Deconfounding User Satisfaction Estimation from Response Rate Bias

Konstantina Christakopoulou, Madeleine Traverse, Trevor Potter, Emma Marriott, Daniel Li, Chris Haulk, Ed H. Chi, Minmin Chen
{konchris,mtrav,tpotter,emarriott,danielfengli,haulk,edchi,minminc}@google.com
Google, Inc
Mountain View, CA

ABSTRACT

Improving user satisfaction is at the forefront of industrial recommender systems. While significant progress has been made by utilizing logged implicit data of user-item interactions (i.e., clicks, dwell/watch time, and other user engagement signals), there has been a recent surge of interest in measuring and modeling user satisfaction, as provided by orthogonal data sources. Such data sources typically originate from responses to user satisfaction surveys, which explicitly ask users to rate their experience with the system and/or specific items they have consumed in the recent past. This data can be valuable for measuring and modeling the degree to which a user has had a satisfactory experience on the recommendation platform, since what users *do* (engagement) does not always align with what users *say they want* (satisfaction as measured by surveys).

We focus on a large-scale industrial system trained on user survey responses to predict user satisfaction. The predictions of the satisfaction model for each user-item pair, combined with the predictions of the other models (e.g., engagement-focused ones), are fed into the ranking component of a real-world recommender system in deciding items to present to the user. It is therefore imperative that the satisfaction model does an equally good job on imputing user satisfaction across slices of users and items, as it would directly impact which items a user is exposed to. However, the data used for training satisfaction models is biased in that users are more likely to respond to a survey when they will respond that they are more satisfied. When the satisfaction survey responses in slices of data with high response rate follow a different distribution than those with low response rate, response rate becomes a confounding factor for user satisfaction estimation.

We find positive correlation between response rate and ratings in a large-scale survey dataset collected in our case study. To address this inherent response rate bias in the satisfaction data, we propose an inverse propensity weighting approach within a multi-task learning framework. We extend a simple feed-forward neural network architecture predicting user satisfaction to a shared-bottom multi-task learning architecture with two tasks: the user satisfaction estimation task, and the response rate estimation task. We concurrently train these two tasks, and use the inverse of the predictions of the response rate task as loss weights for the satisfaction task to address the response rate bias. We showcase that by doing this, (i) we can accurately model whether a user will respond to a survey, (ii) we improve the user satisfaction estimation error for the data slices with lower response rate while not hurting slices with higher response rate, and (iii) we demonstrate in live A/B experiments that applying the resulting satisfaction predictions to rank recommendations translates to higher user satisfaction.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning approaches**; **Neural networks**; • **Information systems** → **Recommender systems**.

KEYWORDS

Inverse propensity weighting, User satisfaction, Response rate bias

ACM Reference Format:
Konstantina Christakopoulou, Madeleine Traverse, Trevor Potter, Emma Marriott, Daniel Li, Chris Haulk, Ed H. Chi, Minmin Chen. 2020. Deconfounding User Satisfaction Estimation from Response Rate Bias. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3383313.3412208

1 INTRODUCTION

Recommender systems have historically focused on user engagement-related metrics, such as click through rate and consumption time (e.g., dwell time for news articles or videos) [25]. These metrics assume that the implicit user feedback is indicative of how much they value their user experience. This assumption has been challenged recently [6, 9, 17, 24]. In order to better understand and improve user experience on the platform, recommender systems have started to rely more on surveys in which users are explicitly asked to rate their experience on the platform, or specific items they have recently consumed [7, 8, 13, 16]. For simplicity, we will focus on the latter kind of surveys, which request explicit point-wise feedback on items. We will refer to these surveys as *satisfaction surveys* throughout the paper.

Compared with implicit feedback, responses to satisfaction surveys on the other hand are scarce. First, it is disruptive to ask users about every item they recently consumed. Second, survey response rate can be very low in an environment where primary user intention is to consume content, not to provide feedback. In our systems, the measured survey response rate is roughly 2%. As a result, we only have access to a small amount of survey responses covering

RecSys '20, September 22–26, 2020, Virtual Event, Brazil

ACM ISBN 978-1-4503-7583-2/20/09.
https://doi.org/10.1145/3383313.3412208

an extremely small fraction of the user-item pairs on the platform. Imputation models are therefore required to infer user satisfaction on all user-item pairs on the platform for scoring/ranking, from the small set of collected survey responses. We will refer to these models as *satisfaction models*.

Building such a model to estimate user satisfaction can be quite challenging due to the nature of the data itself:

- **Sparsity:** Although implicit interaction data can be abundant for estimating engagement proxy metrics, by definition we have access to orders of magnitude fewer data for estimating user satisfaction;
- **Response Bias:** On top of the low response rate, a user's satisfaction level with an item (as measured by the provided response), tends to correlate with the fact that the user decided to respond to a survey about this item in the first place. A user tends to respond to a survey only when they feel very strongly about the item asked. This confounding ultimately leads to imbalanced data, with most labels indicating being satisfied with the items presented in surveys.

This paper focuses on the latter challenge, i.e, response bias [18], with the goal of decoupling true user satisfaction with an item, from the confounding factor of deciding to respond to the survey. Similar bias has been studied in econometric models relying on survey data [1], in which the sample surveyed is not representative of the entire population. One widely used technique to address such bias is to use *inverse propensity weighting* to change the data distribution to reflect that of the underlying population intended for the study [10, 11, 15]. It involves creating a propensity model which estimates the probability of an observed example occurring in the data, and then weighs the example by the inverse of this propensity score when training the model estimating the target treatment effect. This technique creates a pseudo-population where the treatment is independent of the measured confounders. Translating this to our setting, the measured confounder is the deciding to respond variable, and we aim to have a dataset of survey responses irrespective of the response rate. We therefore build a propensity model estimating the probability of observing a survey response for a user-item pair under certain context, and then use the inverse of these probabilities as weights to adjust the importance of each survey response example in the satisfaction models.

Although this technique has been widely adopted for correcting selection bias [20, 22], position bias [23], or exposure bias [14], and has shown promising results both in traditional econometrics and survey sampling [11], as well as more recently in learning-to-rank [12] and recommender systems [4, 19, 26], to the best of our knowledge, this is the first time to apply it for correcting for response rate bias, or as a means to improve user satisfaction. In particular, we offer the following contributions:

- **Response rate as a confounder for user satisfaction:** We showcase in a large-scale satisfaction dataset that the feedback users give in surveys on items they have consumed tends to correlate with response rate (Section 3).
- **User Satisfaction shared-bottom multi-task learning architecture:** We propose a novel architecture adapting a standalone satisfaction estimation model, to a two-headed shared-bottom architecture predicting (1) user satisfaction, and (2) tendency to respond, given the same latent representation of user-item-context for both tasks, but also utilizing task-specific parameters (Figure 3(b)).
- **Inverse Propensity Weighting Response Bias:** We utilize inverse propensity weighting on the predicted response probabilities for the training of the satisfaction head, which has not been applied to the best of our knowledge for addressing response rate bias in user satisfaction (Section 4). To address the low response rate challenge, we showcase how balancing the survey impression dataset containing labels of responses/non-responses, and calibrating the predicted probabilities, can help training stability and lead to accurate response rate predictions (Figure 4).
- **Satisfaction improvements:** We provide evidence from offline experiments that our approach indeed reduces estimation errors of user satisfaction across data slices with low estimated propensities to respond (Figure 5). We demonstrate through A/B experiments that applying predictions from the proposed model to rank top-K recommendations leads to an overall decrease of dissatisfying user experience. We also find that the improvement in satisfaction-related metrics is more accentuated for low-response data slices, as expected, underlining the importance of paying equal focus to data slices during training satisfaction models (Figure 6).

2 USER SATISFACTION ESTIMATION

We begin by providing some background on user satisfaction estimation, and introduce notation that will be used throughout this paper. We then provide details on the architecture of a satisfaction estimation model used in the large-scale commercial recommender system which serves as the case study for our work.

2.1 Background: User Satisfaction

Keeping users satisfied with the items they have engaged with is vital to the success of a recommender system [13]. For this, both measuring and modeling user satisfaction is imperative. Throughout the paper, we will assume that the ground truth of user satisfaction is measured directly from user-provided survey responses. These surveys are shown uniformly to all users, and ask users to rate on a scale how satisfying they found a sampled item from their recent engagement history. Given the sparsity of this data, as a relatively low percentage of user-item pairs are surveyed, with even fewer elicited a response, satisfaction models are required in order to impute satisfaction values for user-item pairs not surveyed or responded.

To set the context, we will assume for the remainder of this paper that the predictions of the satisfaction model are utilized to rank, along with the output of other (standalone or multi-task) engagement models, what should be the top-K recommendations for a user u at a context c, ranking items from a corpus $\mathcal{I}$ [5, 27]. To ensure equal user experience on the platform, we need to ensure the model's mistakes are not pronounced in certain data slices, for example, in slices with lower response propensities.

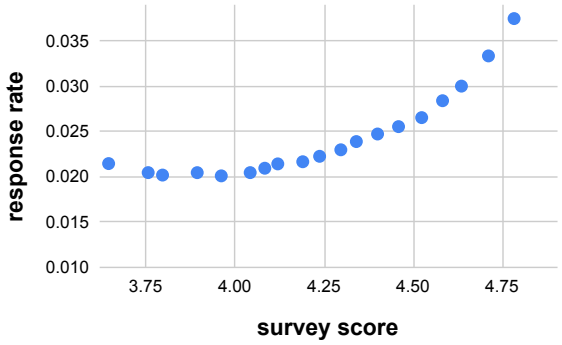

Figure 1: Average response rate (y-axis) vs. satisfaction level quantiles (x-axis).

2.2 Notation

We denote by $\mathcal{D}_r = \{(u, i, c, s(u, i, c))\}$ the dataset of responses to satisfaction surveys. We assume access to features $\mathbf{x}_u$, $\mathbf{x}_i$, $\mathbf{x}_c$ representing the user $u \in \mathcal{U}$, item $i \in \mathcal{I}$, and the context $c \in C$ respectively. The survey response s falls in a certain range specified by the feedback type elicited. For example, the user could be asked to rate the item in a scale from one to five. Let $\mathcal{D}_i = \{(u, i, c, s(u, i, c), r(u, i, c))\}$ be the dataset of survey impressions, which is a superset of survey responses – $\mathcal{D}_r \subset \mathcal{D}_i$. Here $r(u, i, c) \in \{0, 1\}$ denotes whether the user responded to the survey. If they did, the response $s(u, i, c)$ is recorded, which is otherwise unknown.

2.3 User Satisfaction Model trained on Survey Responses

Typically, user satisfaction models are trained on the dataset of survey responses, $\mathcal{D}_r$, corresponding to Figure 2(a). The objective of such a machine learning model is to learn parameters θ such that the prediction error between the predicted survey responses and the ground truth responses is minimized,

$$\mathcal{L}(\theta) = \sum_{(u,i,c)\in\mathcal{D}_r} \ell_r\left(s(u, i, c), \hat{s}(u, i, c; \theta)\right) \tag{1}$$

where ℓ_r denotes the loss used, and $\hat{s}$ is the imputed survey response. We can consider any loss function for ℓ. Commonly used loss functions include logistic loss when mapping the survey responses to a binary label by specifying a threshold of satisfaction, or square loss to predict the raw survey response value. In terms of the model parameterization to predict $\hat{s}$, we will assume a deep neural network architecture like the one illustrated in Figure 3(a). This architecture learns the non-linear function mapping of (user, item, context) representations to survey response values. The user is represented by dense and sparse features summarizing their recent activity and profile (i.e., indicating their tendency to interact with fresh content, their favorite topics, their interaction history length, location, etc.). The item is represented by features characterizing it, such as topic, item age, overall popularity. The context features can indicate the device the user is using, the time of the day, and so on. Each of these features is embedded/projected into dense vectors, with each feature embedding having potentially different dimensions. Concretely, the concatenation of the user embeddings $\mathbf{x}_u$, item embeddings $\mathbf{x}_i$, and context embeddings $\mathbf{x}_c$, is passed to a feed-forward architecture consisting of Rectified Linear Units (ReLU) layers. Finally, a linear layer (or one with logistic transformation, if binary labels are utilized) maps the output of the feed-forward network to the survey responses. Embeddings, along with all other model parameters are learned jointly through gradient back-propagation. Other model architectures can be explored; it is however not the scope of this work, and we refer readers to [16] for references on modeling choices. It is worth noting though that due to the data sparsity, highly parameterized architectures could lead to over-fitting. In Section 4, we build our method on top of this architecture, but our approach is agnostic to the underlying architecture choices.

3 A CAUSAL FRAMEWORK FOR USER SATISFACTION

The dataset $\mathcal{D}_r$ only provides us measurements of user satisfaction on items at certain contexts when users decide to respond, i.e., when $r(u, i, c) = 1$, but not on the others. Since the response $s(u, i, c)$ is not independent of $r(u, i, c)$ without knowing the underlying confounders, it creates a biased dataset commonly referred to as missing-not-at-random (MNAR) [21].

Formally, we can pose this in a causal framework, as illustrated in Figure 2(b). Given a survey impression, the user decides if they want to respond, and if they do ($r = 1$), we observe the survey response. Both variables of 'decide to respond' and 'survey response' have the common cause of confounders Z. Intuitively, variables affecting the inherent preference of the user to the item would affect both whether the user will respond, and the response value itself. As a result, the unobserved survey responses are not missing-at-random. We validate this point in Figure 1 by analyzing survey impressions data from an industrial recommender system. Slicing the survey impressions based on their satisfaction survey score, we can see that the higher this score, the higher the response rate, and vice-versa, making the case for the existence of response rate bias in real-world satisfaction data.

To address this confounding, ideally we would like access to some intervention, in which we externally set the decide-to-respond variable, i.e., do(Decide to respond=1), forcing users to always respond. This way, we can see in Figure 2(c), that the arrow from the latent confounder Z to the decide-to-respond variable, as well as the arrow from survey impression to decide-to-respond, are dropped. This results in deconfounding satisfaction estimation from the confounder and the response rate bias.

4 INVERSE PROPENSITY WEIGHTING AS A DEBIASING MECHANISM FROM RESPONSE RATE BIAS

Without such an intervention mechanism, we propose to use *inverse propensity weighting* to de-bias user satisfaction estimation from response bias. Let us denote with $P(r(u, i, c) = 1|\mathbf{x}_u, \mathbf{x}_i, \mathbf{x}_c)$ the propensity score of user u deciding to respond to a survey on item i at context c. The propensity scores are typically learned in econometrics using an independent logistic regression model [11], and then are utilized as inverse weights in the training of the main

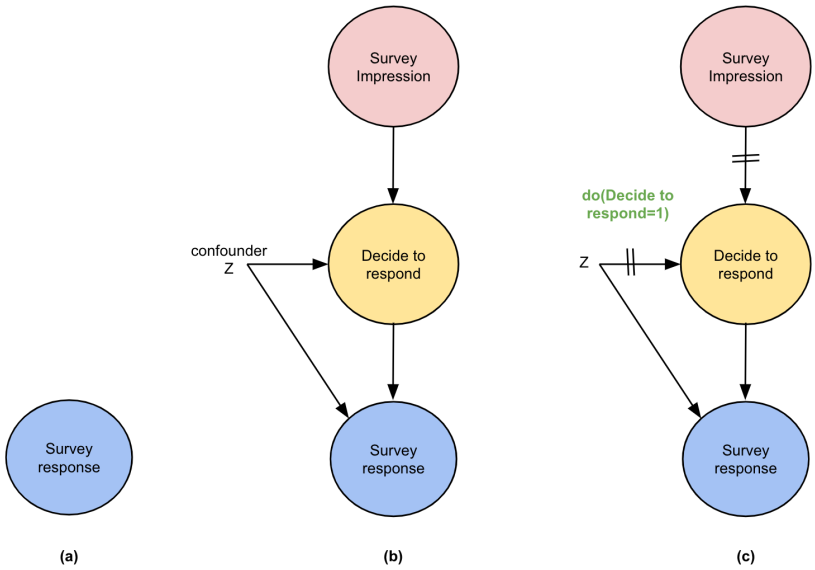

Figure 2: (a) User satisfaction models are trained only on survey responses, ignoring the response rate bias. (b) Causal model demonstrating that deciding to respond is a cause for survey response, and both 'decide to respond' and 'survey response' are coupled by the confounder variable Z. (c) Ideally, we would like to have access to a do-dataset, where we set do(Decide to respond=1), as in this case we have removed the confounder variable and the deciding to respond (response rate) bias.

estimation task,

$$\mathcal{L}_r(\theta) = \sum_{(u,i,c)\in\mathcal{D}_r} \frac{1}{P(r(u,i,c)=1|\mathbf{x}_u,\mathbf{x}_i,\mathbf{x}_c)} \ell_r(s(u,i,c),\hat{s}(u,i,c;\theta). \tag{2}$$

The estimation of the survey response becomes unbiased when the propensity scores are accurate.

In Figure 3(b) we illustrate our proposed solution. Instead of building an independent logistic propensity model, we extend the standalone satisfaction model described in Section 2.3 to a multi-task learning framework. Our new model has a shared-bottom two-headed architecture [2] — the two heads share input embeddings, but also have task-specific parameters. The left tower in Figure 3(b) represents the user satisfaction task trained on survey responses $\mathcal{D}_r$— ignoring the right tower, this architecture is exactly the same as Figure 3(a). The right tower represents the response rate task trained on survey impressions $\mathcal{D}_i$, predicting the contextual propensity scores in $[0, 1]$ of deciding to respond. The response rate tower takes as input the *same* embeddings of user, item and context features $\mathbf{x}_u, \mathbf{x}_i, \mathbf{x}_c$ as those input to the satisfaction task, and maps them through its own task-specific ReLU layers and a final logistic layer to predict the probability of deciding-to-respond $r(u, i, c)$. That is,

$$\mathcal{L}_i(\theta) = \sum_{(u,i,c)\in\mathcal{D}_i} r(u,i,c)\log P(r(u,i,c)=1|\mathbf{x}_u,\mathbf{x}_i,\mathbf{x}_c;\theta) + (1-r(u,i,c))\log\left(1-P(r(u,i,c)=1|\mathbf{x}_u,\mathbf{x}_i,\mathbf{x}_c;\theta)\right). \tag{3}$$

The loss of the satisfaction task is changed from the one in Equation (1), to a weighted loss, utilizing as weights the inverse of the response rate head predictions, as specified in Equation (2). The parameters of both tasks, as well as the common embeddings, are learned concurrently, with the total loss being the sum of the two tasks' losses, i.e., $\mathcal{L}_r(\theta) + \mathcal{L}_i(\theta)$.

Given the importance of having accurate propensity scores so to de-bias user satisfaction from response rate, we needed to combat training challenges associated with the extreme sparsity of survey responses out of survey impressions. In the large-scale data we used, users responded to roughly 2% of surveys, making this a highly imbalanced dataset. We experimented with two training schemes for the response rate head, and show their results in Figure 4: training on (a) the actual imbalanced $\mathcal{D}_i$ data, and (b) balanced data by sub-sampling the non-responded surveys to have 1:1 ratios of responses to non-responses, but calibrating the propensity predictions to match the ground truth ones in the original $\mathcal{D}_i$ dataset [3]. We observed a clear win of strategy (b) for learning an accurate propensity model, converging much faster and reaching a higher AUC ROC.

To address the known issue of increased variance the inverse propensity weighting technique introduces, we clip the inverse weights so they are always within a certain range, at the cost of not reaching a fully unbiased estimate [12]. In practice, we found using as maximum and minimum values the inverse of the 25-percentile and 75-percentile of the response rate prediction gave good empirical results for our experiments.

In Figure 5, we showcase the promise of our approach in offline experiments, comparing it with the standalone model ignoring the response rate bias as shown in Figure 3(a). We slice the satisfaction model's estimation error across the predicted propensities of deciding-to-respond, where we define here as error the Mean

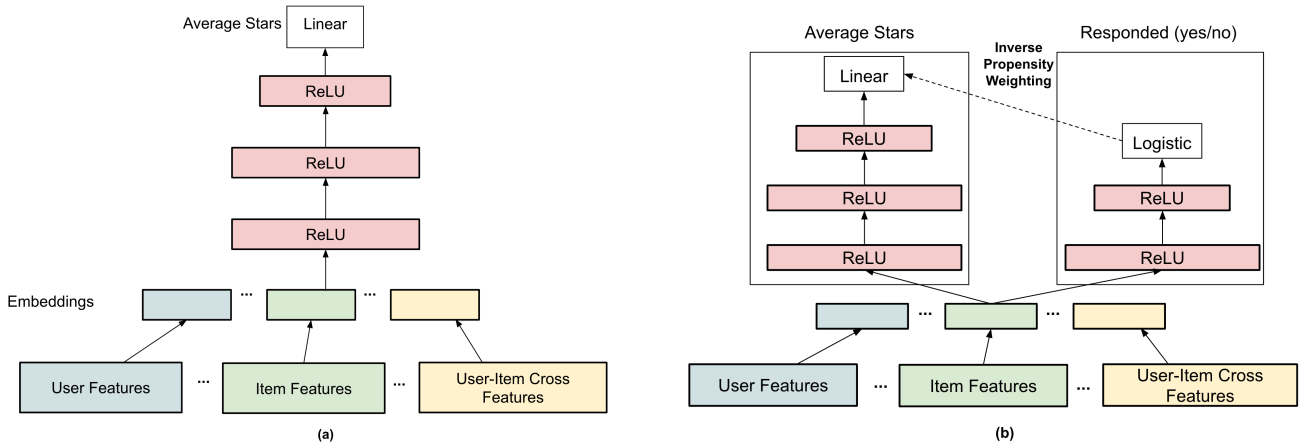

Figure 3: (a) Feed-forward User Satisfaction Imputation Model trained on survey responses. (b) Proposed Shared-bottom User Satisfaction model, trained with two-tasks: user satisfaction and response rate, using inverse propensity weighting.

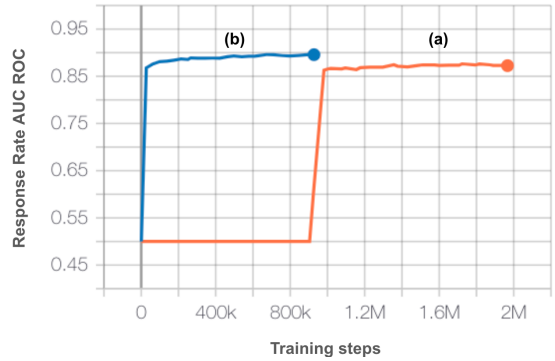

Figure 4: AUC-ROC of Response Rate Head in test set (y-axis) as training progresses (x-axis) for the two schemes.

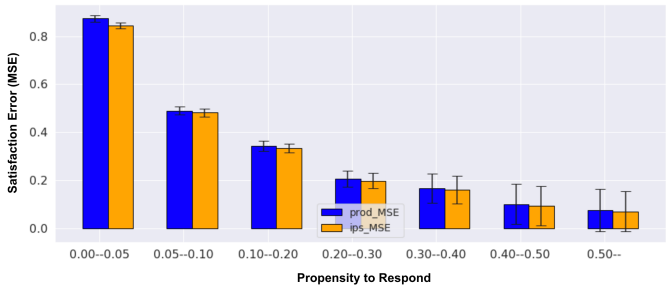

Figure 5: Offline comparison in terms of Mean Squared Error (MSE) of standalone satisfaction model with the proposed inverse propensity weighting (ips) approach across predicted propensities to respond ranging from 0 to 1. Lower MSE values are better.

Squared Error (MSE) of the predicted survey responses and the ground truth. We find that not only we improve the data slices with lower propensities to respond, but we also do not hurt the slices with higher response propensity.

We also tested in live traffic our approach, compared with the standalone satisfaction model. If we hash the user cookies to random buckets and consider their corresponding response rates, i.e., number of times users responded to surveys out of number of times surveys were shown to them, we can see how our approach affects live satisfaction metrics across response rate buckets. Particularly, we computed the quantiles of these response rates, and for each of them calculated the percentage difference in live metrics indicating dissatisfying experience.

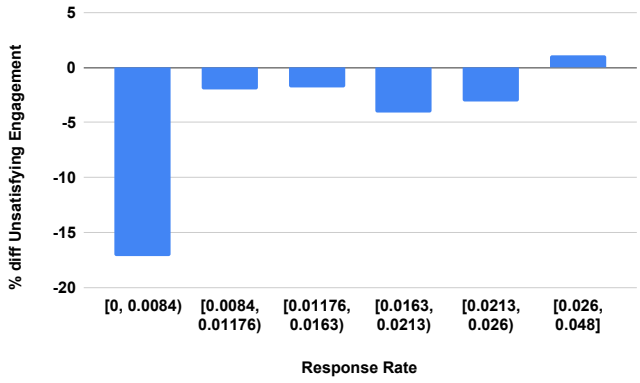

Figure 6: A/B experiment results, comparing standalone satisfaction model with our proposed approach.

We can see in Figure 6 that the decrease of non-satisfying engagement is much more pronounced in the lower quantile of response rates, but is observed nonetheless in almost all response rate buckets. In the highest response rate quantile, we do indeed notice a slight increase of non-satisfying experience, but this is a much lower loss compared to the intended gain of decreasing non-satisfying experiences across all other response rate groups. Finally, although not

shown here, the *overall* live metrics of non-satisfying experience decrease significantly.

5 CONCLUSION

In this paper, we considered the problem of estimating user satisfaction based on survey data, which is at the forefront of large-scale industrial recommender systems. We emphasized that the data used to train such user satisfaction models are inherently biased, as users are more likely to respond to a survey when they will respond that they are more satisfied. We provide empirical evidence of the bias in a real-world large scale dataset. We also posed the existence of this response rate bias in a causal framework. At the absence of intervention data to deconfound, we instead proposed to utilize the inverse propensity weighting technique to reweigh survey response examples by the inverse of their corresponding propensities to respond. To do this, we introduced a shared bottom two-headed architecture, where both satisfaction and response rate tasks are learned concurrently, and the inverse predictions of the latter are used as weights for the loss of the former. We showed via both offline and live A/B experiments the merit of our approach. The model estimation error is decreased for lower propensity-to-respond slices. Live metrics indicate that non-satisfying experiences are decreased overall, and the decrease is much more pronounced in slices with lower propensities to respond, as expected.

ACKNOWLEDGMENTS

The authors would like to thank Chris Berg, Eric Bencomo Dixon for enabling the work, and providing valuable support.

REFERENCES

[1] John Bound, Charles Brown, and Nancy Mathiowetz. 2001. Measurement error in survey data. In *Handbook of econometrics*. Vol. 5. Elsevier, 3705–3843.
[2] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.
[3] Olivier Chapelle, Eren Manavoglu, and Romer Rosales. 2014. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 4 (2014), 1–34.
[4] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 456–464.
[5] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on Recommender Systems*. 191–198.
[6] Paolo Dragone, Rishabh Mehrotra, and Mounia Lalmas. 2019. Deriving User-and Content-specific Rewards for Contextual Bandits. In *Proceedings of The Web Conference 2019*. 2680–2686.
[7] Jean Garcia-Gathright, Christine Hosey, Brian St Thomas, Ben Carterette, and Fernando Diaz. 2018. Mixed methods for evaluating user satisfaction. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 541–542.
[8] Jean Garcia-Gathright, Brian St. Thomas, Christine Hosey, Zahra Nazari, and Fernando Diaz. 2018. Understanding and evaluating user satisfaction with music discovery. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 55–64.
[9] Qi Guo and Eugene Agichtein. 2012. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of the 21st International conference on World Wide Web*. 569–578.
[10] Keisuke Hirano, Guido W Imbens, and Geert Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71, 4 (2003), 1161–1189.
[11] Guido W Imbens and Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.
[12] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*. 781–789.
[13] Mounia Lalmas. 2019. Metrics, Engagement & Personalization. In *REVEAL workshop, The ACM Conference Series on Recommender Systems*. https://www.slideshare.net/mounialalmas/engagement-metrics-and-recommenders
[14] Dawen Liang, Laurent Charlin, and David M Blei. 2016. Causal inference for recommendation. In *Causation: Foundation to Application, Workshop at UAI*. AUAI.
[15] Roderick JA Little and Donald B Rubin. 2019. *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons.
[16] Rishabh Mehrotra, Mounia Lalmas, Doug Kenney, Thomas Lim-Meng, and Golli Hashemian. 2019. Jointly leveraging intent and interaction signals to predict user satisfaction with slate recommendations. In *Proceedings of The Web Conference 2019*. 1256–1267.
[17] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2243–2251.
[18] Delroy L Paulhus. 1991. Measurement and control of response bias. (1991).
[19] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. *arXiv preprint arXiv:1602.05352* (2016).
[20] Matthias Schonlau, Arthur Van Soest, Arie Kapteyn, and Mick Couper. 2009. Selection bias in web surveys and the use of propensity scores. *Sociological Methods & Research* 37, 3 (2009), 291–318.
[21] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 713–722.
[22] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 115–124.
[23] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 610–618.
[24] Hongyi Wen, Longqi Yang, and Deborah Estrin. 2019. Leveraging post-click feedback for content recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 278–286.
[25] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. 2014. Beyond clicks: dwell time for personalization. In *Proceedings of the 8th ACM Conference on Recommender systems*. 113–120.
[26] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-scale Causal Approaches to Debiasing Post-click Conversion Rate Estimation with Multi-task Learning. In *Proceedings of The Web Conference 2020*. 2775–2781.
[27] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 43–51.