Training Machine Learning Models With Causal Logic

Ang Li, Suming J. Chen, Jingzheng Qin, Zhen Qin
Google
Mountain View, California, USA
{angliangli,suming,jzq,zhenqin}@google.com

ABSTRACT

Machine-learning (ML) models are ubiquitously used to make a variety of inferences, a common application being to predict and categorize user behavior. However, ML models often suffer from only being exposed to biased data – for instance, a search ranking model that uses clicks to determine how to rank will suffer from position bias. The difficulty arises due to user feedback only being observed for one treatment and not existing counterfactually for other potential treatments. In this work, we discuss a real-world situation in which a binary classification model is used in production in order to make decisions about how to treat users. We introduce the model and discuss the limitations of our modeling approach. We show that by using unit selection criterion we can do a better job classifying users. Following, we propose a causal modeling method in which we can take the existing data and use it to derive bounds that can be used to modify the objective function in order to incorporate causal learning into our training process. We demonstrate the effectiveness of this approach in a real-world setting.

CCS CONCEPTS

• **Mathematics of computing → Causal networks**.

KEYWORDS

counterfactual learning

ACM Reference Format:
Ang Li, Suming J. Chen, Jingzheng Qin, Zhen Qin. 2020. Training Machine Learning Models With Causal Logic. In *Proceedings of ACM (IID 2020).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Correctly predicting and categorizing user behavior is critical in many industry areas. For example, in online advertising [4, 11, 15, 18], there are companies whom are interested in identifying users who would only click on an advertisement if and only if the said advertisement is highlighted. Another example lies in customer relationship management [7, 16], where it's desirable to predict which customers are about to churn but are likely to change their minds if enticed towards retention. A common difficulty in predicting and categorizing behavior arises due to user feedback only being observed for one treatment and not defined counterfactually in terms of what the individual would do under hypothetically unrealized conditions. For example, if we see that a user clicked on a promoted advertisement, we don't know if they would have clicked on that advertisement had it not been highlighted.

To better categorize user behavior for prediction, it's useful to classify individual behavior into four response types, labeled complier, always-taker, never-taker, and defier [2, 3]. *Compliers* are individuals who would respond positively had they been encouraged and negatively if not encouraged. *Always-takers* and *never-takers* always respectively respond positively/negatively whether or not they get encouragement. *Defiers* are individuals who response negatively if encourage and positively if not encouraged.[1] Commonly, unit selection is used to target compliers since that would result in the most effective treatment.

[9] treats the unit selection problem using the structural causal model (SCM) [14] in order to take into account the counterfactual nature of the desired behavior, similar to [5, 6], and found that unit selection can derive selection criteria that allows for ways to decide which group to expose to a treatment in order to yield greater benefit than standard methods. In the work of [9], they found that the unit selection problem entails two sub-problems of evaluation and search, and propose a solution for the evaluation sub-problem – theoretically useful, but often impractical in a real world setting where treatments need to be made at the individual level.

In this work, we propose a method in which the search sub-problem can be approximately solved by computing a group-wise attribute (e.g. a label for a group of users) with causal unit-selection derived bounds. In other words, we modify the learning objective function in order to train a better performing decision-making model by providing counterfactual information to the training process. We applied this methodology to an application where the goal is to balance search quality with resource utilization, and saw a significant improvement over models trained with the baseline procedure.

At a high level, our work deals with the bias present in the implicit feedback, similar to the rich literature in unbiased and counterfactual learning [1, 8, 10]. However, our work is unique that 1) unlike work that focuses on presentation bias (e.g. position bias [8]) that is constant for all users, we focus on the model-induced bias when targeting individual users, and 2) unlike existing counterfactual learning methods [12] that depend on propensity score weighting (typically using online randomization), to the best of our knowledge, our work is the first to apply a principled causal logic for personalized decision making under purely observational data.

This paper is structured as follows: we first provide background knowledge on the motivating real-world example and discuss how

IID 2020, April 2020, Taipei

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] For instance, a user who would click an advertisement if only it wasn't promoted.

unit selection can help. Next, we present background for the counterfactual logic associated with SCM and show how it can be used to represent the unit selection problem. Following, we present our novel methodology of modifying the training objective to incorporate unit-selection derived selection criteria into the training process. We conclude with experimental results of our approach.

2 BACKGROUND

2.1 Real-world Motivating Example

There is a search bar in GMail that shows "instant" results, where every keypress may trigger different email search results to render. In addition to this, the search bar may also show search results from users' Google Drive accounts, as seen in Figure 1.

Since not all users who use GMail necessarily will use Drive, a decision to make is whether or not to display the Drive section, as if it's not displayed we can suppress that additional call to Drive to prevent wasting resources. For consistent user experience, after deciding whether or not to suppress that section, we lock that decision for a window of time. The more users we enable the section for, the more *effective* the search system is, whereas the fewer users we enable the section for, the more *efficient* our search system becomes, thus putting us in a position to trade off effectiveness and efficiency, similar to [17].

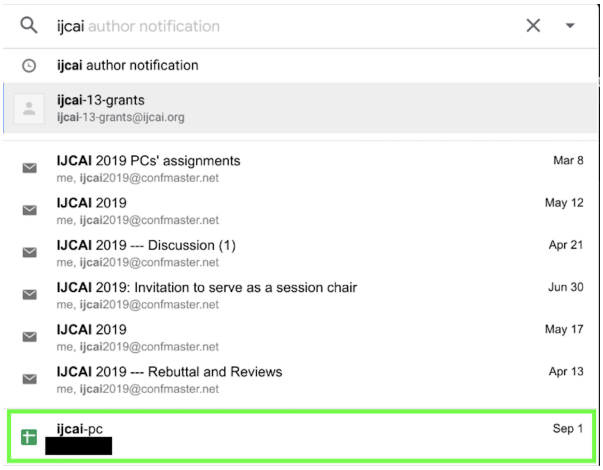

Figure 1: Example search bar with suggestions and search results. Results from both GMail and Drive are shown, with Drive results in a "Drive section" bounded in the green box.

Initially, there was a heuristic method set up to enable/disable the Drive section based on statistics of user's past activities (how often they click on the Drive section, how often they click on Drive documents). We then developed a machine-learned (ML) model in order to decide whether or not to suppress the Drive section.

At a high level, we model this as a *binary classification* problem where we are determining which users should and shouldn't have their Drive section suppressed, with an objective that considers both click and resources-used (using the logged number of keypresses per-search as a proxy). Each training data point is at user level, with the input features being the frequency of user activities (e.g. the number of views, edits, creates on Drive documents). At serving time, our model needs to decide whether or not to suppress the user's Drive section. Note that because of the method in which initial heuristic model was deployed, offline training and evaluation data is *biased* – i.e. if the heuristic rule turns the Drive section off for a user, we will not get any clicks on the Drive section, which is an important factor on whether to enable/disable the Drive section.

To get labels for the training data, we consider `click` and `keypress`, respectively the number of clicks on Drive section and number of keypresses from the user in the searchbar. The per-user objective function, given positive hyperparameters α and β is:

$$((\alpha * \text{click} - \text{keypress}) > 0 ? 1 : 0) \tag{1}$$

which we treat as the objective function for binary classification, where each example is weighted by:

$$\beta * abs(\alpha * \text{click} - \text{keypress}) \tag{2}$$

This results in an intuitive setting: if a user has many clicks but a low number of key-presses, then the weight in Equation 2 is high and the label is positive, which means it's especially important for the model to predict this user as positive during training (turn on the Drive section). If a user does not have many clicks but has a lot of keypresses, the weight is also high and label is negative, the model is more likely to disable the Drive section for the user, saving resources while not losing clicks.

This model led to performance gains but suffered from the bias problem. There are methods which allow for running experiments, including with contextual bandit approaches [11], in order to collect unbiased data to train a fairer model. These methods typically need some exploration (perhaps via randomization) to have a newly trained model make a different decision. However, these existing methods are often not practical since they would harm user experience (exposing to multiple confusing treatments).

This motivates us to look for alternative methods in order to more accurately predict whether or not a user really needs the Drive section enabled. This can be determined with the unit selection problem discussed in the introduction, where if we are able to infer the latent user type, we can have an additional dimension that our model can be trained on in order to make more accurate predictions.

2.2 Counterfactual Logic

In this section, we review the counterfactual logic [5, 6, 14] associated with Pearl's SCM, which is used in the remainder of this paper. The basic counterfactual statement associated with model M is denoted by $Y_x(u) = y$, and stands for: "Y would be y had X been x in unit $U = u$,". Let M_x denote a modified version of M, with the equation(s) of set X replaced by $X = x$ (i.e., all edges that go into X have been removed). Then, the formal definition of the counterfactual $Y_x(u)$ is as follows:

$$Y_x(u) \triangleq Y_{M_x}(u) \tag{3}$$

In words, the counterfactual $Y_x(u)$ in model M is defined as the solution of Y in the "modified" submodel M_x. In [5, 6], a complete axiomatization of structural counterfactuals, embracing both recursive and nonrecursive models, is given.

Equation (3) implies that the distribution $P(u)$ induces a well-defined probability for the counterfactual event $Y_x = y$, written

as $P(Y_x = y)$, which is equal to the probability that a random unit u would satisfy the equation $Y_x(u) = y$. Therefore, the probability of the event "Y would be y had X been x", $P(Y_x = y)$, is well-defined and $P(Y_x = y)$ is also equivalent to $P(Y = y|do(X = x))$. $P(Y = y|do(X = x))$ can be interpreted as experimental data [13]. With the same reasoning, the SCM model assigns a probability to every counterfactual or combination of counterfactuals that are defined using the variables in SCM.

Using the above formal language for the counterfactual expression, all events involving a counterfactual scenario can be well defined, because the event represented by the subscript does not actually occur. For example, $P(Y_x = y|X = x')$ defines the probability of the event "Y would be y had X been x if we observed $X = x'$" (note that x and x' are counterfactual scenarios), $P(Y_x = y, Y_{x'} = y')$ defines the probability of the event "Y would be y had X been x and Y would be y' had X been x'" (note that x and x' is a counterfactual scenario; y and y' is a counterfactual scenario), and $P(Y_x = y|X = x', Y = y')$ defines the probability of the event "Y would be y had X been x, if we observed $X = x'$ and $Y = y'$".

In the rest of the paper, let y denote that the user would click the Drive link and y' denotes that the user would not click the Drive link. Let x denote that the Drive section is shown to the user and x' denotes that the Drive section is not shown to the user. As such, we use y_x to denote the event $Y_x = y$ (user would click if Drive section was shown), $y_{x'}$ to denote the event $Y_{x'} = y$ (user would click if Drive section wasn't shown – meaning the user had to go to Drive app to click), y'_x to denote the event $Y_x = y'$ (user would not click if Drive section was shown), and $y'_{x'}$ to denote the event $Y_{x'} = y'$ (user would not click if Drive section wasn't shown).

3 CAUSAL METHODOLOGY

3.1 Motivation

Unit selection based on counterfactual logic has been proven in [9] to be effective in the unit selection problem, where a decision maker must determine which group of users should receive an experimental treatment. We draw inspiration from this to derive a new method that allows for *extending* a training objective function to incorporate unit selection counterfactual logic.

With the previously introduced notation and groups defined in [2, 3], we have the following individuals to consider for our decision-making problem:

- Complier ($y_x, y'_{x'}$): Users who would access Drive files if and only if the Drive section was shown.
- Always-taker ($y_x, y_{x'}$): Users who would access Drive files whether or not the Drive section was shown.
- Never-taker ($y'_x, y'_{x'}$): Users who would not access Drive files whether or not Drive section was shown.
- Defier ($y_{x'}, y'_x$): users who would access Drive files if and only if Drive section wasn't shown.

By modeling users this way, we can see that it's optimal to provide the Drive section to the always-taker (as the first priority) and then show Drive section to compliers as a short-cut (as second priority if there are enough resources). Never-takers should clearly never be shown the Drive section, and defiers in this scenario we believe to be not practical or necessary to consider.[2]

Note that the previously introduced ML model was not trained on this kind of counterfactual information. The key takeaway here: although we can never know the response type for a particular user, we can bound their probabilities if we have experimental and observational data $P(y_x)$, $P(y_{x'})$, and $P(x, y)$, and this bound can be incorporated into the naive objective function of Equation 1.

3.2 Causal Model

Our objective is to find a subset of users that maximizes the benefit associated with the resulting mixture of compliers, defiers, always-takers, and never-takers. Our ideal objective, then, should be

$$\begin{aligned} &\alpha * \text{click} - \text{keypress} + \beta * P(y_x, y'_{x'}) + \gamma \\ &\quad * P(y_x, y_{x'}) + \theta * P(y'_x, y'_{x'}) + \delta * P(y'_x, y_{x'}) \end{aligned} \tag{4}$$

Note that in this application, we set

$$\beta > \gamma > 0 = \delta > \theta$$

to indicate that always-taker is the first priority and complier is the second priority. $\delta = 0$ is set to express that for this scenario we don't consider defiers to be a valid group to consider. (for the most accuracy model, we should set δ to be negative) We would set $\theta < 0$ to penalize never-takers.

3.3 Simplified Causal Model

Equation 4 is the ideal modeling objective to train our model on since it allows us to exactly assign utility to each type of user we encounter. Unfortunately, the type of the user is latent and only can be estimated with bounds, as is discussed in [9]. Theoretically, we could infer the user type by running a variety of experiments to disable/enable the Drive section for a user in order to measure the effect. Practically, we have limited data availability which constrains our ability to infer user type.

Instead of attempting to infer the above terms (e.g. $P(y_x, y'_{x'})$), we focus our efforts on a more manageable term: $P(y_x|y')$. In other words, if we observed that a user has no Drive file clicks, the probability that this user would click on a Drive result if we *had* shown the Drive section. We could then use this proxy objective function:

$$\alpha * \text{click} - \text{keypress} + \beta * P(y_x|y') \tag{5}$$

Including the term $P(y_x|y')$ has two benefits:

- we will only turn off users who previously had the Drive section enabled and did not click there.
- users with newly-enabled Drive section have higher probability to click there.

3.3.1 Computing a practical bound. Since $P(y_x|y')$ is still intractable and cannot actually be computed, we need to find a way to evaluate this term by the available data.

[2]Previous work which classified users into these response types used examples of targeting advertisement to a user – defier in this setting would make more intuitive sense, but can be ignored in this scenario.

We can bound $P(y_x|y')$ as following:

$$P(y_x|y') = \frac{P(y_x, y')}{P(y')} \geq \frac{[P(y_x) - P(y)]}{P(y')}$$

Although we have the data for $P(y_x)$ from experimental data, we do not have $P(y)$, as it is biased observational data. We can also infer $P(y)$ with a proxy variable w – denoting whether or not a user has Drive activity. We have $P(y) = P(w, y) + P(w', y) = P(w, y)$, because if we have Drive section click, we must have Drive activity, thus:

$$\begin{aligned} P(y_x|y') &\geq \frac{[P(y_x) - P(y)]}{P(y')} \\ &= \frac{[P(y_x) - P(y, w)]}{P(y')} \\ &\geq \frac{[P(y_x) - P(w)]}{1} \end{aligned} \tag{6}$$

3.3.2 Group-level to user-level objective function. $P(y_x|y')$ is clearly a group-level term, i.e. for a group of users, we can evaluate this term among the group, and all users in the group have the same value of $P(y_x|y')$. Therefore, given some training data with n users, we can only identify a group of $m < n$ users that maximize the objective function. However, in an ML model that is serving live traffic, we need to be able to have a user-level decision for whether or not to serve the Drive section, rendering the group-level objective function we derived to be non-trivial to apply.

Given that there are 2^n subsets of the users, there is at least one subset that maximize $P(y_x|y')$. We postulate that whether or not a user is in the desired subset is partially determined by the users attributes. Our methodology consists of finding the subset of users to maximize the group-level objective function defined in Equation 6, and then taking their group membership into account into the training loss function. In other words, if we provide a 0/1 label indicating whether the user is in the desired subset to the ML model, the training process would have additional information to be able to exploit the relation between the user attributes with this counterfactual logic.

Therefore, we modified our model training process to use the following objective function in place of Equation 1:

$$\alpha * \text{click} - \text{keypress} + \beta * \Phi[\text{member}] \tag{7}$$

where Φ[member] is an indicator function that returns 1 iff the user is in the group that maximize $P(y_x|y')$ and 0 otherwise.

3.3.3 Group determination method. We need a simple closed-form labeling method that can determine a group of users that comes close to maximizing Equation 6. Let a be the number of users who have a Drive section click, b be the number of users with the Drive section displayed, c be the number of users who have Drive activity, and n be the number of users. Then we have:

$$P(y_x) - P(w) = \frac{a}{b} - \frac{c}{n}$$

The key insight here is that whether or not we add a user with Drive section click and Drive activity to the target group depends on the relation of $\frac{a}{b}$ to $\frac{c}{n}$. We propose the group determination method shown in Figure 2, as it is the simplest condition to make

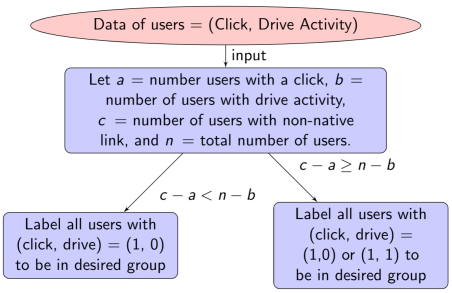

Figure 2: Group determination method

$P(y_x) - P(w)$ larger, because if (click,drive) = $(1, 0)$, we have the following:

$$\begin{aligned} &(P(y_x) - P(w))_{\text{new}} - (P(y_x) - P(w))_{\text{old}} \\ =&\frac{a+1}{b} - \frac{c+1}{n+1} - (\frac{a}{b} - \frac{c}{n}) \\ =&\frac{n^2 + n - nb + c}{bn(n+1)} \geq 0 \end{aligned}$$

and if (click,drive) = $(1, 1)$ and $(c - a \geq n - b)$, we have the following:

$$\begin{aligned} &(P(y_x) - P(w))_{\text{new}} - (P(y_x) - P(w))_{\text{old}} \\ =&\frac{a+1}{b+1} - \frac{c+1}{n+1} - (\frac{a}{b} - \frac{c}{n}) \\ =&\frac{b-a}{b(b+1)} - \frac{n-c}{n(n+1)} \geq 0 \end{aligned}$$

3.4 Experimental Results

We ran experiments to compare the standard ML model that is used in production against the causal-trained model. We trained a model using the objective function defined in Equation 7 and compared it to the normal ML-model (which was trained with the objective function defined in Equation 1). We ran a week-long experiment and found that the causal-trained model led to to a statistically significant 9.15% increase in Drive section click-through rate (CTR) with a non-significant increase of 1.4% in terms of resource usage.

This is a significant improvement – to contrast this with previous improvements, we previously saw that the initial deployment of the first machine-learned model led to 2.5% relative increase in Drive section click-through rate (CTR) with a further saving of -6.5% resource saving.

4 CONCLUSION AND FUTURE WORK

We introduced a novel method in which we can incorporate causal information into the model training process for a real-world decision making problem. This method allows us to deal with bias in implicit feedback without needing to do any randomization. Additionally, we demonstrate that we see empirical wins from using this method in live traffic experiments. For future work, we plan to compare against other work that focuses on learning with bias in implicit feedback and prove theoretically that such methods are robust by running various experiments over both synthetic and actual data in different domains.

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 5–14.

[2] Joshua D Angrist, Guido W Imbens, and Donald B Rubin. 1996. Identification of causal effects using instrumental variables. *Journal of the American statistical Association* 91, 434 (1996), 444–455.

[3] Alexander Balke and Judea Pearl. 1997. Bounds on treatment effects from studies with imperfect compliance. *J. Amer. Statist. Assoc.* 92, 439 (1997), 1171–1176.

[4] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research* 14, 1 (2013), 3207–3260.

[5] David Galles and Judea Pearl. 1998. An axiomatic characterization of causal counterfactuals. *Foundations of Science* 3, 1 (1998), 151–182.

[6] Joseph Y Halpern. 2000. Axiomatizing causal reasoning. *Journal of Artificial Intelligence Research* 12 (2000), 317–337.

[7] Shin-Yuan Hung, David C Yen, and Hsiu-Yu Wang. 2006. Applying data mining to telecom churn management. *Expert Systems with Applications* 31, 3 (2006), 515–524.

[8] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 781–789.

[9] Ang Li and Judea Pearl. 2019. Unit selection based on counterfactual logic. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI'19)*. AAAI Press, 1793–1799.

[10] Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. 2014. Counterfactual estimation and optimization of click metrics for search engines. *arXiv preprint arXiv:1403.1891* (2014).

[11] Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. 2015. Counterfactual estimation and optimization of click metrics in search engines: A case study. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 929–934.

[12] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2016. Hyperband: A novel bandit-based approach to hyperparameter optimization. *arXiv preprint arXiv:1603.06560* (2016).

[13] Judea Pearl. 1995. Causal diagrams for empirical research. *Biometrika* 82, 4 (1995), 669–688.

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

[15] Wei Sun, Pengyuan Wang, Dawei Yin, Jian Yang, and Yi Chang. 2015. Causal Inference via Sparse Additive Models with Application to Online Advertising.. In *AAAI*. 297–303.

[16] Chih-Fong Tsai and Yu-Hsin Lu. 2009. Customer churn prediction by hybrid neural networks. *Expert Systems with Applications* 36, 10 (2009), 12547–12553.

[17] Lidan Wang, Jimmy Lin, and Donald Metzler. 2011. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 105–114.

[18] Jun Yan, Ning Liu, Gang Wang, Wen Zhang, Yun Jiang, and Zheng Chen. 2009. How much can behavioral targeting help online advertising?. In *Proceedings of the 18th international conference on World wide web*. ACM, 261–270.