Cascade Model-based Propensity Estimation for Counterfactual Learning to Rank

Ali Vardasbi
University of Amsterdam
Amsterdam, The Netherlands
a.vardasbi@uva.nl

Maarten de Rijke
University of Amsterdam & Ahold Delhaize
Amsterdam, The Netherlands
m.derijke@uva.nl

Ilya Markov
University of Amsterdam
Amsterdam, The Netherlands
i.markov@uva.nl

ABSTRACT

Unbiased counterfactual learning to rank (CLTR) requires click propensities to compensate for the difference between user clicks and true relevance of search results via inverse propensity scoring (IPS). Current propensity estimation methods assume that user click behavior follows the position-based click model (PBM) and estimate click propensities based on this assumption. However, in reality, user clicks often follow the cascade model (CM), where users scan search results from top to bottom and where each next click depends on the previous one. In this cascade scenario, PBM-based estimates of propensities are not accurate, which, in turn, hurts CLTR performance. In this paper, we propose a propensity estimation method for the cascade scenario, called cascade model-based inverse propensity scoring (CM-IPS). We show that CM-IPS keeps CLTR performance close to the full-information performance in case the user clicks follow the CM, while PBM-based CLTR has a significant gap towards the full-information. The opposite is true if the user clicks follow PBM instead of the CM. Finally, we suggest a way to select between CM- and PBM-based propensity estimation methods based on historical user clicks.

ACM Reference Format:
Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2020. Cascade Model-based Propensity Estimation for Counterfactual Learning to Rank. In *Proceedings of the 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/10.1145/3397271.3401299

1 INTRODUCTION

Traditional learning to rank (LTR) and online LTR require explicit relevance labels and intervention through search engine results, respectively [13, 14]. In contrast, counterfactual learning to rank (CLTR) only requires historical click logs for learning. Obtaining and using historical click logs incurs no extra cost and does not impose any risk of reduced user satisfaction. More importantly, such benefits come without any significant reduction in LTR performance [2, 12, 16, 17]. However, user clicks are known to suffer from different types of bias, such as position bias, selection bias, trust bias, etc. [7]. Due to these types of bias, each result on a search engine result page (SERP) has a different *propensity* of being clicked. Since CLTR learns from user clicks, it should take those propensities into account. To make CLTR unbiased, the inverse propensity scoring (IPS) method has been introduced in [12, 16].

In IPS-based CLTR, click models are used to estimate propensities [12]. Even though the theoretical IPS method does not rely on any specific click model [12], current IPS-based CLTR experiments rely on the position-based click model (PBM) [2, 12, 16, 17]. In PBM, the probability of examining a result depends on the result's rank only and not on any other context, such as clicks on other items.

Although PBM is a well-performing click model [7], it does not always approximate user clicks well [9]. Importantly, PBM fails to represent the cascade user click behavior, where a user scans a SERP from top to bottom and where each next click depends on the previous click [8] – such behavior is often observed in practice [7, 9]. In this case, PBM-IPS estimators are not accurate and CLTR performance drops considerably. Look, for example, at Figure 1. There, the PBM-IPS CLTR is trained over two different simulated click logs: one following PBM and the other following dependent click model (DCM) [11] one of the popular cascade-based models. When trained on the same number of clicks and the same queries, PBM-IPS performs significantly better on the PBM than DCM simulated clicks. The "Full Info" legend in this plot shows the performance of LTR trained on real relevance tags instead of simulated clicks. PBM-IPS performs close to the full-info only when trained on PBM simulated clicks.

In this paper, we first experimentally validate this observation for different parameter settings. We apply PBM-IPS unbiased CLTR on various sets of simulated clicks and show that PBM-IPS CLTR only performs well when the simulated clicks are drawn based on the PBM. The significance of these results is also in noticing that the current IPS unbiased CLTR papers all use PBM-IPS estimation in their experiments [2, 12, 17]. To fill in the gaps of the PBM-based CLTR performance on cascade-based clicks, we provide cascade model-based inverse propensity scoring (CM-IPS) and derive closed form formulas for click propensities in three widely

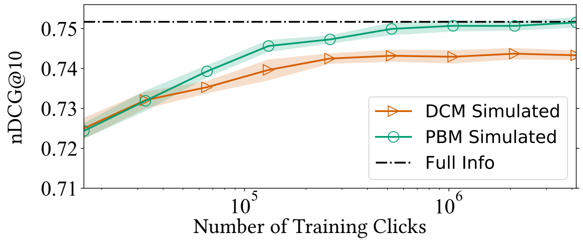

Figure 1: Performance of PBM-IPS CLTR on different sets of simulated clicks.

SIGIR '20, July 25–30, 2020, Virtual Event, China
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401299

Table 1: Notation.

Parameter	Description
x_j	representation of query, document pair at position j
$c_j \in \{0, 1\}$	click on result x_j at position j
$r_j \in \{0, 1\}$	relevance of result x_j at position j
$e_j \in \{0, 1\}$	examination of result x_j at position j
$\mathcal{X}_q$	ordered set of the results corresponding query q
q_S	the query of session S

used cascade-based click models, DCM [11], dynamic bayesian network (DBN) [6], and click chain model (CCM) [10]. We experimentally show the effectiveness of our derived propensity formulas for DCM.

To sum up, in this paper we are interested in the following research questions: (RQ1) Can a PBM-IPS CLTR effectively learn from clicks when the user click behavior is closer to cascade-based models? (RQ2) What are the cascade model-based propensity alternatives which are more suitable for IPS CLTR in the presence of cascade user click behavior? Table 1 summarizes the notation we use in the paper.

2 RELATED WORK

Click models. Click models model user behavior. Most click models factorize the click probability into two independent probabilities: the probability of examination and the probability of attractiveness (or relevance) [7]. In order to predict the examination probability, various probabilistic click models with different assumptions have been proposed. The *position-based click model* (PBM) assumes that the examination probability of a result only depends on its rank in the result list. There are several cascade-based models that assume that a user examines the results on the SERP linearly, from top to bottom, until she is satisfied with a result and abandons the session. See Section 3 for details. The true click model of a given (set of) click log(s) is not known, but unbiased CLTR requires the knowledge of the click propensities. Consequently, a click model is usually assumed for the click logs and the click propensities are estimated based on that assumed click model [12, 17].

Learning PBM Propensities. In LTR, it is common to optimize the sum of some performance metric only over relevant training documents [1, 2, 12]. However, in the click logs, the r_x are unknown. What is observed is c_x. According to the examination hypothesis, clicks appear on the relevant results that are also *examined.* Hence, the click signals are biased by the examination probability. To debias these signals, Joachims et al. [12] propose to use the so-called *inverse propensity scoring* (IPS) method:

$$\hat{\mathscr{L}}_S = \sum_{x_j \in \mathcal{X}_{q_S}} \frac{c_j^{(S)} \cdot \mathcal{L}_{x_j}}{P\left(E_j = 1 \mid q_S\right)}, \tag{1}$$

where $P\left(E_j = 1 \mid q_S\right)$ is the marginalized examination probability over all the sessions with the same query:

$$P(E_j = 1 \mid q) = E_{S|q_S=q}\left[P\left(E_j = 1 \mid S\right)\right]. \tag{2}$$

So far, LTR models dealing with click signals assume PBM for the clicks (in practice) and estimate the propensities based on this assumption [2, 12]. In PBM one can write:

$$P_{\text{PBM}}(E_j = 1 \mid q) = P_{\text{PBM}}(E_j = 1) = \theta_j \tag{3}$$

Existing CLTR work builds on the PBM assumption [2, 17]. But PBM is not necessarily the best fitting model in all situations.

Cascade Bias. Chandar and Carterette [4] discuss the idea that the existing CLTR methods do not consider the cascade bias, i.e. higher ranked relevancy dependent examination of items. They focus on counterfactual evaluation of rankers and show that, in presence of cascade bias, a *Context-Aware* IPS ranker has a higher Kendall's tau correlation with the full information ranker than that of a simple IPS. Though the basic ideas of [4] are the same as the current paper, there at least four important differences: (i) We propose closed form propensity formulas for cascade models, while they directly estimate the propensities using result randomization. (ii) We employ CM-IPS in CLTR to learn the ranker, as opposed to evaluating the rankers. (iii) Unlike them, we prove that the hidden click probabilities can be replaced with observed clicks without violating the unbiasedness. (iv) We use real query-document features for training our CLTR, whereas they only use fully simulated features in their experiments.

3 CASCADE MODEL-BASED PROPENSITY ESTIMATION

We derive recursive formulas for propensity estimation in popular cascade models based on clicks on a query session. We use CM-IPS to refer to an IPS method that uses these formulas. For each of DCM, DBN and CCM, we derive the examination probability at a position, based on the model parameters and the clicks over the previous positions. This exercise is not necessary for PBM since the propensities are the parameters themselves and examination at a position is independent of user behavior on other positions.

Before proceeding to specific propensity formulas for each click model, we need to rewrite the original IPS method proposed in [12] to make it more suitable for cascade-based models (CBM). Let us define the IPS per query loss as $\hat{\mathscr{L}}_q = \sum_{S|q_S=q} \hat{\mathscr{L}}_S$. In what follows we show that, in CBM, if the marginalized $P(E_j = 1 \mid q)$ in (1) is replaced with the session dependent probabilities $P(E_j = 1 \mid C_{<j})$, the per query loss will remain asymptotically unchanged. For brevity, we will drop the summation over positions as well as the $q_S = q$ condition.

$$\begin{aligned}\hat{\mathcal{L}}_q[j] &= \sum_S \frac{c_j^{(S)} \cdot \mathcal{L}_{x_j}}{P\left(E_j = 1 \mid q\right)} = \frac{\mathcal{L}_{x_j}}{P\left(E_j = 1 \mid q\right)} \sum_S c_j^{(S)} \\ &= N_q \cdot \frac{\mathcal{L}_{x_j} \cdot P(C_j = 1 \mid q)}{P\left(E_j = 1 \mid q\right)} = N_q \cdot P(R_j = 1) \cdot \mathcal{L}_{x_j} \\ &= \sum_S \frac{P(C_j = 1 \mid c_{<j}^{(S)})}{P(E_j = 1 \mid c_{<j}^{(S)})} \cdot \mathcal{L}_{x_j} \overset{(5)}{=} \sum_S \frac{c_j^{(S)} \cdot \mathcal{L}_{x_j}}{P(E_j = 1 \mid c_{<j}^{(S)})}\end{aligned} \tag{4}$$

where the last equality is empirically valid based on Eq. (5) below. In CBM, we can write for a general function g depending on the clicks before, and including, position j:

$$\sum_S P(C_j = 1 \mid c_{<j}) \cdot g(c_{\leq j})$$

$$= \sum_{c_{<j} \in \{0,1\}^{j-1}} |S_{c_{<j}}| \cdot P(C_j = 1 \mid c_{<j}) \cdot g(c_{\leq j})$$
$$\simeq \sum_{c_{<j} \in \{0,1\}^{j-1}} \sum_{S \in S_{c_{<j}}} c_j^{(S)} \cdot g(c_{\leq j}) = \sum_{S} c_j^{(S)} \cdot g(c_{\leq j}) \qquad (5)$$

where $S_{c_{<j}} = \{S \mid c_{<j}^{(S)} = c_{<j}\}$ and the third line is the empirical estimation of the second line. For CBM, the marginalized $P(E_j = 1)$ depends on the relevance probabilities of higher ranked results. But relevance is unknown during the CLTR and is yet to be learned. Instead of using EM algorithms to estimate relevance and marginalized examination probabilities, we propose to simply use the $P(E_j = 1 \mid C_{<j})$ which has been shown here to be empirically equivalent to the original loss.

Next we will derive separate formulas for $P(E_j = 1 \mid C_{<j})$ in DCM, DBN and CCM models.

DCM. In DCM, the user examines the results from top to bottom until she finds an attractive result, $P(E_{j+1} = 1 \mid E_j = 1, C_j = 0) = 1$. After each click, there is a position dependent chance that the user is not satisfied, $P(E_{j+1} = 1 \mid C_j = 1) = \lambda_j$. Therefore:

$$P_{\text{DCM}}(E_j = 1 \mid c_{<j}) = \prod_{i<j} (1 - c_i(1 - \lambda_i)) \qquad (6)$$

DBN. In DBN, there is another binary variable to model the user's satisfaction after a click. A satisfied user abandons the session, $P(E_{i+1} = 1 \mid S_i = 1) = 0$. An unsatisfied user may also abandon the session with a constant probability γ. Finally, after a click, the satisfaction probability depends on the document, $P(S_i = 1 \mid C_i = 1) = s_{x_i}$. Thanks to Eq. (4), we only need the session specific examination probability, which can be derived as follows:

$$P_{\text{DBN}}(E_j = 1 \mid c_{<j}) = \prod_{i<j} \gamma \cdot (1 - c_i \cdot s_{x_i}) \qquad (7)$$

CCM. The CCM is a generalization of DCM where continuing to examine the results before a click is not deterministic, $P(E_{j+1} = 1 \mid E_j = 1, C_j = 0) = \alpha_1$. The probability of continuing after a click is not position dependent, but relevance dependent, $P(E_{j+1} \mid C_j = 1) = \alpha_2(1 - R_i) + \alpha_3 R_i$. Similar to DCM we have:

$$P_{\text{CCM}}(E_j = 1 \mid c_{<j}) = \prod_{i<j} (\alpha_1 - c_i(\alpha_1 - \alpha_2(1 - R_i) - \alpha_3 R_i)). \qquad (8)$$

Parameter Estimation. In click model studies, parameter estimation is performed for each query over the sessions initiated by that query [7]: a low variance estimation requires a great number of sessions for each query. In CLTR studies, on the other hand, one uses features of query-document pairs in order to generalize well to tail queries [17]. We leave the feature-based parameter estimation of CBM as future work.

4 EXPERIMENTAL SETUP

Dataset. We use the Yahoo! Webscope [5] dataset for LTR with synthetic clicks. Our methodology follows previous unbiased LTR papers [2, 12]. We use binary relevance, considering the two most relevant levels as $r = 1$. We randomly select 50 queries from the training set and train a LambdaMART model over them to act as the initial ranker. The documents of all the queries are ranked using this initial ranker and the top 20 documents are shown to the virtual user. We remove all the queries which have no relevant documents in their top 20 documents. Consequently, the train and test sets have 11,474 and 4,085 queries, respectively. User behavior is modeled by PBM or DCM with various parameter assignments (see below). Sessions with at least one click are kept in the training set.

The reported results use $4M$ clicks for training, where the performance of CLTR is converged.

Click simulation. We use PBM and DCM for generating click data. For PBM, we use the widely used reciprocal formula for the examination probability [2, 12] (see Eq. (3)):

$$P_{\text{PBM}}(E_j = 1) = \theta_j = \left(\frac{1}{j}\right)^{\eta}, \qquad (9)$$

with $\eta \in \{0.5, 1, 2\}$.

For DCM, we use a similar formula for λ (see Eq. (6)):

$$P_{\text{DCM}}(E_{j+1} = 1 \mid C_j = 1) = \lambda_j = \beta \left(\frac{1}{j}\right)^{\eta}, \qquad (10)$$

where β and η are tuning parameters. We use $\beta \in \{0.6, 1\}$ and $\eta \in \{0.5, 1, 2\}$.

In both PBM and DCM cases, we used a noise (i.e. click on examined non-relevant items) with probability 0.05.

Experimental protocol. To investigate the effectiveness of PBM-IPS as well as CM-IPS, we try to train a CLTR over different sets of simulated click logs as explained above. We use DLA [2] to learn the click propensities based on the PBM assumption and MLE [11] to estimate λ's for DCM. Similar to other works on CLTR, we evaluate the rankings using explicit relevance judgements in the test set. We use nDCG at 10 to compare the rankings. We also report full-information results where the true relevance labels are used for training, i.e., the highest possible performance (skyline).

LTR implementation. Different LTR algorithms have been used for CLTR, including SVMRank [12], neural networks (NNs) [2, 3], and LambdaMART [17]. The differences are minimal [2]. We follow [2] and model the score function by a DNN, with the loss being softmax cross entropy. We use three layers with sizes $\{512, 256, 128\}$ and *elu* activation; the last two layers use dropout with a dropping probability of 0.1. Based on [15] we use a propensity clipping constant of 100 to avoid exploding variance.

5 RESULTS

CM-IPS effectiveness. In order to analyze the effectiveness of CM-IPS, we follow the protocol described in Section 4. Fig. 2 shows the performance of CM-IPS compared to PBM-IPS CLTR on numerous simulated click sets.

The x-axis shows the method used for simulating clicks: either "dcm_β_η" or "pbm_η" as explained in Section 4. The y-axis is the ranking performance of CLTR methods in terms of nDCG at 10. We see both PBM-IPS and CM-IPS improve the biased naïve LTR (indicated by "No IPS") in all cases. When using CM-IPS correction with oracle parameters for DCM simulated click sets (on the left of the vertical dashed line), the performance is consistently improved compared to PBM-IPS. All the differences are significant with $p < 0.0001$ except for dcm_1.0_2.0 which has $p < 0.05$. The reverse holds for PBM-IPS: it has a better performance for PBM simulated click sets (on the right of the vertical dashed line), compared to CM-IPS. In these datasets, the performance of PBM-IPS CLTR is

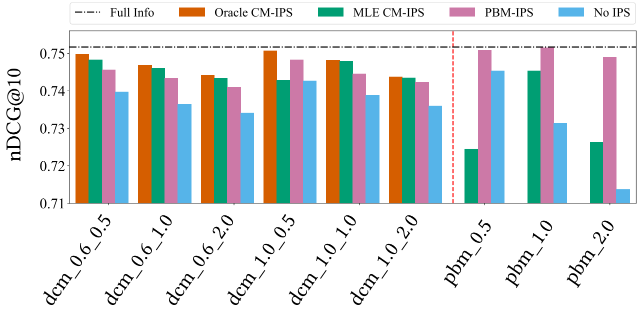

Figure 2: Performance of PBM-IPS and CM-IPS CLTR on different sets of simulated clicks. We repeated each experiment 15 times and report the mean value.

very close to the full-information case. These observations suggest that for a great variety of different parameter settings of DCM, the PBM-IPS correction cannot remove the bias, while CM-IPS can. More generally, Fig. 2 shows that when the click behavior and the correction method agree, the results are consistently better than the other case.

There is one practical issue that we leave as future work. This concerns the parameter estimation of DCM. The above discussions are valid when using the oracle parameter values for λ_j. It is worth mentioning that, unlike PBM, the parameters and the propensities are two different things in CM-IPS. The novelty of CM-IPS lies in computing the propensities given the parameters. We have tested maximum likelihood estimation (MLE) for estimating λ_j's [11]. Though the results with MLE CM-IPS are better than the PBM-IPS in most of the DCM simulated click sets, they are worse for dcm_1.0_0.5 and not significant for dcm_1.0_2.0 (Fig. 2). As argued in [7], MLE for DCM is based on a simplifying assumption which is not always true. Our findings coincide with this fact. Therefore, there is a need for CLTR-based algorithms for parameter estimation for CBM (see Section 3).

Method Selection. In order to choose between PBM- and CM-IPS for debiasing click logs, a measure that uses historical clicks to validate debiasing models is desired. For that, we use click log-likelihood. Click log-likelihood requires the click probabilities which are computed as the examination probability multiplied by the relevance probability. The examination probabilities are discussed in Section 3. For the relevance probabilities we use the output of our ranking function and pass it to different normalizing functions to have a valid probability range.

Our results on the sets presented previously in this section show the followings: (1) softmax always prefers CM-IPS (wrong selection for clicks close to PBM); (2) sigmoid always prefers PBM-IPS (wrong selection for clicks close to CBM); and (3) exponential min-max selects the better performing approach on the test set (correct selection in both cases). We leave more discussions in this regard as future work.

6 CONCLUSION

PBM is the default assumption in IPS-based CLTR. However, it is unable to properly model the cascade behavior of users. We raised the question of PBM effectiveness in IPS unbiased CLTR when users click behavior tends to CBM (RQ1). Through a number of experiments, we have answered our (RQ1) negatively: PBM-IPS is not helpful in CBM situations and, in our tested cases, there is a gap between its performance and the full-info case. This answer leads to a more important question: How to perform IPS correction for clicks close to CBM (RQ2). We provided CM-IPS, with closed form formulas for three widely used CBMs, namely DCM, DBN and CCM. We have shown the effectiveness of CM-IPS on the special case of DCM. Finally, we have given a short discussion about how to select between PBM- and CM-IPS only by looking at the clicks (and not using the true relevance labels).

CODE AND DATA

To facilitate the reproducibility of the reported results this work only made use of publicly available data and our experimental implementation is publicly available at https://github.com/AliVard/CM-IPS-SIGIR20.

ACKNOWLEDGMENTS

This research was supported by Elsevier and the Netherlands Organisation for Scientific Research (NWO) under project nr 652.002.001, and the Innovation Center for Artificial Intelligence (ICAI). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. 2019. Addressing Trust Bias for Unbiased Learning-to-Rank. In *WWW*. ACM, 4–14.

[2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *SIGIR*. ACM, 385–394.

[3] Qingyao Ai, Xuanhui Wang, Sebastian Bruch, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2019. Learning Groupwise Multivariate Scoring Functions Using Deep Neural Networks. In *SIGIR*. 85–92.

[4] Praveen Chandar and Ben Carterette. 2018. Estimating Clickthrough Bias in the Cascade Model. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1587–1590.

[5] Olivier Chapelle and Yi Chang. 2011. Yahoo! Learning to Rank Challenge Overview. In *Proceedings of the Learning to Rank Challenge*. 1–24.

[6] Olivier Chapelle and Ya Zhang. 2009. A Dynamic Bayesian Network Click Model for Web Search Ranking. In *WWW*. ACM, 1–10.

[7] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool Publishers.

[8] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-bias Models. In *WSDM*. 87–94.

[9] Artem Grotov, Aleksandr Chuklin, Ilya Markov, Luka Stout, Finde Xumara, and Maarten de Rijke. 2015. A Comparative Study of Click Models for Web Search. In *CLEF*. Springer, 78–90.

[10] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. 2009. Click Chain Model in Web Search. In *WWW*. ACM, 11–20.

[11] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient Multiple-click Models in Web Search. In *WSDM*. ACM, 124–131.

[12] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *WSDM*. ACM, 781–789.

[13] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.

[14] Harrie Oosterhuis and Maarten de Rijke. 2018. Differentiable Unbiased Online Learning to Rank. In *CIKM*. 1293–1302.

[15] Adith Swaminathan and Thorsten Joachims. 2015. Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization. *Journal of Machine Learning Research* 16, 1 (2015), 1731–1755.

[16] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *SIGIR*. ACM, 115–124.

[17] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. In *WSDM*. ACM, 610–618.