi}^{k+1} \leftarrow$ sol of (7)
  Penalty updates & stopping:
    Compute $\Delta \boldsymbol{z}^{k+1} = \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^{k}\|_1$
    $\rho_j \leftarrow \rho_j \epsilon$ for all $j \in \{u, g, c, s, 0\}$, $k \leftarrow k+1$
until $k \geq K_{\max}$ **or** $\Delta \boldsymbol{z}^{l} = 0$ for $l = k - K_z, \ldots, k$
Distillation: Retrieve $\boldsymbol{\omega} \leftarrow$ sol of (8)
Output: $(\boldsymbol{s}^k, \boldsymbol{z}^k, \boldsymbol{u}^k, \boldsymbol{\xi}^k), \boldsymbol{\omega}$.

Each subproblem (3), found in Block 1 of the MBPD algorithm, can be viewed as a shortest path problem over a weighted directed acyclic (DAG) graph $G = (V, E, w)$, where the vertices are defined as $V = \{(t, s) : t = 1, \ldots, T+1, s \in \mathbb{R}\}$, the edges exist between vertices of the form $(t, s) \to (t+1, s')$ where $s' \geq s\alpha$ to ensure non-negativity of u, and edge weights are $R_t^i(z, u, s)$ from equation (3) where u is implicitly $s' - \alpha s$. For learnable $g_{\boldsymbol{\xi}}$, s is discretized over a finite grid $\mathcal{S} = \{0, \Delta s, 2\Delta s, \ldots, S_{\max}\}$. We choose $S_{\max} = (1-\alpha^T)/(1-\alpha)$, since $g_{\boldsymbol{\xi}}$ is upper bounded by $U_{\max} = 1$ and so $s_t^i \leq \sum_{k=0}^{T+1} U_{\max}\alpha^k = (1-\alpha^T)/(1-\alpha)$. Each customer subproblem can be solved in parallel by a standard forward dynamic programming for DAGs or heuristic beam search algorithm in cases where $|\mathcal{S}|$ is prohibitively large; see Appendix D.5 for more implementational details.

Block 2 and 3 have closed form solutions. Block 4 can be solved using the Thomas algorithm with $\mathcal{O}(T)$ iteration complexity since subproblem (6) has a tri-diagonal Hessian. For the purpose of this paper we parameterize $g_{\boldsymbol{\xi}}$ as a multi-layer perceptron neural network with one hidden layer. This choice of ad efficacy function means that $g(x_t^i)$ can be represented as $W_2 \cdot \max\{0, W_1 x_t^i + w_1\} + w_2$ which retains interetability of the model parameters. Specifically, the weights W_1 and w_1 can be interpreted as linear gating hyperplanes that determines which ads to "accept" as having contributed to changes in purchase propensity during a particular time period for a given customer. The weights W_2 and w_2 corresponds to how much each "accepted" ad increases the purchase propensity of a customer. Consequently, block 5 can be approximated by solving (7) through an iterative solver such as stochastic gradient descent. As a simplifying assumption, the following section considers the case where $g_{\boldsymbol{\xi}}$ is constant.

Remark 3.1 (Certificate of Accuracy)**.** *Note that a valid upper bound on the minimum objective value of STDA is available through the distillation step wherein consensus among the copy variables is forced using potentially suboptimal $\boldsymbol{\xi}$ and $\boldsymbol{z}$ generated by the algorithm. As a result, the distillation produces a feasible but potentially suboptimal $\tilde{\boldsymbol{\omega}}$ decision boundary. Computing s_t^i using the recursion $s_t^i = \alpha s_{t-1}^i + g_\xi(x_t^i) z_t^i$ and plugging the values into the STDA objective yields the upper bound. A lower bound can be constructed by solving the Lagrangian relaxation of the consensus problem, which separates along time and customers. Then for any choice of Lagrangian multiplier vectors, the resulting minimum value is a valid lower bound. These bound provide a certificate of accuracy of any solution provided by the proposed algorithm.*

3.3.1 Fixed ad efficacy variation

Setting $g_{\boldsymbol{\xi}}$ equal to a fixed constant b corresponds to a case where for any positive group of interactions the effect on the customer is a constant spike in their purchase propensity s_t^i. This choice reduces the complexity of the MBPD algorithm to a 3-block method. This can be seen clearly since the $u_t^i = z_t^i v_t^i$ and $v_t^i = g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ constraints of problem (STDA) are rendered obsolete and as a result the second and fifth block of the splitting scheme described in Section 3.1 can be removed. Additionally the dynamic program in block 1 simplifies to a shortest path on a graph $G = (V, E, w)$, where $V = \{(t, s) : t = 1, \ldots, T+1, s \in \mathbb{R}\}$, each node has the two outgoing edges $(t, s) \to (t+1, \alpha s + b)$ and $(t, s) \to (t+1, \alpha s)$ corresponding to actions $z_t^i = 1$ and 0 respectively, and edge weights are $R_t^i(z, u, s)$ from equation (3). As a result of the finite number of outgoing edges the problem requires no state discretization approximation. Otherwise the algorithm remains unchanged. The derivation of the MBPD splitting scheme for the constant $g_{\boldsymbol{\xi}}$ variation tracks closely with the derivation in Section 3.1. Details are provided in Appendix B.

4 Numerical experiments on synthetic data

In this section we discuss a synthetic data generation process, before applying the MBPD algorithm. All computation in this section and Section 5 is performed on a Linux machine with 16 logical cores and 16GB of RAM. To assess performance under the most transparent specialization of the model, this section focuses on the simplified case where where $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ is set to a constant b. We also initially test on small instances to allow comparison against direct solutions of (STDA) using Gurobi. We later test in Section 5 on larger real-world datasets where Gurobi does not scale. We begin by presenting how accurately the model predicts the period valences (z_t^i) and whether the constructed s_t^i vectors are able to accurately predict whether a customer will purchase a product. We conclude with a discussion on robustness to noise and a comparison to an alternating direction method of multipliers variation.

4.1 Synthetic data generation

To generate synthetic data $(\boldsymbol{X}, \boldsymbol{y})$ we first select d, m, T which are respectively the number of features, customers, and time periods. Next we select the customers' retention discount factor $\alpha \in (0, 1)$, the interaction effect $b \in (0, 1)$, and the initial purchase propensity $s_0 = 0$ for each customer. Next we choose an arbitrary $\boldsymbol{\omega}^{\text{true}}, \omega_0^{\text{true}}$ to serve as the ground truth decision boundary between positive and neutral time periods. Before finally generating random $\hat{\boldsymbol{X}} = \{\boldsymbol{x}_t \in \mathbb{R}^d\}_{t=1}^T$ in which there exists a subset $G \subseteq \hat{\boldsymbol{X}}$, representing the set of positive time periods, such that

$$\langle \boldsymbol{\omega}^{\text{true}}, \boldsymbol{x}_t \rangle + \omega_0^{\text{true}} \begin{cases} > 0, & \boldsymbol{x}_t \in G \\ < 0, & \boldsymbol{x}_t \notin G \end{cases} \quad \text{and} \quad |G| > 0.$$

We repeat this process for each customer to generate their respective interaction data, $\boldsymbol{x}^i = (\boldsymbol{x}_1^i, \ldots, \boldsymbol{x}_T^i)$. Next we set $z_t^i = 1$ if $\langle \boldsymbol{\omega}^{\text{true}}, \boldsymbol{x}_t^i \rangle + \omega_0^{\text{true}} > 0$ and 0 otherwise. Using the ground truth time period valences we construct each customer's true purchase propensity time series s_t^i using the recursion $s_t^i = \alpha s_{t-1}^i + z_t^i b$. We assume that for any time period at which $s_t^i \geq 1$, the customer makes a purchase. As a result we set $y_t^i = 1$ if $s_t^i \geq 1$ and 0 otherwise. This data generation scheme yields $(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{z})$ along with true attribution weights $\boldsymbol{\omega}^{\text{true}}, \omega_0^{\text{true}}$.

4.2 MBPD performance

Throughout this section we use random synthetic datasets $(\boldsymbol{X}, \boldsymbol{y})$ generated using the parameters $m = T = 30, d = 2, \alpha = 0.9, b = 0.7$. The resulting (STDA) problem has 3600 constraints and 1803 variables. We then use the MBPD algorithm with fixed $g_{\boldsymbol{\xi}}$ and the same α, b parameters used to generate the synthetic data, which we refer to as B-MBPD. We compare against an oracle which solves the mixed-integer program (MIP) directly via Gurobi 12.0.3 with default solver parameters, 1 hour time limit and 1e-4 MIP gap stopping criterion. Next we construct a corresponding out-of-sample dataset $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})$ with $\bar{m}$ customers for each random synthetic dataset $(\boldsymbol{X}, \boldsymbol{y})$ using the same true decision boundaries and process described in Section 4.1.

We present the results in Table 1, where Acc is the percentage of the $T\bar{m}$ out-of-sample interaction features $\boldsymbol{x}_t^i$ that were correctly classified as having a positive or neutral effect and Acc_{g} is the percentage of the out-of-sample $\bar{m}$ customers that were correctly classified as having made or not made a purchase during the T time periods. We also provide the precision and recall for the $y_t^i = 1$ class, where precision is defined as the fraction of predicted positives that are true positives and recall is defined as the fraction of true positives that are correctly identified.

On average the MBPD algorithm for $K_{\max} = 250$ and Gurobi MIP solver requires 4.5 and 0.5 seconds of walltime respectively on the synthetic data. As shown in the B-MBPD column of Table 1, the method achieves valence and global accuracy that are close to the corresponding values obtained by the Gurobi MIP solver. Moreover the standard deviation of valence and global accuracy for the B-MBPD algorithm is small, indicating that a large fraction of runs maintain consistently high accuracy. While Gurobi is capable of solving the MIP to optimality and producing higher valence and global purchase accuracy than the MBPD on small synthetic datasets, as shown in the MIP column of Table 1, such an approach does not scale to larger datasets. Additionally, comparing the valence and global accuracy rows of Table 1, we find that the valence structure is recovered more faithfully than the global purchase patterns for both the B-MBPD and Gurobi MIP solvers. This asymmetry is due to error propagation in the latent state dynamics, where even a single misclassified valence can either

push the state s_t^i across the purchase threshold too early or prevent it from crossing at the correct time, which degrades global prediction accuracy. The results of B-ADMM are discussed in Section 4.3.

We conclude with three auxiliary experiments. First, we investigate the algorithm's performance under noisy interaction features $\boldsymbol{x}^i$ to mirror real-world uncertainty in how customers interact with advertisements. Specifically we let $\boldsymbol{x}_\delta^i = \boldsymbol{x}^i + [\mathcal{N}(0,\delta)]_{i,j}^{d,T}$. The results in the B-MBPD column of Table 4 reflect that as noise scales there is only a small decrease in valence and global accuracy. This indicates that the MBPD algorithm is robust to noise in terms of finding effective valence decision boundaries. Second we scale the synthetic datasets to 80201 variables and 160000 and report the valence accuracy in Table 3. We find that as the synthetic dataset grows in size the valence accuracy remains stable and runtime scale linearly in T. Third a fix a random dataset and study the effects of varying α and b on the learned decision boundary. As expected, we find that as α and b decrease the learned decision boundary must classify more period as "good" in order to achieve high global prediction accuracy. In this way, adjusting α and b can be viewed as adjusting the threshold for what constitutes a good or neutral time period; see Appendix D.3.

Table 1: Synthetic classification performance across 100 random training datasets with $\bar{m} = 10000$ customers. The MBPD algorithm and ADMM-Distillation are run with $K_{\max} = 250$ and $g_{\boldsymbol{\xi}} = b$.

Target	Metric	b-mbpd	mip	b-admm
z_t^i	Mean Acc	0.914	0.974	0.885
	Std Acc	0.0854	0.0212	0.120
	Mean Prec	0.791	0.941	0.701
	Mean Rec	0.772	0.891	0.824
y_t^i	Mean Acc_g	0.850	0.936	0.840
	Std Acc_g	0.0827	0.0489	0.0901
	Mean Prec_g	0.895	0.958	0.858
	Mean Rec_g	0.927	0.960	0.966

4.3 ADMM-distillation variation & comparison

A natural extension of the MBPD algorithm, which can be viewed as an alternating minimization scheme, is the alternating direction method of multipliers (ADMM). Incorporating an ADMM-type ascent-descent scheme replaces the quadratic penalty objective $\mathcal{L}$ with an augmented Lagrangian $\bar{\mathcal{L}}$, from which five new block subproblem are derived, and a dual-ascent step is added. For our purposes, the augmented Lagrangian is constructed by relaxing constraints (2c) and (2d), and leaving constraint (2a) and (2b) as hard constraints. Otherwise the algorithm remains similar. We refer to the resulting algorithm as the ADMM-Distillation algorithm. The derivation of ADMM-Distillation is similar to that of MBPD; see Appendix C.

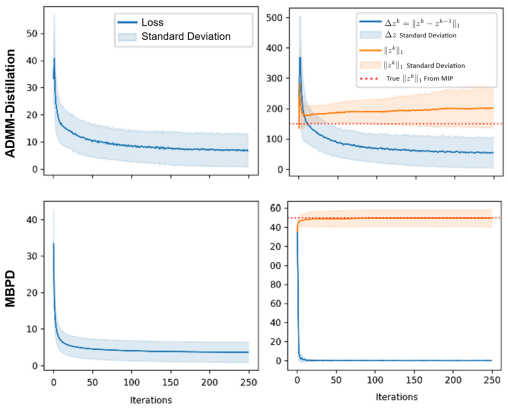


Figure 2: Stability of the valence variables $\boldsymbol{z}$ for ADMM-Distillation and MBPD algorithms. Left panels show the objective value across iterations. Right panels show the change in $\boldsymbol{z}$ over iterations, $\|\boldsymbol{z}\|_1$, and the average $\|\boldsymbol{z}\|_1$ for $\boldsymbol{z}$ obtained by solving the MIP via Gurobi. Quantities contain the mean across runs $\pm$ one standard deviation.

In Tables 1 and 4 we show that the ADMM-Distillation variation with fixed $g_{\boldsymbol{\xi}}$ (B-ADMM) consistently under performs the MBPD algorithm with fixed $g_{\boldsymbol{\xi}}$ on valence accuracy, global purchase accuracy, and most notably when noise is introduced to the feature space. To examine this behavioral difference, we inspect $\Delta z^k = \|\boldsymbol{z}^k - \boldsymbol{z}^{k-1}\|_1$, which captures how stable the binary variables are from iteration to iteration in Figure 2. We find that the MBPD algorithm has binary variables $\boldsymbol{z}$ that stabilize rapidly toward the $\boldsymbol{z}^*$ found by solving the MIP using Gurobi, which enables a quick transition toward the distillation step. Whereas the ADMM-Distillation method does not exhibit the same stabilization and rarely meets the stopping criteria described in Section 3.2. This makes the

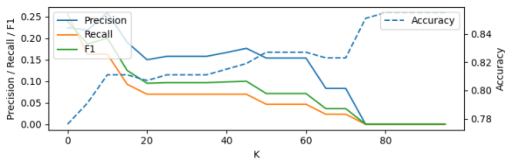


Figure 3: Precision, recall, F1, and accuracy for the MBPD algorithm with fixed $g_{\boldsymbol{\xi}}$ and Context-Enriched features when the top K features ranked by the magnitude of their corresponding ω elements are nullified.

ADMM scheme's distillation step unreliable and results in an average walltime of 46 seconds across the runs in Table 1. Such behavior is consistent with instability in the dual variables induced by the nonconvex integer constraints. Consequently, we employ only the MBPD algorithm in the subsequent case study, since it provides stable valence estimates and reliable distillation behavior.

5 Case study

In this section we investigate the ability of the proposed MBPD algorithm to allocate credit across financial services advertisements. The data $(\boldsymbol{X}, \boldsymbol{y})$ is of the same form described in equation (1) with 217 unique advertisement and 31 customer features. The MTA dataset is comprised of 4666 randomly sampled customers with a total of 365290 advertisements observed over 90 days. The data used is confirmed to contain no personal identifiable information. We find that 14.5% of customers ever make a purchase, 0.3% of advertisements are the last touch before a purchase, and on average a customer who makes a purchase interacts with approximately 49.8 advertisements before the purchase event indicating delays in conversion signals. Consequently the dataset used is representative of many longitudinal MTA datasets and provides a robust testbed to study the MBPD algorithm.

5.1 Case study set up

We begin with the application of the MBPD algorithm to solve (STDA), where $(\boldsymbol{X}, \boldsymbol{y})$ is the financial services MTA data described above. Given the size of the data the resulting optimization problem contains 1679760 constraints and approximately 840190 variables depending on the structure of $\boldsymbol{x}_t^i$ and size of $\boldsymbol{\xi}$. For both cases of $g_{\boldsymbol{\xi}}$ we tune parameters $\lambda_1, \lambda_2, \alpha$, and the purchase threshold which was previously set to 1 to maximize the model's recall on the $y_t^i = 1$ class. In the fixed $g_{\boldsymbol{\xi}}$ we tune the constant b. In the learnable $g_{\boldsymbol{\xi}}$ case we parameterize $g_{\boldsymbol{\xi}}$ as a fully connected multi-layer perceptron neural network with 1 hidden layer using Pytorch 2.9.1 and tune the input dimension of the hidden layer. We denote the MBPD algorithm with $g_{\boldsymbol{\xi}}$ parametrized as a neural network as NN-MBPD and the fixed g variant as b-MBPD. In both cases we set $K_{\max} = 250$.

5.2 Case study results

To benchmark the proposed method we deploy three commonly used models for the task of multi-touch attribution (MTA) and present the global purchase accuracy results in the benchmark group of Table 2. Specifically we benchmark the MBPD algorithm using the prevailing strategy in recent years, which has been to train supervised learning models like logistic regression, tree-based ensembles, or neural sequence models on the customer's full history of advertisement interactions [3, 16, 17, 18]. The details regarding hyper parameter tuning of the benchmark models is left to Appendix D.6. We view our incremental solution approach, of tracking and updating the purchase propensity state, as a mechanism to accomplish purchase outcome prediction and interpretable attribution. Indeed, the benchmarked models accomplish the same fundamental task of predicting purchase outcomes then extracting attributions using Shapely values and are commonly used as benchmarks in the MTA literature [2, 5, 18, 16, 19, 20]. We use a train-validate-test split of 3:1:1 for all models. Note that the ground truth period valence is unavailable in a real-world setting, so we provide the 5-fold cross validated results for predicting global accuracy in Table 2.

Table 2: Global purchase metrics for nn-MBPD, b-MBPD, Logistic Regression (LR), XGBoost (XGB), and LSTM. Values denote mean $\pm$ standard deviation across 5-fold cross validation.

MODEL	ACC$_g$	PREC$_g$	REC$_g$
NN-MBPD	$.851 \pm .035$	$.538 \pm .034$	$.215 \pm .051$
B-MBPD	$.766 \pm .041$	$.257 \pm .059$	$.225 \pm .028$
LR	$.689 \pm .027$	$.250 \pm .032$	$.629 \pm .099$
XGB	$.833 \pm .023$	$.453 \pm .089$	$.139 \pm .025$
LSTM	$.709 \pm .029$	$.230 \pm .051$	$.478 \pm .092$

On average the B-MBPD and NN-MBPD algorithm applied to the financial services dataset had walltime of 26 minutes and 32 minutes respectively. Whereas the Gurobi MIP solver times out after 1 hour without reaching the 1e-4 MIP gap stopping criteria. Both the learnable and fixed $g_{\boldsymbol{\xi}}$ cases exhibit high global purchase accuracy in the fixed and flex rows of Table 2. Additionally, in Figure 3, we find that the decision boundary weights are able to attribute the most important features correctly. Specifically if we remove the top K advertisement types from the dataset, ranked by the magnitude of their corresponding $\boldsymbol{\omega}$ element, we observe a substantial decrease in precision,

recall, and f1 scores. Additionally, accuracy increases to 0.855, however this increase corresponds to the model approaching a trivial no-purchase classifier. The output of the model can be further processed to produce customer trajectories over a sequence of advertisements leading to a purchase which is explored in Appendix G. We find that XGBoost (XGB), in the Bench row of tables 2, achieves the highest accuracy but the lowest recall among the benchmarks which indicates that few converted customers are successfully identified. In contrast, logistic regression (LR) and LSTM exhibit low accuracy but high recall and precision, which suggests over-classification of purchases. The white-box additive hazard model developed by [2] collapses to a trivial classifier that nearly always predicts no-purchase. On the large real-world financial services MTA dataset discussed above, MBPD with b-MBPD attains .766 accuracy on average which remains competitive with benchmarks while retaining easily interpretable model weights. Replacing b with a learnable $g_{\boldsymbol{\xi}}$ increases accuracy and precision to 0.851 and 0.538, which outperforms all tested baselines. We leave the door open for more flexible or domain-specific choices of g based on a practitioner's desired model interpretability.

6 Conclusion & limitations

We propose an interpretable, state- and time- dependent model for multi-touch attribution that explicitly models how advertising interactions accumulate and decay in a customer's latent purchase propensity. We then introduce the scalable MBPD algorithm using consensus decoupling, multi-block minimization, and a knowledge distillation step. We conclude that the proposed MBPD algorithm exhibits stability in the integer variables which enables effective knowledge distillation. Second, that it is possible to bridge the gap between the interpretability of classical adstock models and neural approaches. We note a few limitations. First, the use of customer features within the proposed model and many other methods for MTA may include direct or indirect indications of a customer's membership within a protected group. To this end deployment of MTA algorithms should always be accompanied by regular fairness and transparency audits. Second, that global optimality cannot be guaranteed but optimality can be quantified. Some future directions includes testing augmented version of the model proposed in Appendix E and investigating the incorporation of causal data with and without ignorability assumptions.

References

[1] Hongshuang (Alice) Li and P.K. Kannan. Attributing conversions in a multichannel online marketing environment: An empirical model and a field experiment. *Journal of Marketing Research*, 51(1):40–56, February 2014. ISSN 1547-7193. doi: 10.1509/jmr.13.0050. URL http://dx.doi.org/10.1509/jmr.13.0050.

[2] Ya Zhang, Yi Wei, and Jianbiao Ren. Multi-touch attribution in online advertising with survival theory. *2014 IEEE International Conference on Data Mining*, pages 687–696, 2014. URL https://api.semanticscholar.org/CorpusID:10871245.

[3] Xuhui Shao and Lexin Li. Data-driven multi-touch attribution models. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 258–264, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450308137. doi: 10.1145/2020408.2020453. URL https://doi.org/10.1145/2020408.2020453.

[4] Kan Ren, Yuchen Fang, Weinan Zhang, Shuhao Liu, Jiajun Li, Ya Zhang, Yong Yu, and Jun Wang. Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1433–1442, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271677. URL https://doi.org/10.1145/3269206.3271677.

[5] Ruihuan Du, Yu Zhong, Harikesh Nair, Bo Cui, and Ruyang Shou. Causally driven incremental multi touch attribution using a recurrent neural network, 2019. URL https://arxiv.org/abs/1902.00215.

[6] Dongdong Yang, Kevin Dyer, and Senzhang Wang. Interpretable deep learning model for online multi-touch attribution, 2020. URL https://arxiv.org/abs/2004.00384.

[7] Dinah Shender, Ali Nasiri Amini, Xinlong Bao, Mert Dikmen, Amy Richardson, and Jing Wang. A time to event framework for multi-touch attribution. *Journal of Data Science*, 22:56–76, 2023. URL https://jds-online.org/journal/JDS/article/1336/info.

[8] Jiaming Tang. DCRMTA: Unbiased causal representation for multi-touch attribution, 2024. URL https://arxiv.org/abs/2401.08875.

[9] Randall Lewis, Florian Zettelmeyer, Brett R. Gordon, Cristobal Garib, Johannes Hermle, Mike Perry, Henrique Romero, and German Schnaidt. Amazon ads multi-touch attribution, 2025. URL https://arxiv.org/abs/2508.08209.

[10] John Bencina, Erkut Aykutlug, Yue Chen, Zerui Zhang, Stephanie Sorenson, Shao Tang, and Changshuai Wei. Lidda: Data driven attribution at linkedin, 2025. URL https://arxiv.org/abs/2505.09861.

[11] Christine Köhler, Murali K. Mantrala, Sönke Albers, and Vamsi K. Kanuri. A meta-analysis of marketing communication carryover effects. *Journal of Marketing Research*, 54(6):990–1008, December 2017. ISSN 1547-7193. doi: 10.1509/jmr.13.0580. URL http://dx.doi.org/10.1509/jmr.13.0580.

[12] Maarten J. Gijsenberg, Harald J. van Heerde, M. G. Dekimpe, and Vincent R. Nijs. Understanding the role of adstock in advertising decisions. *SSRN Electronic Journal*, 2011. ISSN 1556-5068. doi: 10.2139/ssrn.1905426. URL http://dx.doi.org/10.2139/ssrn.1905426.

[13] L. R. Klein, L. M. Koyck, and H. Goris. Distributed lags and investment analysis. *The Economic Journal*, 65(259):523, September 1955. ISSN 0013-0133. doi: 10.2307/2227337. URL http://dx.doi.org/10.2307/2227337.

[14] Darral G. Clarke. Econometric measurement of the duration of advertising effect on sales. *Journal of Marketing Research*, 13(4):345–357, November 1976. ISSN 1547-7193. doi: 10.1177/002224377601300404. URL http://dx.doi.org/10.1177/002224377601300404.

[15] Norris I. Bruce, Natasha Zhang Foutz, and Ceren Kolsarici. Dynamic effectiveness of advertising and word of mouth in sequential distribution of new products. *Journal of Marketing Research*, 49(4):469–486, August 2012. ISSN 1547-7193. doi: 10.1509/jmr.07.0441. URL http://dx.doi.org/10.1509/jmr.07.0441.

[16] Brian Dalessandro, Claudia Perlich, Ori Stitelman, and Foster Provost. Causally motivated attribution for online advertising. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, KDD '12, page 1–9. ACM, August 2012. doi: 10.1145/2351356.2351363. URL http://dx.doi.org/10.1145/2351356.2351363.

[17] Kaifeng Zhao, Seyed Hanif Mahboobi, and Saeed Bagheri. Shapley value methods for attribution modeling in online advertising. *arXiv: Econometrics*, 2018. URL https://api.semanticscholar.org/CorpusID:67370957.

[18] Kan Ren, Yuchen Fang, Weinan Zhang, Shuhao Liu, Jiajun Li, Ya Zhang, Yong Yu, and Jun Wang. Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1433–1442, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271677. URL https://doi.org/10.1145/3269206.3271677.

[19] Di Yao, Chang Gong, Lei Zhang, Sheng Chen, and Jingping Bi. Causalmta: Eliminating the user confounding bias for causal multi-touch attribution. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 4342–4352, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539108. URL https://doi.org/10.1145/3534678.3539108.

[20] Timur Kadyrov and D. Ignatov. Attribution of customers' actions based on machine learning approach. 2019. URL https://api.semanticscholar.org/CorpusID:208232698.

[21] J. O. Royset and R. Wets. *An Optimization Primer*. Springer, 2021.

[22] Tammo H.A. Bijmolt, Leo J. Paas, and Jeroen K. Vermunt. Country and consumer segmentation: multi-level latent class analysis of financial product ownership. *International Journal of Research in Marketing*, 21(4):323–340, 2004. ISSN 0167-8116. doi: https://doi.org/10.1016/j.ijresmar.2004.06.002. URL https://www.sciencedirect.com/science/article/pii/S0167811604000424. Special Issue on Global Marketing.

Appendix

A Full augmented objective with quadratic penalties for problem (2)

In section 3 we propose problem (2) with consensus copies. We relax the problem by constructing an augmented objective function with quadratic penalties for constraints (2c) and (2d). We leave constraints (2a) and (2b) as hard constraints. Recall that $\boldsymbol{p} = (\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{\Omega}}, \tilde{\boldsymbol{\sigma}}, \boldsymbol{\xi})$ and the convention established in Section 2. Specifically, that $\tilde{\boldsymbol{\omega}}_t^i = (\boldsymbol{\omega}_t^i, \omega_{0,t}^i)$, $\tilde{\boldsymbol{\Omega}}_t = (\boldsymbol{\Omega}_t, \Omega_{0,t})$ and $\tilde{\boldsymbol{\sigma}}_t = (\boldsymbol{\sigma}_t, \sigma_{0,t})$. The resulting augmented objective is

$$\begin{aligned}\mathcal{L}(\boldsymbol{p}) = &\sum_{i=1}^{m}\sum_{t=1}^{T} \ell(y_t^i, s_t^i) + \beta \sum_{i=1}^{m}\sum_{t=1}^{T} |z_t^i| + \sum_{t=1}^{T} \mu \|\boldsymbol{\Omega}_t\|_2^2 + \gamma \sum_{t=1}^{T} \|\boldsymbol{\sigma}_t\|_0 + \lambda \|\boldsymbol{\xi}\|_2^2 \\ &+ \frac{\rho_u}{2} \sum_{i=1}^{m}\sum_{t=1}^{T} \left(u_t^i - z_t^i v_t^i\right)^2 + \frac{\rho_g}{2} \sum_{i=1}^{m}\sum_{t=1}^{T} \left(v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)\right)^2 \\ &+ \frac{\rho_c}{2} \sum_{i=1}^{m}\sum_{t=1}^{T} \left\|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\right\|_2^2 + \frac{\rho_s}{2} \sum_{t=2}^{T} \left\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1}\right\|_2^2 + \frac{\rho_0}{2} \sum_{t=1}^{T} \left\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t\right\|_2^2.\end{aligned} \tag{9}$$

B MBPD algorithm for fixed b

When $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ is set to a constant b, problem (STDA) reduces to

$$\min_{\boldsymbol{\omega}, \omega_0, \boldsymbol{s}, \boldsymbol{z}} \sum_{i=1}^{m}\sum_{t=1}^{T} \ell(y_t^i, s_t^i) + \mu\|\boldsymbol{\omega}\|_2^2 + \beta\|\boldsymbol{z}\|_1 + \gamma\|\boldsymbol{\omega}\|_0 \tag{10a}$$

$$\text{s.t. } s_t^i = \alpha s_{t-1}^i + b z_t^i \qquad \forall i \in [m], \forall t \in [T] \tag{10b}$$

$$-1 + M z_t^i \geq \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \geq 1 - M(1 - z_t^i) \qquad \forall i \in [m], \forall t \in [T] \tag{10c}$$

$$\boldsymbol{z} \in \{0,1\}^{mT}. \tag{10d}$$

To solve this problem we first introduce consensus copies of $\boldsymbol{\omega}$. Specifically we introduce $\boldsymbol{\omega}_t^i \in \mathbb{R}^d$ and $\boldsymbol{\Omega}_t \in \mathbb{R}^d$ then add the additional constraints $(\boldsymbol{\omega}_t^i, \omega_{0,t}^i) = (\boldsymbol{\Omega_t}, \Omega_{0,t})$, and $(\boldsymbol{\Omega}_t, \Omega_{0,t-1}) = (\boldsymbol{\Omega}_{t-1}, \Omega_{0,t-1})$ for all $t = 1, \ldots, T$ and $i = 1, \ldots, m$. We also introduce $\boldsymbol{\sigma}_t$ and $\sigma_{0,t}$ as copy variables of $\boldsymbol{\Omega}_t$ and $\Omega_{0,t}$ respectively, along with constraints $(\boldsymbol{\Omega}_t, \Omega_{0,t}) = (\boldsymbol{\sigma}_t, \sigma_{0,t})$. Next we construct the augmented objective function with quadratic penalties. Using the same convention established in Section 2 that $\tilde{\boldsymbol{\omega}}_t^i = (\boldsymbol{\omega}_t^i, \omega_{0,t}^i)$, $\tilde{\boldsymbol{\Omega}}_t = (\boldsymbol{\Omega}_t, \Omega_{0,t})$ and $\tilde{\boldsymbol{\sigma}}_t = (\boldsymbol{\sigma}_t, \sigma_{0,t})$, we get that the penalty augmented optimization problem is

$$\begin{aligned}\min_{\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{\omega}, \boldsymbol{\Omega}, \boldsymbol{\sigma}} \mathcal{L}^b(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{\omega}, \boldsymbol{\Omega}, \boldsymbol{\sigma}) = &\sum_{i=1}^{m}\sum_{t=1}^{T} \Big(\ell(y_t^i, s_t^i) + \beta|z_t^i|\Big) + \sum_{t=1}^{T} \mu\|\boldsymbol{\Omega}_t\|_2^2 + \gamma \sum_{t=1}^{T} \|\boldsymbol{\Omega}_t\|_0 \\ &+ \sum_{i=1}^{m}\sum_{t=1}^{T} \frac{\rho_c}{2}\|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\|_2^2 + \sum_{t=2}^{T} \frac{\rho_s}{2}\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1}\|_2^2 + \sum_{t=1}^{T} \frac{\rho_0}{2}\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t\|_2^2 \\ \text{s.t} \quad & s_t^i = \alpha s_{t-1} + b z_t^i \qquad \forall i \in [m], \forall t \in [T] \\ & -1 + M z_t^i \geq \langle \boldsymbol{\omega}_t^i, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \geq 1 - M(1 - z_t^i) \qquad \forall i \in [m], \forall t \in [T] \\ & \boldsymbol{z} \in \{0,1\}^{mT}.\end{aligned} \tag{11}$$

To solve the penalty augmented problem, we employ a similar multi-block splitting scheme. We opt to solve the problem in three blocks where the first block is over $(\boldsymbol{s}, \boldsymbol{z}, \tilde{\boldsymbol{\omega}})$,the second block is over $\tilde{\boldsymbol{\sigma}}$,and the third block is over $\tilde{\boldsymbol{\Omega}}$. The subproblem for the first block splits into single customer

problems for $i = 1, \ldots, m$ of the form

$$\min_{\boldsymbol{s},\boldsymbol{z},\tilde{\boldsymbol{\omega}}} \sum_{t=1}^{T} \Big(\ell(y_t^i, s_t^i) + \beta \|z_t^i\|_1 + \frac{\rho_c}{2} \|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\|_2^2 \Big) \tag{12a}$$

$$\text{s.t} \quad s_t^i = \alpha s_{t-1}^i + b z_t^i \qquad \forall t \in [T] \tag{12b}$$

$$-1 + M z_t^i \geq \langle \boldsymbol{\omega}_t^i, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \geq 1 - M(1 - z_t^i) \qquad \forall t \in [T] \tag{12c}$$

$$\boldsymbol{z} \in \{0,1\}^{mT}. \tag{12d}$$

We can solve the problem as the following dynamic program.

$$V_t^i(s) = \min_{z \in \{0,1\}} \Big\{ \ell(y_t^i, \alpha s + bz) + \beta \|z\|_1 + \psi(z) + V_{t+1}^i(\alpha s + bz) \Big\} \tag{13a}$$

$$\psi(z) = \min_{\tilde{\boldsymbol{\omega}}} \frac{\rho_c}{2} \|\tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\Omega}}_t\|_2^2 \quad \text{s.t} \quad \begin{cases} \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \geq 1 & \text{if} \quad z = 1 \\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \leq -1 & \text{if} \quad z = 0. \end{cases} \tag{13b}$$

The second block subproblem is over $\tilde{\boldsymbol{\sigma}}$ which reduces to problem (5). Finally, to the solve the 3rd block subproblem over $\tilde{\boldsymbol{\Omega}}$, we can solve the following quadratic program.

$$\min_{\tilde{\boldsymbol{\Omega}}} \sum_{t=1}^{T} \Big(\mu \|\boldsymbol{\Omega}_t\|_2^2 + \frac{\rho_c}{2} \sum_{i=1}^{m} \|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\|_2^2 + \frac{\rho_s}{2} \|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1}\|_2^2 + \frac{\rho_0}{2} \|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t\|_2^2 \Big). \tag{14}$$

As described in Section 3.3.1, The MBPD algorithm remains the same except that the five block problems are replaced with the minimization blocks described above and the dynamic program can be solved without approximation.

C ADMM-Distillation derivation

In this section we derive the natural counterpart of the MBPD algorithm; the ADMM-Distillation algorithm. We derive the algorithm for the learnable $g_{\boldsymbol{\xi}}$ case. Like Appendix B the derivation for the fixed $g_{\boldsymbol{\xi}}$ case is similar in structure. To incorporate an ADMM-type scheme into the MBPD algorithm we begin by introducing consensus copies of $\boldsymbol{\omega}$ and ω_0. This yields problem (2). We then apply the alternating direction method of multipliers (ADMM) approach to solve problem (2). The first step in applying ADMM is constructing an augmented Lagrangian. To do so, we relax constraints (2c) and (2d) within the augmented Lagrangian and leave constraint (2a) and (2b) as hard constraints. We make use of the standard two-norm augmented Lagrangian construction (see, e.g., [21, Section 6.B]) which yields

$$\begin{aligned}
\bar{\mathcal{L}}(\boldsymbol{p}, \widehat{\boldsymbol{p}}) = & \sum_{i=1}^{m} \sum_{t=1}^{T} \big(\ell(y_t^i, s_t^i) + \beta |z_t^i| \big) + \sum_{t=1}^{T} \mu \|\boldsymbol{\Omega}_{\boldsymbol{t}}\|_2^2 + \gamma \sum_{t=1}^{T} \|\boldsymbol{\sigma}_{\boldsymbol{t}}\|_0 + \lambda \|\boldsymbol{\xi}\|_2 \\
& + \sum_{i=1}^{m} \sum_{t=1}^{T} \big\langle \widehat{u}_t^i,\ u_t^i - z_t^i v_t^i \big\rangle + \frac{\rho_u}{2} \sum_{i=1}^{m} \sum_{t=1}^{T} \big\| u_t^i - z_t^i v_t^i \big\|_2^2 \\
& + \sum_{i=1}^{m} \sum_{t=1}^{T} \big\langle \widehat{v}_t^i,\ v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) \big\rangle + \frac{\rho_g}{2} \sum_{i=1}^{m} \sum_{t=1}^{T} \big\| v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) \big\|_2^2 \\
& + \sum_{i=1}^{m} \sum_{t=1}^{T} \big\langle \widehat{\boldsymbol{\omega}}_t^i,\ \tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t \big\rangle + \frac{\rho_c}{2} \sum_{i=1}^{m} \sum_{t=1}^{T} \big\| \tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t \big\|_2^2 \\
& + \sum_{t=2}^{T} \big\langle \widehat{\boldsymbol{\Omega}}_t,\ \tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1} \big\rangle + \frac{\rho_s}{2} \sum_{t=2}^{T} \big\| \tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1} \big\|_2^2 \\
& + \sum_{t=1}^{T} \big\langle \widehat{\boldsymbol{\sigma}}_t,\ \tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t \big\rangle + \frac{\rho_0}{2} \sum_{t=1}^{T} \big\| \tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t \big\|_2^2,
\end{aligned} \tag{15}$$

where $\boldsymbol{p} = (\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{\Omega}}, \tilde{\boldsymbol{\sigma}}, \boldsymbol{\xi})$ are the primal variables, the corresponding dual variables $\widehat{\boldsymbol{p}} = (\widehat{\boldsymbol{u}}, \widehat{\boldsymbol{v}}, \widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\Omega}}, \widehat{\boldsymbol{\sigma}})$, and penalty variables are $\rho_u, \rho_c, \rho_g, \rho_s, \rho_0 > 0$.

Algorithmically, ADMM requires first making an initial guess of $\widehat{\boldsymbol{p}}^{(0)}$, second solving $\boldsymbol{p}^{(k+1)} \leftarrow \arg\min_{\boldsymbol{p}} \bar{\mathcal{L}}(\boldsymbol{p}; \widehat{\boldsymbol{p}}^{(k)})$ subject to the un-relaxed constraints (2a) and (2b), third updating the dual variables as

$$\begin{aligned}
\widehat{u}_t^{i,(k+1)} &\leftarrow \widehat{u}_t^{i,(k)} + \rho_u\big(u_t^{i,(k+1)} - z_t^{i,(k+1)} v_t^{i,(k+1)}\big), \\
\widehat{v}_t^{i,(k+1)} &\leftarrow \widehat{v}_t^{i,(k)} + \rho_g\big(v_t^{i,(k+1)} - g_{\boldsymbol{\xi}^{(k+1)}}(\boldsymbol{x}_t^i)\big), \\
\widehat{\boldsymbol{\omega}}_t^{i,(k+1)} &\leftarrow \widehat{\boldsymbol{\omega}}_t^{i,(k)} + \rho_c\big(\tilde{\boldsymbol{\omega}}_t^{i,(k+1)} - \tilde{\boldsymbol{\Omega}}_t^{(k+1)}\big), \\
\widehat{\boldsymbol{\Omega}}_t^{(k+1)} &\leftarrow \widehat{\boldsymbol{\Omega}}_t^{(k)} + \rho_s\big(\tilde{\boldsymbol{\Omega}}_t^{(k+1)} - \tilde{\boldsymbol{\Omega}}_{t-1}^{(k+1)}\big), \\
\widehat{\boldsymbol{\sigma}}_t^{(k+1)} &\leftarrow \widehat{\boldsymbol{\sigma}}_t^{(k)} + \rho_0\big(\tilde{\boldsymbol{\Omega}}_t^{(k+1)} - \tilde{\boldsymbol{\sigma}}_t^{(k+1)}\big),
\end{aligned} \tag{16}$$

then iterating. We split the problem, $\min_{\boldsymbol{p}} \bar{\mathcal{L}}(\boldsymbol{p}; \widehat{\boldsymbol{p}}^{(k)})$ subject to (2a) and (2b), into five separate blocks that are updated sequentially within each ADMM iteration. The first block is solved over $(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \tilde{\boldsymbol{\omega}})$. When all other variables are "frozen" the problem becomes separable over customers. After completing the square and dropping constant terms within the augmented Lagrangian, we find that the i^{th} customer's subproblem is

$$\begin{aligned}
\min_{\boldsymbol{s}^i, \boldsymbol{u}^i, \boldsymbol{z}^i, \tilde{\boldsymbol{\omega}}^i} \quad & \sum_{t=1}^{T} \Big(\ell(y_t^i, s_t^i) + \tfrac{\rho_u}{2} \big\| u_t^i - z_t^i v_t^i + \tfrac{1}{\rho_u} \widehat{u}_t^i \big\|_2^2 + \beta |z_t^i| + \tfrac{\rho_c}{2} \big\| \tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t + \tfrac{1}{\rho_c} \widehat{\boldsymbol{\omega}}_t^i \big\|_2^2 \Big) \\
\text{s.t.} \quad & s_t^i = \alpha s_{t-1}^i + u_t^i && \forall i \in [m], t \in [T], \\
& -1 + M z_t^i \;\geq\; \langle \boldsymbol{\omega}_t^i, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \;\geq\; 1 - M(1 - z_t^i) && \forall i \in [m], t \in [T], \\
& \boldsymbol{z}^i \in \{0,1\}^T.
\end{aligned}$$

Each customer subproblem can be recast into a dynamic program

$$\begin{aligned}
V_t^i(s) &= \min_{z \in \{0,1\},\, u \in \mathbb{R}} \Big\{ R_t^i(z, u, s) + V_{t+1}^i(\alpha s + u) \Big\}, \\
R_t^i(z, u, s) &= \ell\big(y_t^i, \alpha s + u\big) + \beta |z| + \tfrac{\rho_u}{2} \big\| u - z v_t^i + \tfrac{1}{\rho_u} \widehat{u}_t^i \big\|_2^2 + \psi_t^i(z) \\
\psi_t^i(z) &= \min_{\tilde{\boldsymbol{\omega}} \in \mathbb{R}^{d+1}} \tfrac{\rho_c}{2} \big\| \tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\Omega}}_t + \tfrac{1}{\rho_c} \widehat{\boldsymbol{\omega}}_t^i \big\|_2^2 \quad \text{s.t} \quad \begin{cases} \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \geq 1 & z = 1 \\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \leq -1 & z = 0. \end{cases}
\end{aligned} \tag{17}$$

The second block is solved over $\boldsymbol{v}$. The resulting problem is

$$\min_{\boldsymbol{v}} \quad \frac{\rho_u}{2} \sum_{i,t} \Big\| u_t^i - z_t^i v_t^i + \tfrac{1}{\rho_u} \widehat{u}_t^i \Big\|_2^2 + \frac{\rho_g}{2} \sum_{i,t} \Big\| v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) + \tfrac{1}{\rho_g} \widehat{v}_t^i \Big\|_2^2,$$

which separates over (i, t) and admits the closed form solution

$$v_t^i = \frac{\rho_u z_t^i \Big(u_t^i + \tfrac{1}{\rho_u} \widehat{u}_t^i \Big) + \rho_g \Big(g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) - \tfrac{1}{\rho_g} \widehat{v}_t^i \Big)}{\rho_u (z_t^i)^2 + \rho_g}. \tag{18}$$

for all $i \in [m]$ and $t \in [T]$. The third block subproblem is over $\tilde{\boldsymbol{\sigma}}$, which separates in time and in the elements of $\tilde{\boldsymbol{\sigma}}$, leaving us with $T(d+1)$ one-dimensional problems

$$\min_{\sigma_{t,j}} \gamma \|\sigma_{t,j}\|_0 + \frac{\rho_0}{2} \Big(\sigma_{t,j} - \big(\tilde{\Omega}_{t,j} + \tfrac{1}{\rho_0} \widehat{\sigma}_{t,j} \big) \Big)^2. \tag{19}$$

Subproblem (19) permits a closed form solution of $\sigma_{0,t} = \Omega_{0,t} + \widehat{\sigma}_{0,t}$ and a hard thresholding closed form solution

$$\sigma_{t,j} = \begin{cases} 0, & \text{if } |\Omega_{t,j} + \widehat{\sigma}_{t,j}| \leq \sqrt{2\gamma/\rho_0} \\ \Omega_{t,j} + \widehat{\sigma}_{t,j} & \text{otherwise,} \end{cases} \tag{20}$$

Algorithm 2 ADMM-Distillation for problem (STDA)

Input: data ($\boldsymbol{X} \in \mathbb{R}^{mTd}, \boldsymbol{y} \in \{0,1\}^{mT}$), coefficients $\beta, \mu, \gamma, \lambda, \lambda_1, \lambda_2, C, \nu_0, \nu_2 > 0$, penalty parameters $\rho_u, \rho_g, \rho_c, \rho_s, \rho_0 > 0$, max iterations $K_{\max}$, $\boldsymbol{z}$ window K_z, residual threshold δ, and $k = 0$

g **Type:** $\mathrm{g}_{\text{flex}} \sim$ Learnable $g_{\boldsymbol{\xi}} \in \{\text{True,False}\}$

repeat

  Block 1:

  for $i = 1, \ldots, m$ **do**

    $(\boldsymbol{s}^{i,k+1}, \boldsymbol{z}^{i,k+1}, \boldsymbol{u}^{i,k+1}, \tilde{\boldsymbol{\omega}}^{i,k+1}) \leftarrow$ Dynamic program (17) for customer i

  end for

  Block 2: $\boldsymbol{v}^{k+1} \leftarrow$ closed-form update (18) **if** g_{flex}

  Block 3: $\tilde{\boldsymbol{\sigma}}^{k+1} \leftarrow$ hard-thresholding update (20)

  Block 4 $\tilde{\boldsymbol{\Omega}}^{k+1} \leftarrow$ solution of (21)

  Block 5: $\boldsymbol{\xi}^{k+1} \leftarrow$ solution of (22) **if** g_{flex}

  Dual updates & stopping:

    Update dual variables via (16).

    Compute $\Delta \boldsymbol{z}^{k+1} = \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^{k}\|_1$.

    $k \leftarrow k + 1$

until $k \geq K_{\max}$ **or** $\Delta \boldsymbol{z}^{l} = 0$ for $l = k - K_z, \ldots, k$

Distillation: Retrieve $\boldsymbol{\omega} \leftarrow$ solution of (8)

Output: $(\boldsymbol{s}^k, \boldsymbol{z}^k, \boldsymbol{u}^k, \boldsymbol{\xi}^k), \boldsymbol{\omega}$.

for all $j \in [d]$ and $t \in [T]$. The fourth block is solved over $\tilde{\boldsymbol{\Omega}}$ which yields the quadratic problem

$$\begin{aligned} \min_{\tilde{\boldsymbol{\Omega}}} \quad \sum_{t=1}^{T} \Big(\mu \|\boldsymbol{\Omega}_t\|_2^2 + \frac{\rho_c}{2} \sum_{i=1}^{m} \big\| \tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t + \frac{1}{\rho_c} \widehat{\boldsymbol{\omega}}_t^i \big\|_2^2 + \frac{\rho_s}{2} \big\| \tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1} + \frac{1}{\rho_s} \widehat{\boldsymbol{\Omega}}_t \big\|_2^2 \\ + \frac{\rho_0}{2} \big\| \tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t + \frac{1}{\rho_0} \widehat{\boldsymbol{\sigma}}_t \big\|_2^2 \Big). \end{aligned} \tag{21}$$

The fifth block is solved over $\boldsymbol{\xi}$ and involves minimizing a quadratic penalty associated with the $v_t^i = g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ constraint,

$$\min_{\boldsymbol{\xi}} \quad \frac{\rho_g}{2} \sum_{i=1}^{m} \sum_{t=1}^{T} \Big\| v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) + \tfrac{1}{\rho_g} \widehat{v}_t^i \Big\|_2^2 + \lambda \|\boldsymbol{\xi}\|_2^2. \tag{22}$$

Incorporating these minimization block problems into the MBPD algorithm and recognizing that in the fixed $g_{\boldsymbol{\xi}}$ case block two and five are removed, we get algorithm 2. We treat g_{flex} as a toggle that enables the learnable $g_{\boldsymbol{\xi}}$ case.

D Additional experiments & experimental details

D.1 Scalability of synthetic experiments

In this section we test the proposed MBPD algorithm with flexible g_ξ on larger synthetic datasets constructed using the methodology described in Section 4.1. Specifically we test the algorithm's valence accuracy on a grid of synthetic datasets with time periods T, number of customers m, and number of features d ranging from 20-200. This means that the number of variables and constraints in the SDTA formulation ranged from 821 to 80201 and 1600 to 160000 respectively. We also record the peak memory consumption and average runtime of each (T, m, d) tuple in table 3. We found that as the number of variables/constraints increased the valence accuracy remained roughly stable. Runtime scales approximately linearly in the time horizon T and sublinearly in the number of customers m and the feature dimension d.

T	m	d	Runtime (s)	Valence Acc.
20	20	20	15.95	0.901
20	20	100	17.57	0.835
20	20	200	21.17	0.845
20	100	20	27.46	0.835
20	100	100	29.15	0.860
20	100	200	30.73	0.850
20	200	20	37.87	0.825
20	200	100	39.78	0.845
20	200	200	43.28	0.845
100	20	20	33.09	0.875
100	20	100	42.66	0.668
100	20	200	57.09	0.753
100	100	20	98.53	0.882
100	100	100	108.25	0.677
100	100	200	120.02	0.795
100	200	20	158.19	0.891
100	200	100	174.23	0.710
100	200	200	195.30	0.787
200	20	20	72.31	0.923
200	20	100	80.25	0.762
200	20	200	108.84	0.638
200	100	20	213.15	0.935
200	100	100	238.81	0.764
200	100	200	261.72	0.615
200	200	20	373.54	0.933
200	200	100	470.42	0.725
200	200	200	521.31	0.625

Table 3: Out-of-sample performance of the nn-MBPD algorithm across different problem sizes, without early termination for 250 iterations. Presented are the average runtime and valence accuracy across 10 datasets.

D.2 Noisy synthetic experiments

We repeat the experiment used to generate Table 1, except we let $\boldsymbol{x}_\delta^i = \boldsymbol{x}^i + [\mathcal{N}(0,\delta)]_{i,j}^{d,T}$. The results in the B-MBPD column of Table 4 reflect that as noise scales there is only a small decrease in valence and global accuracy. This indicates that the MBPD algorithm is robust to noise in terms of finding effective valence decision boundaries.

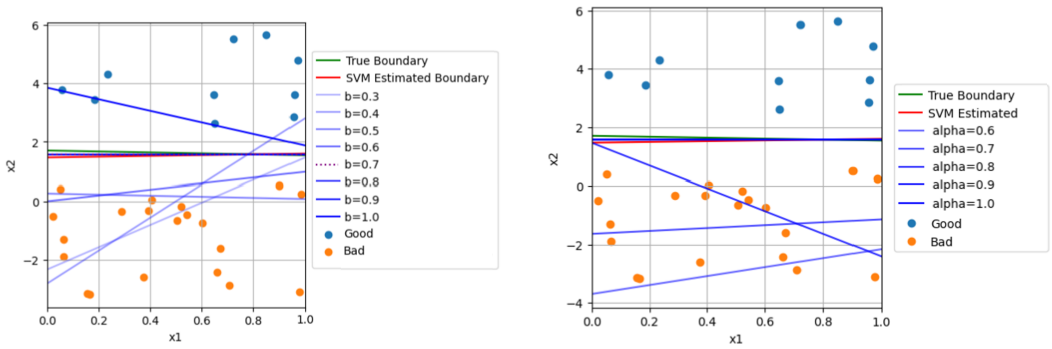
Table 4: Synthetic classification performance across 100 random training datasets with $\bar{m} = 10000$ customers and with gaussian noise added to context with variance δ. The MBPD algorithm and ADMM-Distillation are run with $K_{\max} = 250$ and $g_{\boldsymbol{\xi}} = b$.

Target	Noise	b-mbpd	b-admm
z_t^i	$\delta = 0.1$	0.912	0.839
	$\delta = 0.5$	0.925	0.837
	$\delta = 1.0$	0.925	0.811
y_t^i	$\delta = 0.1$	0.846	0.791
	$\delta = 0.5$	0.856	0.779
	$\delta = 1.0$	0.851	0.778

D.3 Sensitivity analysis on α and b in synthetic experiments.

In this section we discuss a sensitivity analysis of the state recursion parameters. Specifically we study the sensitivity of the linear decision boundary, ω, to changes in the memory (α) and ad efficacy (b) parameters. We found that varying α and b will have a "boundary shifting" effect. For example

assume that the SDTA model was solved with fixed parameters α and b. We found that if we solve the optimization problem again, with the same synthetic data where ground truth is known (described in Section 4.1), but with a new parameter $\tilde{b} < b$ then each customer requires a larger number "good" time periods to trigger a sale, and so the boundary will shift in order to classify more periods of interactions as "good" periods. So in practice decrease or increasing b controls the operator's threshold for a good or neutral period of interactions. The same effect can be observed when decrease α, since such a change corresponds to a more forgetful customer who will also require the classification of more "good" time periods.



(a) Boundary shifting with $\tilde{b}$ (true $b = 0.7$).

(b) Boundary shifting with $\tilde{\alpha}$ (true $\alpha = 0.9$).

Figure 4: Effect of misspecifying parameters on the decision boundary. The true boundary corresponds to the boundary used when generating the synthetic data used in this plot. Additionally the true b and α used to generate the data were $b = 0.7$ and $\alpha = 0.9$ respectively. Each blue line corresponds to the decision boundary found by solving the SDTA problem using the MBPD approach on the generated synthetic data.

D.4 Sensitivity to penalty parameters

We emphasize that the penalty coefficients $\rho_u, \rho_g, \rho_s, \rho_0$ and ϵ are algorithmic parameters controlling the convergence of the penalty based splitting scheme. This interpretation of these parameters' role is consistent with classical literature on ADMM, penalty methods, and augmented Lagrangian methods. In these setting analytical results for the optimal selection of penalty parameters or growth rate ϵ are limited. Even in cases where convergence guarantees exist, theory typically requires only that the penalty parameters are sufficiently large without providing explicit selection rules. As a result, in practice these parameters are selected using traditional hyper-parameter tuning or heuristics and are problem-dependent.

In our experiments, we observe that the method is robust to a wide range of parameter choices and enjoys small variation in final attribution vectors ω. Specifically selecting $\rho \in [0.5, 5]$ and $\epsilon \in [1.01, 1.5]$ typically yields stable solutions across synthetic datasets with a preference toward small initial values of ρ and large values of ϵ.

Table 5: Coefficient of determination between the $\tilde{\omega}$ retrieved from each run and the true ω used during synthetic data generation. Used the nn-MBPD algorithm with 8 synthetic datasets.

$\rho_{\text{init}} \backslash \epsilon$	1.01	1.10	1.25	1.50
0.5	0.139	0.108	0.081	0.081
1.0	0.111	0.092	0.071	0.079
2.0	0.098	0.075	0.070	0.078
5.0	0.108	0.088	0.089	0.091

D.5 State space discretizations for problem (3)

The first block of the MBPD algorithm can be recast as a dynamic programming problem, which can be viewed as a shortest path problem over a weighted directed acyclic graph. In theory this problem can be solved by discretizing the state space and applying a standard backward dynamic programming algorithm. However as noted in the section 3.3 we use a heuristic beam search algorithm to solve the DP since this avoids the problem of needing to find an optimal discretization scheme. Instead the beam search algorithm operates in continuous state space and prunes high-cost valence sequences $\{z_t^i\}_{t=1}^T$. This means that in practice the discretizations of the state space is not a practical consideration. However the beam width of the beam search algorithm is certainly of interest and the sensitivity of the results the beam width can be studied. Beam-width controls the exactness of the dynamic programming block solutions in block 1 of the MBPD algorithm.

Table 6: Relative variability of the L2 error between the recovered attribution vector and the true attribution vector ω used to generate the synthetic data as a function of beam width. Values are averaged over 8 randomly generated synthetic datasets using the nn-MBPD (flex) variant.

Beam width K	Std/Mean
4	0.497
16	0.385
64	0.274
128	0.172

We observe that the coefficient of variation decreases with beam width. This indicates that the variability of the recovered attribution vector relative to its magnitude predictable decreases as the beam search approximation becomes more accurate. In other words choosing a large beam-width is always better for solution accuracy and is limited only by the amount of compute available to the practitioner. For our purposes we choose a beam-width of 64 as not to substantially limit the speed of the MBPD algorithm.

D.6 Benchmark training details

In training the LSTM, XGB, and Logistic Regression baselines we tuned each model's relevant hyper-parameters using the Optuna library, which relies on Bayesian hyper-parameter tuning. Specifically for XGB we tuned the number of trees, max depth of each tree, learning rate, minimum leaf weight, and the minimum split gain (γ). For the Logistic Regression we tuned the L1 and L2 penalty parameters and inverse regularization parameter C. We also tested each model in the hyper-parameter grid with and without class weighting, along with the solver type. For the LST model we tuned the hidden layer size on a range of 32-512, the number of layers on a range of $1-5$, the dropout rate, learning rate and batch-size. Additionally we use a focal loss to handle the dataset's severe purchase to no-purchase class imbalance. Each benchmark model was tuned to maximize recall.

D.7 Ethics and generalizability of case study data

The real-world dataset we use utilizes fully de-identified customer-level behavioral data, confirmed to be PII-free through the use of synthetic identifiers, and its external sharing was conducted following all required internal ethics and governance procedures. Specifically we confirm that the real-world dataset used for the application does not contain any columns classified as Personally Identifiable Information. The data is at the customer level but uses only synthetic, system-generated, and obfuscated identifiers (such as a Single Sign-On ID). These IDs are non-obvious pseudonyms and cannot be traced back to an individual's real-world identity without accessing a separate, secured identity management database. Additionally the data share for this research was conducted under the supervision of the financial institution's business counsel and external data share team. The use and sharing of this data were subject to internal review and consent procedures appropriate for the data's classification, which confirmed its suitability for external collaboration. We will include this clarification in the revised draft of the paper.

E Model variations

E.1 reversal

Currently the proposed optimization problem (STDA) permits consecutive customer purchases over a contiguous period of time if $s_t^i \geq 1$ for a contiguous range of t. While in some advertising setting such consecutive purchases are common, in others it may be more likely that once a customer has made a purchase their propensity to make another purchase "resets." Formally, s_t^i gets reset to 0 once s_t^i crosses the purchase threshold of 1. Such a modeling decision could better capture sparse purchase behavior, and could be accomplished by solving

$$\min_{\boldsymbol{\omega},\omega_0,\boldsymbol{s},\boldsymbol{r},\boldsymbol{z},\boldsymbol{u}} \quad \sum_{i=1}^{m}\sum_{t=1}^{T} \ell(y_t^i, s_t^i) + \mu\|\boldsymbol{\omega}\|_2^2 + \beta\|\boldsymbol{z}\|_1 + \gamma\|\boldsymbol{\omega}\|_0 \tag{23a}$$

$$\text{s.t.} \quad r_t^i = \alpha s_{t-1}^i + b z_t^i \qquad \forall i \in [m],\ \forall t \in [T] \tag{23b}$$

$$-Mu_t^i \ \leq\ s_t^i - r_t^i \ \leq\ Mu_t^i \qquad \forall i \in [m],\ \forall t \in [T] \tag{23c}$$

$$-M(1-u_t^i) \ \leq\ s_t^i \ \leq\ M(1-u_t^i) \qquad \forall i \in [m],\ \forall t \in [T] \tag{23d}$$

$$-1 + Mz_t^i \ \geq\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \ \geq\ 1 - M(1-z_t^i) \qquad \forall i \in [m],\ \forall t \in [T] \tag{23e}$$

$$z_t^i, u_t^i \in \{0,1\} \qquad \forall i \in [m],\ \forall t \in [T], \tag{23f}$$

for the fixed $g_{\boldsymbol{\xi}} = b$ setting. In other words the additional constraints are enforcing that $r_t^i = \alpha s_t^i + b z_t^i$ and that the state recursion is

$$s_t^i = \begin{cases} r_t^i & \text{if} \quad u_t^i = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Algorithmically, the reversal variation is identical to the MBPD algorithm, with the exception that when implementing the dynamic programming function we enforce that every time a customer passes the sale threshold of 1 their respective s_t^i is reset to 0.

E.2 Time-dependent decision boundaries

Problem (STDA) assumes that a group of m customer's preferences are constant from time $t = 1$ to T. In some contexts it may be more likely that preferences slowely evolve over time. Such a model is already built into the consensus formulations in equation (15) and (11). So long as the penalty parameters do not grow too large, there is some flexibility for preferences to change overtime. However such a change, where $\boldsymbol{\Omega}_{t+1}$ is penalized for drifting too far from $\boldsymbol{\Omega}_t$ would reflect sticky preferences. Algorithmically this corresponds, to simply treating $\{\boldsymbol{\Omega_t}\}_{t=1}^T$ as the final output of the model instead of employing the distillation step.

E.3 Negative-neutral-positive time periods

Problem (STDA) assumes that a time-period is either positive or negative. In the fixed $g_{\boldsymbol{\xi}} = b$ case, the problem assumes that during a positive time period the customer's purchase propensity increases by some fixed b, and during a neutral time period there is no jump in s_t^i. Instead s_t^i is allowed to continue decaying with time at rate α. There may be some advertising settings where instead of the period valence being binary, $z_t^i \in \{-1, 0, 1\}$ corresponding to a time period that has a negative impact on the customer, a neutral effect on the customer and a positive effect on the customer respectively.

Such a model could be achieved by adjusting model (10) as

$$\min_{\substack{\boldsymbol{\omega},\omega_0,\boldsymbol{s},\boldsymbol{z},\\ \boldsymbol{r}^+,\boldsymbol{r}^0,\boldsymbol{r}^-}} \ \sum_{i=1}^{m}\sum_{t=1}^{T}\ell\big(y_t^i,s_t^i\big) \ + \ \mu\|\boldsymbol{\omega}\|_2^2 \ + \ \beta\sum_{i=1}^{m}\sum_{t=1}^{T}|z_t^i| \ + \ \gamma\|\boldsymbol{\omega}\|_0 \tag{25a}$$

$$\text{s.t.} \quad s_t^i \ = \ \alpha s_{t-1}^i + b\, z_t^i \qquad \forall i\in[m],\ \forall t\in[T], \tag{25b}$$

$$r_t^{i,+} + r_t^{i,0} + r_t^{i,-} = 1 \qquad \forall i\in[m],\ \forall t\in[T], \tag{25c}$$

$$z_t^i = r_t^{i,+} - r_t^{i,-} \qquad \forall i\in[m],\ \forall t\in[T], \tag{25d}$$

$$-M\big(1-r_t^{i,+}\big) \ \le \ \langle\boldsymbol{\omega},\boldsymbol{x}_t^i\rangle+\omega_0 \ \le \ M \qquad \forall i\in[m],\ \forall t\in[T], \tag{25e}$$

$$-\varepsilon - M\big(1-r_t^{i,0}\big) \ \le \ \langle\boldsymbol{\omega},\boldsymbol{x}_t^i\rangle+\omega_0 \ \le \ M\big(1-r_t^{i,0}\big) \qquad \forall i\in[m],\ \forall t\in[T], \tag{25f}$$

$$-M \ \le \ \langle\boldsymbol{\omega},\boldsymbol{x}_t^i\rangle+\omega_0 \ \le \ -\varepsilon + M\big(1-r_t^{i,-}\big), \qquad \forall i\in[m],\ \forall t\in[T], \tag{25g}$$

$$r_t^{i,+},\ r_t^{i,0},\ r_t^{i,-}\in\{0,1\} \qquad \forall i\in[m],\ \forall t\in[T], \tag{25h}$$

for some $\epsilon > 0$. If the features $\boldsymbol{x}_t^i$ lie on the positive side of the decision boundary then the time period is classified as positive, if it lies on the negative side but has a score $\langle\boldsymbol{\omega},\boldsymbol{x}_t^i\rangle+\omega_0 \geq -\epsilon$ then the time period is neutral and if $\boldsymbol{x}_t^i$ is far into the negative half-space then it is classified as a negative time period with adverse effect $-b$ on s_t^i. This ablation would require a reformulation of the penalty augmented problem in equation (11) and would therefor result in slightly different algorithms than the MBPD algorithm. However the same algorithmic framework of splitting the problem into blocks using consensus copies would still be applied.

E.4 Ownership classes

Problem (STDA) assumes that customers have homogeneous preferences. Such an assumption may be acceptable if the customers are pre-split into groups and one finds decision boundaries for each customer group. Alternatively we propose the following ablation. Assume that there are O ownership classes, where if a customer already owns a certain subset of products then they will belong to an ownership class. For example a potential two class structure would split customers into those that own products typically associated with commercial or personal use. Let each customer's ownership data at time t be $\boldsymbol{o}_t^i \in \{0,1\}^O$. Then the model is defined as

$$\min_{\boldsymbol{\omega},\boldsymbol{s},\boldsymbol{z}} \ \sum_{i=1}^{m}\sum_{t=1}^{T}\ell(y_t^i,s_t^i) + \mu\|\boldsymbol{\omega}\|_2^2 + \beta\|\boldsymbol{z}\|_1 + \gamma\|\boldsymbol{\omega}\|_0 \tag{26a}$$

$$\text{s.t. } s_t^i = \alpha s_{t-1}^i + b z_t^i \qquad \forall i\in[m],\forall t\in[T] \tag{26b}$$

$$-1 + M z_t^i \geq \langle(\boldsymbol{\omega},\boldsymbol{x}_t^i\rangle+\omega_0)\frac{\boldsymbol{o}_t^i}{\|o_t^i\|_2^2} \geq 1 - M(1-z_t^i) \qquad \forall i\in[m],\forall t\in[T] \tag{26c}$$

$$\boldsymbol{z}\in\{0,1\}^{mT}, \boldsymbol{\omega}\in\mathbb{R}^{dO}, \omega_0\in\mathbb{R}^O. \tag{26d}$$

This ablation can be understood as each ownership class having their own unique preferences regarding which interaction types the class deems important. If a customer belongs to multiple ownership classes then their decision boundary is the average decision boundary of each class the customer belongs to. This particular model ablation can be expanded beyond ownership classes and toward any segmentation of a population. We conclude by noting that ownership classes are well studied indicators of distinct customer behavior [22].

E.5 K-SVM quantized distillation

Previously we discussed a potential ablation where the modeler explicitly constructs ownership classes, however such a task may be difficult. We propose that, in order to construct customer classes in an unsupervised fashion, the distillation step of the MBPD algorithm can be adjusted to resemble a code-book style quantization training scheme. Specifically N different $\{(\boldsymbol{\omega}^j,\boldsymbol{\omega}_0^j)\}_{j=1}^N$

would be initialized. Using the $\boldsymbol{z}$ period valences as labels, found using the MBPD algorithm, each customer would be assigned to one of the N classes with parameters $\boldsymbol{\omega}^j, \boldsymbol{\omega}_0^j$ that best reflects that customers period valence data $\{z_t^i\}_{t=1}^T$. After each customer has been assigned, each class's decision boundary would be retrained using problem (8), followed by a reassignment of each customer. The process would repeat iteratively until class participation stabilizes. Such a scheme would segment the population based on their preferences (characterized by $\boldsymbol{w}^j, w_0^j$), while simultaneously identifying those preferences. The resulting problem is

$$\min_{\boldsymbol{\omega},\boldsymbol{u}} \sum_{i=1}^{m}\sum_{j=1}^{N} u_j^i \Big(\sum_{t=1}^{T} \max\{0, 1 - z_t^i(\langle \omega^j, \boldsymbol{x}_t^i \rangle + w_0^j)\} \Big) + \lambda \sum_{j=1}^{N} \|\boldsymbol{w}^j\|_2^2 + \beta \sum_{j=1}^{N} \Big(\sum_{i=1}^{m} u_j^i - \frac{m}{N} \Big)^2$$
$$\text{s.t} \quad \sum_{j=1}^{N} u_j^i = 1, \quad u_j^i \geq 0 \quad \forall i \in [m], \forall j \in [N],$$

where the final term is a regularizer that incentivizes classes of equal size.

F Alternative scoring metrics

Typically accuracy of a multi-touch-attribution model is determined by calculating

$$\mathrm{Acc}_1(s, y) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{T} \Big| 1 - (y_t^i + \mathbb{I}\{\|s_t^i\| > 0\}) \Big|. \tag{27}$$

In other words, the customer is marked correct if the model correctly classifies the customer as having made a purchase at any time in $[1, T]$. However such an accuracy metric ignores whether the model has the ability to predict when a purchase is made and how far s_t^i was from the purchase threshold at the time of purchase. To capture such qualities we define two more accuracy metrics. First we use the piecewise constant $\ell(y, s)$ function described in Section 2 to define

$$\mathrm{Acc}_2(s, y) = \frac{1}{Tm} \sum_{i=1}^{m} \sum_{t=1}^{T} y_t^i [1 - \ell(y_t^i, s_t^i)], \tag{28}$$

which measures the average distance from the purchase threshold when a purchase is made. However such a metric expects perfect alignment in time. Specifically if a customer makes a purchase at time t^*, then ℓ marks any purchase prediction at time $t^* + \delta$ for δ arbitrarily small, as incorrect. Such a definition may be too strict for some use cases so, we define and additional accuracy metric, $\mathrm{Acc}_3(s, y)$), as equation (28) where we use

$$\ell_\delta(y, s) = \begin{cases} \lambda_1 \min\left(1, \ \dfrac{\max(1 - s, 0)}{\delta}\right), & y = 1, \\ \lambda_2 \min\left(1, \ \dfrac{\max(s - 1, 0)}{\delta}\right), & y = 0, \end{cases} \tag{29}$$

instead of ℓ. The accuracy metric $\mathrm{Acc}_3(s, y)$ is distance sensitive so that if a customer makes a purchase at time t^* and the model predicts that the customer makes a purchase at time $t^* + \delta$ then this error is penalized less than a model that predicts a purchase at time $t^* + 2\delta$.

G Interpreting output

Next we discuss the interpretation of the model's output in the context of multi-touch attribution. After training a model by solving problem (STDA), we retrieve $\boldsymbol{\omega}, \omega_0$ and $\boldsymbol{z}$. If we are only interested in understanding which advertisements were the most important in triggering sales across all m customers then the elements of $\boldsymbol{\omega}$ already encode the relevant information. If we are only interested in which time period had the highest efficacy then $\boldsymbol{z}$ already holds that information, since an effective time period has $z_t^i = 1$ and 0 otherwise. Consequently, we could observe time period importance by computing $\frac{1}{m}(\sum_{i=1}^m z_1^i, \ldots, \sum_{i=1}^m z_T^i)$. If we aim to construct customer "trajectories" to make

statements like "customer i made a purchase after observing advertisement $a_1 \to a_2 \to a_3$", then we use the model outputs to directly construct the chain as

$$\begin{aligned}
\boldsymbol{C}^i &= \left(z_t^i \cdot \arg\max_{1\le j\le d} \quad (\omega_j \mathbb{I}\{x_t^{j,i} > 0\})_+ : t = 1, \ldots, T\right) \in \{1, \ldots, d\}^T \\
\boldsymbol{A}^i &= \left(z_t^i \cdot \max_{1\le j\le d} \quad (\omega_j \mathbb{I}\{x_t^{j,i} > 0\})_+ : t = 1, \ldots, T\right) \in \mathbb{R}^T, \quad \tilde{\boldsymbol{A}}^i = \boldsymbol{A}^i / \|\boldsymbol{A}^i\|_2,
\end{aligned} \tag{30}$$

where $(\cdot)_+ = \max\{\cdot, 0\}$. Namely, on every positive time period that contributed to the customer's purchase, we select the top positive and present interaction advertisement type along with it's corresponding weight $\boldsymbol{\omega}_j$. That interaction gets placed in the customer's chain $\boldsymbol{C}^i$ and the corresponding weight is placed in the corresponding customer attribution vector $\boldsymbol{A}^i$ which is normalized to $\tilde{\boldsymbol{A}}^i$. All strictly positive elements of $\boldsymbol{C}^i$ and $\tilde{\boldsymbol{A}}^i$ constitute the customers chain. One may be interested in constructing a more complete chain with multiple advertisements per positive touch-point. In which case we define a more general K interaction-per-touch chain that is constructed as

$$\begin{aligned}
S^i &= \{(\omega_j \mathbb{I}\{\boldsymbol{x}_t^{j,i} > 0\})_+ : j = 1, \ldots, d\} \\
\boldsymbol{C}_{(k)}^i &= \left(z_t^i \cdot \arg \mathrm{TopK}(S^i) : t = 1, \ldots, T\right) \in \{1, \ldots, d\}^{KT} \\
\boldsymbol{A}_{(k)}^i &= \left(z_t^i \cdot \mathrm{TopK}(S^i) : t = 1, \ldots, T\right) \in \mathbb{R}^{KT}, \quad \tilde{\boldsymbol{A}}_{(k)}^i = A_{(k)}^i / \|A_{(k)}^i\|_2,
\end{aligned} \tag{31}$$

where the $\mathrm{TopK}(S)$ function returns the top K elements of a real-valued set S and $\arg \mathrm{TopK}(S)$ returns the indices corresponding to the top K elements of the set. Such a chain would enable more complex analysis of attribution as a multi-dimensional quantity at each touch point.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction are limited in scope to the paper's contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Core assumptions are contained in the main body and proofs are available in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we provide experimental details in 4 and appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We provide the code necessary to reproduce the main numerical experiments on synthetic data. Due to restrictions associated with the financial-services collaboration, we cannot publicly release the dataset required for section 5.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we provide experimental details in 4 and appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We perform appropriate statistical tests and report the results in section 4 and section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the appropriate computational resources in section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, we adhere to the NeurIPS code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The societal impacts of our work is limited as its primarily theoretical in nature. We report these limited potential societal harms in section 6.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any new high-risk data or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, our work uses open sources and the original owners are properly attributed. The proprietary dataset used in Section 5 will be properly attributed after acceptance to avoid revealing identifying information during the double-blind-review.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We only release code which is properly documented.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not include human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not include human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLM were not used as a core component of the work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.