Influence Function for Unbiased Recommendation

Jiangxing Yu*†
futuresoul00@outlook.com
The University of Melbourne

Hong Zhu*
zhuhong8@huawei.com
Huawei Noah's ark lab

Chih-Yao Chang*
alex.chang.prsn@gmail.com
Huawei Noah's ark lab

Xinhua Feng
fengxinhua1@huawei.com
Huawei Noah's ark lab

Bowen Yuan
f03944049@csie.ntu.edu.tw
National Taiwan University

Xiuqiang He
hexiuqiang1@huawei.com
Huawei Noah's ark lab

Zhenhua Dong‡
dongzhenhua@huawei.com
Huawei Noah's ark lab

ABSTRACT

Recommender system is one of the most successful machine learning technologies for commerce. However, it can reinforce the closed feedback loop problem, where the recommender system generates items to users, then the further recommendation model is trained with the data that users' feedback to the items. Such self-reinforcing pattern can cause data bias problems. There are several debiasing methods, inverse-propensity-scoring (IPS) is a practical one for industry product. Since it is relatively easy to reweight training samples, and ameliorate the distribution shift problem. However, because of deterministic policy problem and confoundings in real-world data, it is hard to predict propensity score accurately. Inspired by the sample reweight work for robust deep learning, we propose a novel influence function based method for recommendation modeling, and analyze how the influence function corrects the bias. In the experiments, our proposed method achieves better performance against the state-of-the-art approaches.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

recommender system; Influence Function; counterfactual learning

ACM Reference Format:

Jiangxing Yu, Hong Zhu, Chih-Yao Chang, Xinhua Feng, Bowen Yuan, Xiuqiang He, and Zhenhua Dong. 2020. Influence Function for Unbiased Recommendation. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401321

*All the authors contributed equally to this research.
†Work done at Huawei Noah's Ark Lab as an intern.
‡Corresponding author.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401321

1 INTRODUCTION

Recommender systems usually need to train models for predicting users' preferences to items via large amounts of interaction log data. Then, the trained models will try to fit the training distribution of historical interaction data, but there exists distribution shift between training distribution $P(x, y)$ and target unbiased distribution $Q(x, y)$, where x and y are one sample's feature vector and label, since recommender systems are always subject to variety of biases, such as selection bias [12, 15, 17], position bias [4], gender bias [14], popularity bias [1]. The distribution shift makes it difficult to generate good online prediction performance.

How to get an unbiased recommendation with biased training data? This is a basic and long-standing research question, which has been studied in many research areas, such as *covariate shift* [8], *missing not at random* (MNAR) [12, 15], *domain adaptation* [3], *counterfactual learning* [2, 5, 17]. Among the various methods, IPS method, error-imputation-based (EIB) method, and doubly robust method which can be regarded as the combination of IPS and EIB, are the recent mainly methods to address the bias problem in recommendation [15, 17]. In this paper, our work mainly focuses on the IPS family of methods, because IPS is relatively easy to be deployed in industry, and it is foreseeable that our proposed method could improve the performance of doubly robust method, since IPS is a part of doubly robust method.

The core principle of IPS is to estimate the ratio $P(x, y)/Q(x, y)$, which is called propensity score, and use the propensity score to reweight the training samples such that the mean of the training distribution after reweighted is equal to target unbiased distribution. Although intuitively, this method has two main problems to accurately learn the propensity score in recommendation scenario:

(1) From the definition of propensity score, we can see that there is a prerequisite assumption that the logging policy should be stochastic, namely every individual item has a nonzero chance of being assigned to any user [5]. However, the real-world recommendation ranking is decided by the expected benefit, then a user can only browse the top-K items with high predicted Click-Through-Rate (CTR) or effective-Cost-Per-Mile (eCPM). This is a deterministic logging policy which will reduce the accuracy of IPS [10, 17].

(2) There always exits unobserved confoundings in real-world data. In fact, not only one kind of bias exists in recommendation systems, such as the aforementioned four kinds of bias could happen together, furthermore, gender bias and popularity bias could affect selection bias, selection bias could further affect position bias.

In light of the above considerations, we try to reweight training sample directly, rather than to model the bias propensity. Moreover, IPS can be regarded as a specific case of reweight learning. On the other hand, Influence Function (IF) measures the effect on the estimator when adding an small perturbation on one training sample, this could reveal the importance of one training sample. Hence, we propose a novel method based on IF for unbiased recommendation (IF4URec), which utilizes IF to reweight training loss of each sample to get less loss in an unbiased validation for debiasing. The most relevant prior work is [11], which follows a meta-learning paradigm to learn the sample weights for robust deep learning via IF. Compared to [11], our work extends deep learning to other recommendation methods, and proposes a novel reweighted method, which has better performance validated in our experiments.

To summarize, our contributions are the following:

- We directly learn to reweight training samples with a novel method, which could break through the bottleneck of IPS.
- The proposed reweighting method is applicable for mainly recommendation models, and we analyze the inner mechanism of influence function for correcting bias.
- We derive influence function of general low rank model, and propose the details of implementation.
- Completed experiments are conducted to demonstrate the superiority of our method.

2 BACKGROUND

In this section, we will briefly introduce IPS and influence function.

2.1 IPS

In general, a learning model minimizes the regularized variable of empirical risk, $L(z,\theta) = \frac{1}{n}\sum_{i=1}^{n} l(z_i,\theta) + \lambda R(\theta)$, where $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ is an observed sample drawn from training distribution $P(x,y)$, $\theta \in \Theta$ is a model parameter, $l(z_i,\theta)$ is a loss function, $R(\theta)$ is a regularizer to alleviate overfitting. Then the parameters of optimal function is $\hat{\theta} \triangleq \arg\min_{\theta\in\Theta} L(z,\theta)$.

If $P(x,y)$ is different from target distribution $Q(x,y)$, IPS method reweights loss function with the inverse of propensity score:

$$\hat{\theta}_{IPS} \triangleq \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n}\frac{l(z_i,\theta)}{p_i} + \lambda R(\theta), \tag{1}$$

where $p_i = P(x_i,y_i)/Q(x_i,y_i)$ is propensity score.

It has been proved in past works [8, 12, 15, 17] that IPS can get an unbiased estimator. In theory, IPS method can correct any kinds of biases in recommender system. Especially, if assume $P(y|x) = Q(y|x)$, then $p_i = P(x_i,y_i)/Q(x_i,y_i) = P(y_i|x_i)P(x_i)/[Q(y_i|x_i)Q(x_i)] = P(x_i)/Q(x_i)$, this case is called selection bias [8].

An accurate IPS estimator should satisfy two assumptions, *overlap* and *unconfoundeness* [5], which have been discussed in the introduction section.

2.2 Influence Function

Influence Function is an important concept within the scope of Robust Statistics, and it is firstly used to measure example-wise influence on validation loss in [9]. The influence of a training sample z_i over a validation sample z_j is defined as follow:

$$\phi(z_i,z_j) \triangleq \frac{dl(z_j,\hat{\theta}_\epsilon)}{d\epsilon}|_{\epsilon=0} = -\nabla_\theta l(z_j,\hat{\theta})^\top H_{\hat{\theta}}^{-1}\nabla_\theta l(z_i,\hat{\theta}), \tag{2}$$

where $\hat{\theta}_\epsilon$ is the optimal parameter after a specific training sample z_s is upweighted by a small perturbation ϵ, namely $\hat{\theta}_\epsilon = \arg\min_{\theta\in\Theta} L(z,\theta) + \epsilon l(z_s,\theta)$; $H_{\hat{\theta}} \triangleq \nabla_\theta^2 L(z,\hat{\theta})$ is the Hessian matrix and assumed to be positive definite. In this paper, *the validation set is unbiased.*

3 PROPOSED METHOD

In this section, we propose a novel method named IF4URec to transfer the IF of a training sample to its corresponding weight, and analyze the inner mechanism of IF for correcting bias, then describe how to calculate IF efficiently.

3.1 IF4URec

According to the definition of IF in Eq. (2), the influence of a training sample z_i over the whole validation set is:

$$\phi(z_i) = \sum_j \phi(z_i,z_j) = -[\sum_j \nabla_\theta l(z_j,\hat{\theta})]^\top H_{\hat{\theta}}^{-1}\nabla_\theta l(z_i,\hat{\theta}) \tag{3}$$

Let ϵ_i be a considered small perturbation on z_i, $\hat{\theta}_{\epsilon_i}$ the optimal parameter under the perturbation, we can approximate the validation loss change through $\sum_j l(z_j,\hat{\theta}_{\epsilon_i}) - \sum_j l(z_j,\hat{\theta}) \approx \epsilon_i\phi(z_i)$. Further, if there are n perturbations $\vec{\epsilon} = (\epsilon_1,\epsilon_2,...,\epsilon_n)^\top$ for each training sample, with the optimal parameter $\hat{\theta}_\epsilon$, the validation loss change can be approximated as:

$$\sum_j l(z_j,\hat{\theta}_\epsilon) - \sum_j l(z_j,\hat{\theta}) \approx \sum_{i=1}^{n}\epsilon_i\phi(z_i) \tag{4}$$

Meanwhile, the training objective function under perturbation $\vec{\epsilon}$ is $L(z,\theta,\vec{\epsilon}) = \frac{1}{n}\sum_{i=1}^{n}(1+n\epsilon_i)l(z_i,\theta) + \lambda R(\theta)$, which indicates $L(z,\theta,\vec{\epsilon})$ has the same structure with IPS, and the loss of each training sample is weighted by $\pi_i = 1 + n\epsilon_i \in [0,B]$, where B is the upper bound of weights to reduce variance of estimator. For the sake of debiasing, a good sample weighting $\vec{\pi} = (\pi_1,\pi_2,...,\pi_n)^\top$ should minimize the loss of the unbiased validation:

$$\begin{aligned}\vec{\pi}^\star &= \arg\min_{\vec{\pi}}\sum_j l(z_j,\hat{\theta}_\epsilon) = \arg\min_{\vec{\pi}}\sum_j[l(z_j,\hat{\theta}_\epsilon) - l(z_j,\hat{\theta})] \\ &= \arg\min_{\vec{\pi}}\sum_{i=1}^{n}\frac{\pi_i - 1}{n}\phi(z_i)\end{aligned} \tag{5}$$

From Eq. (5), sample z_i with a less $\phi(z_i)$ should have a greater π_i. Let's go back the Eq. (3) to see how a sample could have a less influence. According to Eq. (3), $\phi(z_i)$ mainly consists of two parts, $\sum_j \nabla_\theta l(z_j,\hat{\theta})$ and $H_{\hat{\theta}}^{-1}\nabla_\theta l(z_i,\hat{\theta})$. In fact, the first part is the gradient direction of the estimator trained by gradient descent optimization algorithm on the unbiased validation set at parameter $\hat{\theta}$; the second part is the contribution of sample z_i to the iterative update direction of the estimator trained on the biased training set by Newton's method at $\hat{\theta}$. Therefore, the directions of the above two parts are more closed, the influence $\phi(z_i)$ is less, and increasing weight π_i

can make the estimator trained on the biased training set updated towards the distribution of unbiased validation set. Above analysis is the inner mechanism of utilizing IF for debiasing, and note that there is no need to estimate the propensity score $P(x_i, y_i)/Q(x_i, y_i)$. In other words, the proposed method does not need to consider overlap and unconfoundeness, which are necessary to IPS methods.

In fact, the Eq. (5) has mathematically optimal solution $\pi_i = \{if\ \phi(z_i) < 0,\ B;\ otherwise,\ 0\}$. And in previous work [11], $\pi_i = max(-\phi(z_i), 0)$. However, these two weighting methods will introduce overfitting over validation set, because the transfer function from IF to sample weight is no longer Lipschitz continuous [16]. Therefore, we propose a novel weighting method as follow:

$$\pi_i = \frac{1}{1 + e^{\frac{\alpha\phi(z_i)}{\max(\{\phi(z_i)\}) - \min(\{\phi(z_i)\})}}}, \tag{6}$$

where $\alpha \in \mathbb{R}^+$. This sigmoid style function can be treated as the smooth approximation of the optimal solution of Eq. (5) which is a step function. In addition, the empirical estimator of the expectation of $\vec{\pi}$ is normal distribution, so we follow the setting in [8] to enforce the constraint $\frac{1}{n}\sum_{i=1}^{n} \pi_i = 1$.

Algorithm 1: IF4URec

Input: training set $\{z_i\}$, unbiased validation set $\{z_j\}$

1 Train the initial estimator $\hat{\theta}$;
2 Calculate the IF $\phi(z_i)$ for every training sample z_i ;
3 Get the weight π_i for every training sample z_i via Eq. (6), then $\pi_i := \pi_i / \sum_{i=1}^{n} \pi_i$, and enfore $\pi_i = B$ if $\pi_i > B$;
4 Train unbiased estimator $\hat{\theta}_{IF} \triangleq \arg\min_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^{n} \pi_i l(z_i, \theta) + \lambda R(\theta)$;

Output: $\hat{\theta}_{IF}$

Our proposed method is summarized in Algorithm 1. It's worth noting that the framework of IF4URec is applicable for common recommendation models, since we could calculate the IF for almost all the models, including convex and non-convex ones [9].

3.2 Computing Influence Function

Efficient computation for IF is very crucial in our method. Here, without losing generality, we study the IF calculation for Logistic Regression (LR), a classic convex problem, and Factorization Machine (FM), a classic non-convex problem. We compute

$$h = H_{\hat{\theta}}^{-1} \sum_j \nabla_\theta l(z_j, \hat{\theta}) \tag{7}$$

via Conjugate Gradient (CG) [6] firstly. Then, we compute the IF of each training instance z_i via $-\nabla_\theta l(z_i, \hat{\theta})^\top h$. For LR, we apply Mixed Preconditioned CG [7] to reduce CG steps. For FM, note that the Hessian Matrix of a non-convex problem is not positive definite. So, we apply Gauss-Newton method [13] to compute (7) for FM.

Due to the limitation of length, we leave details and derivation in Supplementary [1].

[1] https://bit.ly/2SVCO7A

4 EXPERIMENTS

In this section, we firstly introduce the data sets and considered baselines for comparison. Then, we conduct experiments to demonstrate and analyze the effectiveness of our proposed method.

4.1 Data sets

We conduct experiments on two real-world data sets, Yahoo!R3 and Coat, which are commonly applied in related work [12, 15].

Yahoo!R3[2] consists of user-song ratings in which the 311,704 MNAR ratings were generated from 15400 users' normal interactions with Yahoo! Music services to 1000 songs, while the 54,000 missing at random (MAR) ratings from users' ratings of randomly selected songs conducted by Yahoo! Research.

Coat[3] contains 6960 MNAR ratings and 4640 MAR ratings. It was collected through a simple online web-shop interface, where 290 users were asked to rate 24 coats from 300 selected by themselves and then rate extra 16 ones randomly selected.

The two data sets are both on a one-five scale. We treat ratings of one and two as the negative while four and five as the positive, and discard samples rated three. Each set is split into 4 parts: non-uniform training, uniform training, validation and test, the later three sets are obtained via equally splitting MAR data. Firstly we do parameter selection via validation data, then compute sample weights of non-uniform data via uniform training data, and train on the weighted non-uniform training data and uniform training data together to do performance test over test data at last.

4.2 Considered baselines

4.2.1 *conventional CTR prediction methods.*

We use LR and FM as conventional CTR methods. Three kinds of data set below are applied to train conventional models.

- S_t: A large non-uniform training data set.
- S_c: A small uniform training data set.
- $S_t \cup S_c$: This data set contains both S_t and S_c.

4.2.2 *IPS.*

We use two methods to estimate propensity scores.

- IPS via Naive Bias (IPS-NB) [12]: it assumes that rating is independent of features but only depend on rating.

$$p_i \approx P(O = 1 | y_i = y) \approx P(y, O = 1)/P(y), \tag{8}$$

  where $y = \{1, 0\}$ is the rating, $P(y, O = 1)$ is the probability that events labeled y are shown, and $P(y)$ is the marginal probability of events labeled y in unbiased data set S_c.
- IPS via Naive Bias for item-wise (IPS-NB-IW): we also expand the approach in [12]. It assume the dependencies between rating y and item A are not negligible.

$$p_i \approx P(O = 1 | y_i = y, A_i = a) \approx P(y, a, O = 1)/P(y, a). \tag{9}$$

4.2.3 *Learning to reweight approach.*

- Influence function based learning to reweight (IF-L2R): the most relevant prior work proposed by [11], which employs IF to calculate weight of ith sample as following,

$$w_i = max(-\phi_i, 0) / \sum_j max(-\phi_j, 0). \tag{10}$$

[2] https://webscope.sandbox.yahoo.com
[3] https://www.cs.cornell.edu/ schnabts/mnar

Table 1: Performances of the methods in two data sets.

Metric	NLL	AUC	NLL	AUC
Approach \ Data set	**Yahoo! R3**		**Coat**	
Logistics Regression				
LR (S_t)	-0.6053	0.7298	-0.5627	0.6279
LR (S_c)	-0.3271	0.7370	-0.5501	0.6170
LR ($S_t \cup S_c$)	-0.5123	0.7754	-0.5547	0.6316
IPS-NB	-0.3029	0.7656	**-0.5457**	0.6318
IPS-NB-IW	-0.3076	0.7543	-0.5490	0.6172
IF-L2R	-0.3778	0.6552	-0.8030	0.5780
IF4URec	**-0.2980**	**0.7969**	-0.5464	**0.6328**
Factorization Machine				
FM (S_t)	-0.6405	0.7499	-0.5658	0.6224
FM (S_c)	-0.3078	0.7493	-0.5561	0.6041
FM ($S_t \cup S_c$)	-0.4775	0.7817	-0.5513	0.6245
IPS-NB	**-0.3009**	0.7806	-0.5457	0.6254
IPS-NB-IW	-0.3051	0.7776	-0.5484	0.6112
IF-L2R	-0.3429	0.6609	-0.5564	0.6220
IF4URec	-0.3672	**0.7855**	**-0.5452**	**0.6276**

4.3 Results

We apply two commonly used metrics in recommendation, the Negative Log Loss (NLL) and the Area Under the ROC Curve (AUC), their computational formulas could be found in [17]. By comparing data in Table1 and Table2, we conclude the following observation.

In terms of conventional methods, the methods based on S_c have better NLL scores than those based on S_t and $S_t \cup S_c$. Because S_c has lower positive sample ratios which could lead to higher NLL scores, and the distribution of S_c is consistent with that of the test set. However, the AUC of $LR(S_c)$ and $FM(S_c)$ are not good, because the data volume of S_c is much smaller than S_t. Another interesting observation is that FM is not always better than LR despite careful hyperparameter tuning, especially in Coat data set.

IPS methods have good NLL scores, even IPS-NB has the best NLL scores in Coat for LR and Yahoo!R3 for FM. The reason is that it adjusts the positive ratio to make it approach to the uniform dataset (S_c). However, its AUC is not competitive, even lower than conventional approaches, because it greatly reduces the information of positive samples and over-amplifies the information of negative samples, then the model is not fully learned.

IF-L2R does not perform well, because it only considers the data with negative influence function, which causes over-fitting and information loss. For example, about 56.66% samples get positive IF values in Yahoo! R3 for LR, so IF-L2R will ignore more than half information in training data. This results in worse AUC (0.6552). However, by means of calculating influence function in each epoch to update sample weight, [11] shows good results in deep learning, but this will introduce expensively time consuming cost.

For IF4URec, it adjusts the positive ratios closer to S_c, but it is still much more balanced than the uniform dataset. NLL scores are susceptible to data imbalance, the lower positive ratio leads to better NLL more easily. Therefore, NLL scores of IF4URec are not the best all the time, such as lower than IPS methods in Coat for LR and Yahoo! R3 for FM, but still always better than S_t and $S_t \cup S_c$. On the other hand, it performs the best in term of AUC, which is not sensitive to data imbalance in all sets for both LR and FM. To sum up, IF4URec demonstrates the best comprehensive performance.

Table 2: Positive ratios of the methods in CPC data set.

DataSet	S_t	S_c	IPS	IF4URec-LR	IF4URec-FM
Yahoo! R3	47.64%	10.05%	10.05%	33.98%	47.48%
Coat	36.30%	22.71%	22.71%	32.57%	31.75%

5 CONCLUSIONS

In this work, we analyze two main challenges of conventional IPS methods in recommender system. (1) The deterministic policy reduces the accuracy of IPS methods; (2) IPS methods can not debias when there exists unobserved confounding in data. To overcome the difficulties, we propose a widely applicable method, which directly learns to reweight training samples with influence function, and we design some efficient algorithms for the computation of influence function. Experiments show that our proposed method outperforms both of conventional IPS methods and reweighting models.

REFERENCES

[1] Abdollahpouri, H., Burke, R., and Mobasher, B. Managing popularity bias in recommender systems with personalized re-ranking. *arXiv preprint arXiv:1901.07555* (2019).
[2] Bonner, S., and Vasile, F. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), ACM, pp. 104–112.
[3] Chen, Y.-J., Chu, H.-C., Chen, Y.-M., and Chao, C.-Y. Adapting domain ontology for personalized knowledge search and recommendation. *Information & Management 50*, 6 (2013), 285–303.
[4] Guo, H., Yu, J., Liu, Q., Tang, R., and Zhang, Y. Pal: A position-bias aware learning framework for ctr prediction in live recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems* (New York, NY, USA, 2019), RecSys '19, ACM, pp. 452–456.
[5] Hassanpour, N., and Greiner, R. Counterfactual regression with importance sampling weights. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (2019), IJCAI'19, AAAI Press, pp. 5880–5887.
[6] Hestenes, M. R., and Stiefel, E. Methods of conjugate gradients for solving linear systems. *JRNBS 49* (1952), 409–436.
[7] Hsia, C.-Y., Chiang, W.-L., and Lin, C.-J. Preconditioned conjugate gradient methods in truncated newton frameworks for large-scale linear classification. In *Asian Conference on Machine Learning* (2018), pp. 312–326.
[8] Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. J. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems* (2007), pp. 601–608.
[9] Koh, P. W., and Liang, P. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), JMLR. org, pp. 1885–1894.
[10] Lawrence, C., Sokolov, A., and Riezler, S. Counterfactual learning from bandit feedback under deterministic logging: A case study in statistical machine translation. *arXiv preprint arXiv:1707.09118* (2017).
[11] Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050* (2018).
[12] Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., and Joachims, T. Recommendations as treatments: Debiasing learning and evaluation. *arXiv preprint arXiv:1602.05352* (2016).
[13] Schraudolph, N. N. Fast curvature matrix-vecctor products for second-order gradient descent. *Neural Computation 14* (2002), 1723–1738.
[14] Timmermans, A. C., de Boer, H., Amsing, H., and van der Werf, M. Track recommendation bias: Gender, migration background and ses bias over a 20-year period in the dutch context. *British Educational Research Journal 44*, 5 (2018), 847–874.
[15] Wang, X., Zhang, R., Sun, Y., and Qi, J. Doubly robust joint learning for recommendation on data missing not at random. In *International Conference on Machine Learning* (2019), pp. 6638–6647.
[16] Wang, Z., Zhu, H., Dong, Z., He, X., and Huang, S.-L. Less is better: Unweighted data subsampling via influence function. *arXiv preprint arXiv:1912.01321* (2019).
[17] Yuan, B., Hsia, J.-Y., Yang, M.-Y., Zhu, H., Chang, C.-Y., Dong, Z., and Lin, C.-J. Improving ad click prediction by considering non-displayed events. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019), ACM, pp. 329–338.