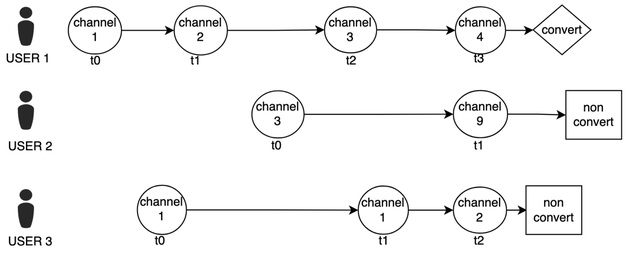
NEURAL ODE FOR MULTI-CHANNEL ATTRIBUTION

Yudi Zhang, Oshry Ben-Harush, Siyu Zhu, Xin Liang

Amazon Web Services, Seattle, USA

Introduction

Multi-Touch Attribution (MTA) plays a crucial role in both marketing and advertising, offering insight into the complex series of interactions within customer journeys during transactions or impressions. This holistic approach empowers marketers to strategically allocate attribution credits for conversions across diverse channels, not only optimizing campaigns but also elevating overall marketplace strategies. Traditional methods like first-touch and last-touch oversimplify the problem, while existing MTA models have various drawbacks in capturing nuanced interactions and handling non-uniform time intervals. Acknowledging the irregular time series nature of customer journey data (customer journey figure given below), to address these limitations, the paper introduces a novel approach using ODE-LSTM (Ordinary Differential Equation) [5] networks with an attention mechanism. This approach can handle irregular time gaps in customer journey data and is shown to outperform other MTA methods, particularly when time intervals are not excessively irregular. While its performance declines with increasing irregularity, the ODE-LSTM approach excels in estimating attributions compared to alternative approaches.



Model

Let $\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N)$ denote the input sequence data, where each $\mathbf{X}_i = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L)$ represents a sequence with length L and $\mathbf{x}_l$ denotes the features at the lth location. Let $\mathcal{Y} = (y_1, y_2, \ldots, y_N)$ be the class of the sequential data.
Follow [1] and [5], we use autoregressive modeling with ODE-LSTM with an additional attention layer to model the customer journey seuqences.
Assume each input data $\mathbf{x}_l$ is associated with an timestamp t_l and denote hidden states as well as memory cell as $\mathbf{h}_l$ and $\mathbf{m}_l$. The ODE-LSTM algorithm follows:

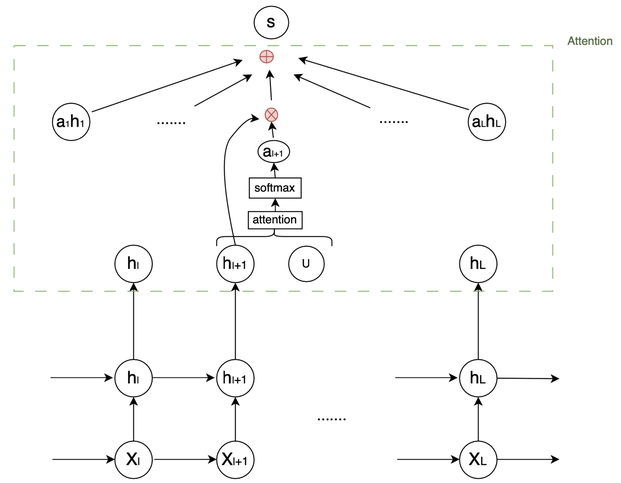
$$\mathbf{h}'_l, \mathbf{m}_l = \text{LSTMCell}\left(\mathbf{m}_{l-1}, \mathbf{h}_{l-1}, \mathbf{x}_l\right),$$
$$\mathbf{h}_l = \text{ODESolve}\left(f_\theta, \mathbf{h}_{l-1}, \mathbf{h}'_l, (t_{l-1}, t_l)\right),$$

where the function f_θ specifies the dynamics of the hidden state, using a neural network with parameters θ.
In order to obtain the attribution, the above hidden states are further feed to an attention layer to identify pivotal touchpoints contributing to conversions. Subsequently, we consolidate the representations of these significant touchpoints, creating a comprehensive context vector.

$$\begin{aligned} \mathbf{v}_l &= \tanh\left(\mathbf{W}\mathbf{h}_l\right) \\ a_l &= \frac{\exp\left(\mathbf{v}_l^T \mathbf{u}\right)}{\sum_l \exp\left(\mathbf{v}_l^T \mathbf{u}\right)} \\ \mathbf{s} &= \sum_l a_l \mathbf{h}_l \end{aligned} \tag{1}$$

The hidden states $\mathbf{h}_l$ are feed through a one-layer multilayer perceptron (MLP) to get $\mathbf{v}_l$, where $\mathbf{W}$ is a learnable matrix. Then, we measure the importance of the touchpoint by assessing the similarity of $\mathbf{v}_l$ with the vector $\mathbf{u}$ and obtain a normalized importance weight a_l through a softmax function. It is noteworthy that, by design, $a_l > 0$. This construction offers the advantage that the contribution of every touchpoint is always positive. Afterward, we compute the vector $\mathbf{s}$ as the weighted sum of touchpoint representations based on the non-negative weights.

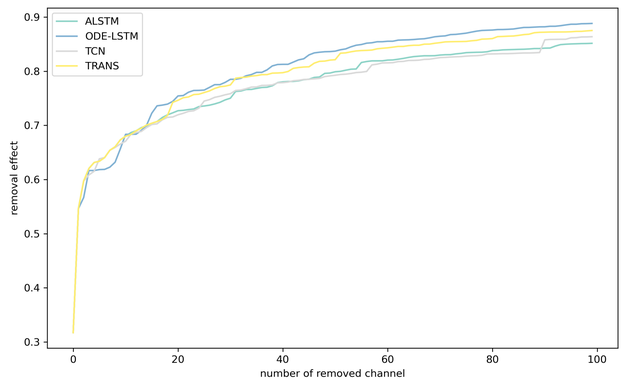
Essentially, $\mathbf{s}$ is the convex combination of all $\mathbf{h}_l$. $\mathbf{u}$ can be seen as a high-level representation of a fixed sequence. We can customize this attribution model by imposing constraints on $\mathbf{u}$ based on domain knowledge about touchpoint importance, it can either be kept fixed or initialized randomly and jointly learned during the process. In our modeling, we adopt the latter approach.
In MTA problem, customer journeys are categorized into positive (leading to conversions) and negative (not leading to conversions). This problem can be treated as binary classification in the transformed journey vector space $\mathbf{s}$, which combines hidden outputs and attention weights. Thus, we optimize a cross-entropy loss to train the model. Up to this point, we have the estimated conversion probability $p(y|\mathbf{X}_i)$ and the attention scores $\mathbf{a}$. With these outcomes, we can inherently allocate attribution to channels at each touchpoint l. As we want to estimate the impact of each channel on successful conversions, our calculations exclusively focus on customers who have achieved successful conversions. The total attribution of a channel is the accumulative sum of the touchpoint attention scores if that touchpoint visit that channel.

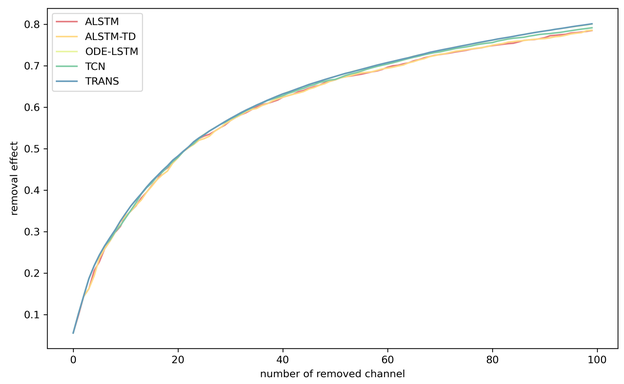


Results

Below table demonstrates the model performance in terms of the AUC and PRAUC. On the data (Criteo) that the time scale are relative small and most time differences are valid, ODE-LSTM performs the best, however, as the time difference goes large and most differences are 0 as in the marketing signup data, ODE-LSTM was beat by ALSTM and TCN. The reason might be the ODE-LSTM faces challenges due to its focus on continuous transitions. ALSTM and TCN are more robust in capturing patterns in such scenarios. However, the simple Transformer model is the worst on the both data. It might be because its lack of inherent temporal understanding, which means it may not capture important temporal dependencies in the data. For such irregular time intervals or missing data points data, Transformers may not handle such irregularities well, and additional preprocessing or better time encoding is often needed.
Additionally, we introduce a novel metric called AURE (Area Under Removal Effects). AURE tracks the cumulative impact on conversion probabilities by successively removing channels based on the ranked attributions. It appears that ODE-LSTM and Transformers are highly consistent on Criteo data. The results are shown in right figures. The top 100 removal effects for ODE-LSTM and Transformer are 0.801 and 0.800 respectively, indicating a slight better performance of ODE-LSTM.

Models	Criteo		Marketing Sign-up	
	AUC	PRAUC	AUC	PRAUC
ODE-LSTM	**0.9832**	**0.9293**	0.9200	0.8507
ALSTM	0.9827	0.9286	**0.9710**	**0.9292**
TCN	0.9817	0.9238	0.9629	0.9079
TRANS	0.9813	0.9226	0.9262	0.8394





Remarks

We proposed a ODE-LSTM combined with an attention mechanism to estimate the attribution in MTA problem. ODE-LSTM is not necessary the best the model if simply comparing the AUC and PRAUC, and ALSTM is the most robust method for predicting conversion. However, by comparing the proposed AURE metrics, ODE-LSTM gives the best results. Although Neural ODE handles continuous data, e.g. irregularly-sampled data or test-time sampling shift automatically and are mathematically tractable to analyze. They are extremely slow at both training and inference. To further enhance the performance of ODE-related methods, a potential avenue for improvement lies in refining the handling of time dynamics inherent in the attribution problem. Or we could try newly developed methods, for example NCDE [4] and State Space Model [3, 2].

References

[1] Ricky T. Q. Chen et al. “Neural Ordinary Differential Equations”. In: arXiv:1806.07366 (2019). DOI: 10.48550/arXiv.1806.07366. arXiv: 1806.07366[cs,stat].
Albert Gu and Tri Dao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. 2023. arXiv: 2312.00752 [cs.LG].
Albert Gu, Karan Goel, and Christopher Ré. *Efficiently Modeling Long Sequences with Structured State Spaces*. 2022. arXiv: 2111.00396 [cs.LG].
Patrick Kidger et al. *Neural Controlled Differential Equations for Irregular Time Series*. 2020. arXiv: 2005.08926 [cs.LG].
Mathias Lechner and Ramin Hasani. *Learning Long-Term Dependencies in Irregularly-Sampled Time Series*. 2020. arXiv: 2006.04418 [cs.LG].