Efficient Off-Policy Evaluation of Content Blending in Station-Based Music Experiences

Chelsea Weaver
Amazon Music
Seattle, WA, USA
weachels@amazon.com

Arvind Balasubramanian
Amazon Music
Seattle, WA, USA
arvibala@amazon.com

Juan Borgnino
Amazon Music
Culver City, CA, USA
jborgnin@amazon.com

Ben London
Amazon Music
Seattle, WA, USA
blondon@amazon.com

Abstract

Audio streaming services, on both voice assistants and in visual apps, often field requests such as "play more like Foo Fighters." The service then returns a sequence of tracks that is both relevant to the request and personalized to the requester. While it is natural to evaluate the policies that produce these sequences in terms of customer engagement, such metrics do not assess their performance on other key business goals. We present our work to implement a *content blending strategy* to increase the prevalence of specific strategically-important content in these sequences, while minimizing harm to playback rates. In particular, we describe our efficient extension of *off-policy evaluation* to evaluate how blending impacts both overall engagement and the number of successful new release plays. We demonstrate how we used this work to choose blend rates for new policies so as to maximize overall engagement while preserving the new release metric baseline set by the current production policy. We also investigate the accuracy of these methods by comparing our estimates to online results.

CCS Concepts

• **Information systems → Learning to rank**; *Relevance assessment*.

Keywords

Off-Policy Evaluation, Cross Content Ranking, Recommendation, Music Recommendation, Sequencing

ACM Reference Format:
Chelsea Weaver, Arvind Balasubramanian, Juan Borgnino, and Ben London. 2025. Efficient Off-Policy Evaluation of Content Blending in Station-Based Music Experiences. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems (RecSys '25), September 22–26, 2025, Prague, Czech Republic.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3705328.3748114

1 Introduction

Amazon Music is an audio streaming service offering music, podcasts, and audiobooks. We focus here on *seed-based* music experiences. These are sessions of radio station-like playback initiated by a customer selecting a "seed" such as a song, artist, album, or playlist. The customer expects to receive a sequence of tracks that is both similar to the seed and reflective of their personal taste.

When evaluating the goodness of these experiences, we are often interested in multiple, possibly competing business metrics. Of primary interest is the customer's engagement with the stream, often measured by total listening time or average track completion. We also care about which *types* of music are consumed—and in particular, the prevalence of *new releases* ("NRs"). Promoting this content may help to address cold-start bias in the experience algorithms, since NRs tend to have less listening history. It also aids in music discovery and the overall freshness of the experience, which have been shown to promote long-term customer satisfaction [16]. In addition to NRs, there may be other content types to promote, such as less popular music or spoken audio. Thus, our goal is to increase overall engagement while also increasing engagement with these sub-categories.

We present a simple approach to increase the prevalence of a specific content type in seed-based playback queues while maintaining relevance as measured by average track playback length. Our primary contribution is an efficient method to estimate and tune the performance of the resulting algorithms offline, using the formalism of *off-policy evaluation* (OPE) from the contextual bandit literature. Additionally, we show how this method can be used to select blend rates for new policies so as to (conversely) maximize overall engagement while keeping content-specific engagement flat. Finally, we demonstrate the effectiveness of our approach by comparing these offline estimates to results from online experiments.

2 Model and Evaluation

At a high-level, Amazon Music produces station-based experiences in two steps: *Selection*, to select a few hundred candidates from the millions available in the catalog, followed by *Sequencing* (our focus here), to identify the (next) best track to play from the eligible candidates. The following subsections describe the sequencing model, and how we evaluate it, in greater detail.

RecSys '25, Prague, Czech Republic
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1364-4/25/09
https://doi.org/10.1145/3705328.3748114

2.1 Scoring model

To select the track that is played at each sequencing step, we use a model to score each candidate track, then pick the track with the highest score. We distinguish the scoring *model*, which assigns a score to each track, from the *policy*, which tells us how to select the track given the scores. We train the scoring model to predict a function of how long the customer will listen to the sequenced track. For training data, we use a subset of listening sessions as sequenced by (a slightly randomized version of) a production policy.

2.2 Offline policy evaluation

To measure the impact of modifying the scoring model (e.g., via new features or model architectures) prior to online testing, we use offline evaluation. This allows us to perform parameter tuning and model selection without affecting the customer experience. *Off-policy evaluation* (OPE) [1–4, 6–11, 13, 15, 17–23] is a methodology designed to estimate the performance of a new policy using data collected by a different policy (e.g., a randomized production policy). We focus on *importance weighting* approaches, which reweight the observed data to adapt to the distribution induced by a new policy. The most straightforward example of this is the *Inverse Propensity Scoring* (IPS) estimator [5], defined as $\mathrm{IPS}_r(\pi) := \frac{1}{n}\sum_i w_i r_i$, where π is the new policy, r_i is the ith logged instance of the *reward* r (representing the goodness of the sequencing decision), and w_i is a corresponding importance weight.

We focus on a variant of IPS called *Self-Normalized Inverse Propensity Scoring* (SNIPS) [20], which offers a preferable balance of bias and variance without requiring parameter tuning. The SNIPS estimator is defined as $\mathrm{SNIPS}_r(\pi) := \mathrm{IPS}_r(\pi)/\mathrm{CV}(\pi)$, where $\mathrm{CV}(\pi) := \frac{1}{n}\sum_i w_i$ is a multiplicative *control variate* [20]. Our primary reward is *engagement*, which we measure by the number of seconds the customer listens to the track, denoted r_{sec}. Each time we run OPE to evaluate a new scoring model, we obtain an estimate, $\bar{r}_{\mathrm{sec}}$, of the expected number of seconds each track will be listened to when the model is deployed in a new production policy. In general, we prefer policies with higher $\bar{r}_{\mathrm{sec}}$.

3 Multinomial blending

As mentioned in Section 1, it is not enough for us to optimize for $\bar{r}_{\mathrm{sec}}$; we are also tasked with increasing *the number of successful new release plays*, where "success" is defined as $r_{\mathrm{sec}} \geq 30$, and a track is newly-released if its release date is within 30 days. We can use OPE to evaluate new policies in terms of this metric by defining a new reward function: 1 if the selected track is a NR AND the customer listens to it for ≥ 30 seconds, and 0 otherwise. We refer to this reward as r_{nrs} and denote its average by $\bar{r}_{\mathrm{nrs}}$.

To increase $\bar{r}_{\mathrm{nrs}}$, given stringent time and resourcing constraints, we implemented the stochastic blending approach detailed in Algorithm 1. Dubbed *multinomial blending* by Lichtenberg et al. [12], this approach has been used successfully at Amazon Music to combine music and podcast recommendations. Our proposed method (detailed below) can be viewed as an extension of this work in which we present efficient hyper-parameter estimation.

For simplicity, we will assume a single content type of interest (though the method can be extended to multiple types) and that all candidate tracks—both of the type and not of the type—are scored by a single scoring model (see Section 2.1).

```
Inputs: A model $M$, blend rate $p$;
for each sequencing decision with set of candidate tracks $C$ do
    Have $M$ score the tracks in $C$;
    Flip a weighted coin with probability $p$ of heads;
    if coin flip is tails then
        Return the track with the highest score in $C$;
    else
        if $C$ contains NRs then
            Return highest-scoring NR track;
        else
            Return highest-scoring track;
end
```

Algorithm 1: Multinomial blending for seed-based sequencing

3.1 OPE for blended policies

Since we associate higher scores with higher playback rates, Algorithm 1 presents a clear trade-off between policy optimality—where we select the highest-scoring track, and selecting a NR—which may have a lower score. If the scoring model is accurate, increasing p will most likely result in an increase in $\bar{r}_{\mathrm{nrs}}$ but a decrease in $\bar{r}_{\mathrm{sec}}$.

Recall that we want to use blending to increase $\bar{r}_{\mathrm{nrs}}$. To determine which value of p to use, we can run OPE with different blend rates towards finding an acceptable decrease in $\bar{r}_{\mathrm{sec}}$ given the relative increase to $\bar{r}_{\mathrm{nrs}}$. The key advantage of our approach (presented below) is that, unlike traditional methods requiring $O(N)$ OPE runs to evaluate N different blend rates, we can estimate the effect of any blend rate using just a single OPE run, reducing the computational cost to $O(1)$.

Let π_{ENG} represent an "engagement-based" policy determined by selecting the highest-scoring track per a scoring model M (e.g., what we described in Section 2.1). Let π_{NR} represent a policy that always selects a NR when one is available, and when one is not, agrees with the selection of π_{ENG}. (In other words, π_{NR} always selects tracks according to the "heads" scenario in Algorithm 1, which is equivalent to setting $p = 1$.) Finally, let $\pi_{\mathrm{MNB}(p)}$ represent the blended policy defined in Algorithm 1. Noting that the IPS estimator and control variate are linear in the new policy, we can write the SNIPS estimator for a blended policy, with any $p \in [0, 1]$, as

$$\mathrm{SNIPS}_r(\pi_{\mathrm{MNB}(p)}) = \frac{(1-p)\cdot \mathrm{IPS}_r(\pi_{\mathrm{ENG}}) + p\cdot \mathrm{IPS}_r(\pi_{\mathrm{NR}})}{(1-p)\cdot \mathrm{CV}(\pi_{\mathrm{ENG}}) + p\cdot \mathrm{CV}(\pi_{\mathrm{NR}})}. \quad (1)$$

Because $\mathrm{CV}(\pi) \approx 1$ when the data is suitable for OPE [14], Eq. (1) shows a nearly linear relationship, controlled by p, between the estimated performance of the original policy and its "content type-specific" variant. Further, with these two estimates and their corresponding control variates, we can easily compute the estimated performance of the blended policy for any value of p. Eq. (1) holds for any reward r, and so we can use it to estimate $\bar{r}_{\mathrm{sec}}$ and $\bar{r}_{\mathrm{nrs}}$ of a blended policy $\pi_{\mathrm{MNB}(p)}$.

To illustrate the trade-off in metrics using our technique, we plot the results of Eq. (1) for an example policy, with $p \in [0, 1]$, and the offline rewards $\bar{r}_{\mathrm{sec}}$ (blue) and $\bar{r}_{\mathrm{nrs}}$ (orange) in Fig. 1. Since the two rewards do not use the same units (and are therefore not on the

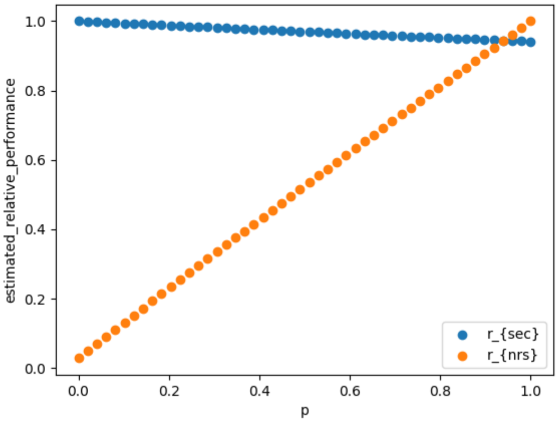

Figure 1: An illustration of the metric trade-off with blending. The expected rewards are divided by their respective maximum values (which happens when $p = 0$ for $\bar{r}_{\text{sec}}$ and when $p = 1$ for $\bar{r}_{\text{nrs}}$) in order to show the *relative* change in each reward as p varies.

same scale), we show the relative change in each reward, rather than their absolute values. As the blend rate p increases from 0 to 1, $\bar{r}_{\text{sec}}$ *slowly decreases from* its maximum value (achieved without blending) and $\bar{r}_{nrs}$ *quickly increases to* its maximum value (achieved with maximum blending).

3.2 Tuning the blend rate

The four quantities in Eq. (1) are model-dependent, i.e., they depend on M in Algorithm 1. If we try to improve or simply retrain the current model, the resulting policy may produce different reward estimates for the same blend rate. For example, if a new feature improves the model's ability to recognize favorable NRs (increasing $\text{IPS}_{r_{\text{nrs}}}(\pi)$), then we may choose to use a smaller value of p so that we can maximize engagement while maintaining performance on NRs.

Let π_{prod} and $\tilde{\pi}$ denote the prod policy and a new policy, respectively. The goal is to choose p for the new blended policy $\tilde{\pi}_{\text{MNB}(p)}$ so that its NR performance is no worse than that of the production policy. We can use Eq. (1) to solve for a blend rate, $p^\star$, such that $\text{SNIPS}_{r_{\text{nrs}}}(\tilde{\pi}_{\text{MNB}(p^\star)}) \geq \text{SNIPS}_{r_{\text{nrs}}}(\pi_{\text{prod}})$. This is satisfied by

$$p^\star = \frac{\text{SNIPS}_{r_{\text{nrs}}}(\pi_{\text{prod}}) \cdot \text{CV}(\tilde{\pi}_{\text{ENG}}) - \text{IPS}_{r_{\text{nrs}}}(\tilde{\pi}_{\text{ENG}})}{\text{SNIPS}_{r_{\text{nrs}}}(\pi_{\text{prod}}) \cdot \Delta(\text{CV}) - \Delta(\text{IPS}_{r_{\text{nrs}}})}, \tag{2}$$

$$\text{where} \quad \Delta(\text{IPS}_{r_{\text{nrs}}}) = \text{IPS}_{r_{\text{nrs}}}(\tilde{\pi}_{\text{ENG}}) - \text{IPS}_{r_{\text{nrs}}}(\tilde{\pi}_{\text{NR}}) \tag{3}$$

$$\text{and} \quad \Delta(\text{CV}) = \text{CV}(\tilde{\pi}_{\text{ENG}}) - \text{CV}(\tilde{\pi}_{\text{NR}}). \tag{4}$$

The resulting blended policy, $\tilde{\pi}_{\text{MNB}(p^\star)}$, will have the same NR estimate as π_{prod}. Further, we can use Eq. (1) (with r_{sec} and $p^\star$) to estimate the engagement of $\tilde{\pi}_{\text{MNB}(p^\star)}$.

Table 1: Estimated impact on $\bar{r}_{\text{sec}}$ and $\bar{r}_{\text{nrs}}$, and online metrics $\Sigma(r_{\text{sec}})$ and $\Sigma(r_{\text{nrs}})$, for policies with the same scoring model at various MNB blend rates. All impact percentages are relative to the policy without blending.

	SNIPS estimates		online	
p	$\bar{r}_{\text{sec}}$	$\bar{r}_{\text{nrs}}$	$\Sigma(r_{\text{sec}})$	$\Sigma(r_{\text{nrs}})$
0.01	-0.05%	30.08%	-0.05%	4.05%
0.03	-0.20%	88.72%	-0.15%	11.53%
0.05	-0.36%	148.12%	-0.20%	17.95%
0.10	-0.66%	294.74%	-0.30%	31.49%
0.25	-1.63%	732.33%	-0.71%	61.81%

4 Experiments

Our empirical study consists of two parts: first, we investigate the accuracy of offline estimates w.r.t. predicting online outcomes; then, we explore tuning the blend rate as described in Section 3.2.

4.1 Using OPE to find candidate blend rates

The first multi-column of Table 1 reports the estimated impact (via SNIPS estimates) to $\bar{r}_{\text{sec}}$ and $\bar{r}_{\text{nrs}}$ for several blend rates p, relative to the unblended policy. As expected, as p increases, $\bar{r}_{\text{sec}}$ decreases while $\bar{r}_{\text{nrs}}$ increases. Further, $\bar{r}_{\text{nrs}}$ increases *relatively* faster than $\bar{r}_{\text{sec}}$—e.g., blending at 25% increases $\bar{r}_{\text{nrs}}$ by 732% while reducing $\bar{r}_{\text{sec}}$ by only 1.63% (compared to no blending). While this might seem like an obvious win, one must remember that these metrics are of different value to the business, and even small changes to $\bar{r}_{\text{sec}}$ may be meaningful.

To assess the accuracy of the offline estimates, we ran a 14-day online experiment with the unblended policy ($p = 0$) as the control baseline, and the five blended variants as treatments. All treatments use the same scoring model as in the offline results (following the approach described in Section 2.1). The "online" columns of Table 1 show the relative change in the *average customer's total listening time* (to the seed-based experience), denoted $\Sigma(r_{\text{sec}})$, and the *average customer's successful NR play count*, denoted $\Sigma(r_{\text{nrs}})$. It is expected that the relative performance between treatments differs from the offline results, given the different metrics (average track-level engagement, $\bar{r}$, vs. average customer-level engagement, $\Sigma(r)$). However, we do find that the online results are *directionally* aligned with our offline estimates.

4.2 Effectiveness of tuning the blend rate

In this section, we demonstrate using Eq. (2) to find appropriate blend rates for new policies. As discussed in Section 3.2, our goal is to maximize overall engagement while maintaining performance on NRs, relative to a production policy. More specifically, we begin with four new (initially unblended) policies $\tilde{\pi}_1, \ldots, \tilde{\pi}_4$, each derived from its own scoring model, and a blended production policy $\pi_{\text{MNB}(p_{\text{prod}})}$. Then for each $\tilde{\pi}_i$, we use Eq. (2) to find $p_i^\star$ so that the NR performance of each resulting *blended* policy $\tilde{\pi}_{i,\text{MNB}(p_i^\star)}$, $i = 1 \ldots 4$, will be close to the NR performance of $\pi_{\text{MNB}(p_{\text{prod}})}$.

We present our results in in Table 2. The first multi-column shows the offline performance of the unblended treatment policies

Table 2: Estimated impact before and after blending, as well as online observed metrics, for candidate policies (treatments) with p tuned per Eq. (2) to match the control policy's NR performance. All results are relative to control.

	offline $\tilde{\pi}_i$		offline $\tilde{\pi}_{i,\text{MNB}(p_i^\star)}$		online $\tilde{\pi}_{i,\text{MNB}(p_i^\star)}$	
treat.	$\overline{r}_{\text{sec}}$	$\overline{r}_{\text{nrs}}$	$\overline{r}_{\text{sec}}$	$\overline{r}_{\text{nrs}}$	$\Sigma(r_{\text{sec}})$	$\Sigma(r_{\text{nrs}})$
$i = 1$	-1.19%	268.62%	-0.77%	4.55%	-0.38%	-2.75%
$i = 2$	-2.21%	280.46%	-1.74%	0.00%	-1.58%	-6.48%
$i = 3$	-2.02%	-1.03%	-1.99%	2.27%	-1.62%	0.95%
$i = 4$	-0.22%	41.06%	-0.15%	-2.44%	0.28%	0.08%

relative to an *unblended* version of the control policy, and the second multi-column shows results after blending.[1] Before blending, for example, $\tilde{\pi}_1$ is expected to slightly underperform the unblended version of control in terms of $\overline{r}_{\text{sec}}$ (-1.19%), but we expect it to excel in terms of $\overline{r}_{\text{nrs}}$ (+268.62%). Per Eq. (1), this means we can likely blend this policy less than we blend control, i.e., we can set $p < p_{\text{prod}}$, and still achieve (roughly) the same $\overline{r}_{\text{nrs}}$. However, even though less blending means we can trade-off less overall engagement (see Fig. 1), $\tilde{\pi}_{1,\text{MNB}(p_1^\star)}$ is still expected to have lower $\overline{r}_{\text{sec}}$ than $\pi_{\text{MNB}(p_{\text{prod}})}$ (-0.77%). This is because the results of Eq. (1) depend specifically on the given policy/scoring model (hence the need to readjust the blend rate for new policies), and the fact that, for these policies, $\overline{r}_{\text{sec}}$ increases much more slowly than $\overline{r}_{\text{nrs}}$ decreases. We can interpret this as the reward of the highest-scoring NR being generally close to that of the overall highest-scoring item, and so selecting a NR results in only a small deviation from optimal expected engagement.

The "online" columns in Table 2 contain the resulting metrics from a 14-day online experiment. While we tuned the blend rates to produce similar $\overline{r}_{\text{nrs}}$ across treatments (towards making $\Sigma(r_{\text{nrs}})$ identically 0), this metric actually varies significantly. Though results for $i = 4$ are as expected, the blend rates for $i = 1, 2$ were too small, and too big for $i = 3$. There is also some directional misalignment between the expected blended performance and the online results—e.g., we expected $i = 4$ to underperform the prod policy in terms of both engagement and NRs, but online we see a small increase in both metrics. We think it likely that both observations are due to the high variance of SNIPS estimates for $\overline{r}_{\text{nrs}}$ and/or the difference between offline and online metrics.

5 Conclusions

In this paper, we proposed an efficient method for evaluating and tuning recommendation policies that probabilistically blend exposure of different content types. The method combines reward estimates for the unblended and "max" blended policies originating from the same scoring model, illustrating a clear trade-off between the unblended policy's original objective and an alternative objective (in our case, overall engagement vs. NR engagement). We also derived a closed-form expression to tune the blend rate in order to match a given baseline performance. Online experiments confirmed that the approach yields directionally accurate predictions but that they are sensitive to variance. To mitigate this, we can try simply collecting more evaluation data for OPE (via longer time windows and/or at higher sample rates). We can also modify our randomized data collection pipeline, which currently branches off the production policy before blending, to instead source from the blended policy, thereby providing more instances of selected NRs and hence more positive NR rewards to be leveraged by OPE.

[1] While our goal was to choose p to make the values in the blended $\overline{r}_{\text{nrs}}$ column all zeroes, this was not fully achieved, due to truncating the blend rates found using Eq. (2) to the nearest tenth.

References

[1] Aman Agarwal, Soumya Basu, Tobias Schnabel, and Thorsten Joachims. 2017. Effective Evaluation Using Logged Bandit Feedback from Multiple Loggers. *Knowledge Discovery and Data Mining* (2017).

[2] Miroslav Dudik, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *International Conference on Machine Learning.*

[3] Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. 2018. More Robust Doubly Robust Off-policy Evaluation. In *International Conference on Machine Learning.*

[4] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline A/B Testing for Recommender Systems. In *Web Search and Data Mining.*

[5] Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *J. Amer. Statist. Assoc.* 47, 260 (1952), 663–685.

[6] Nan Jiang and Lihong Li. 2016. Doubly Robust Off-policy Value Evaluation for Reinforcement Learning. In *International Conference on Machine Learning.*

[7] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Web Search and Data Mining.*

[8] Nathan Kallus. 2018. Balanced Policy Evaluation and Learning. In *Neural Information Processing Systems.*

[9] John Langford, Alexander Strehl, and Jennifer Wortman. 2008. Exploration scavenging. In *International Conference on Machine Learning.*

[10] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Web Search and Data Mining.*

[11] Lihong Li, Remi Munos, and Csaba Szepesvári. 2015. Toward Minimax Off-policy Value Estimation. In *Artificial Intelligence and Statistics.*

[12] Jan Malte Lichtenberg, Giuseppe Di Benedetto, and Matteo Ruffini. 2024. Ranking across different content types: The robust beauty of multinomial blending. In *The ACM Conference Series on Recommender Systems.*

[13] Anqi Liu, Hao Liu, Anima Anandkumar, and Yisong Yue. 2019. Triply Robust Off-Policy Evaluation. *CoRR* abs/1911.05811 (2019).

[14] Ben London and Thorsten Joachims. 2022. Control Variate Diagnostics for Detecting Problems in Logged Bandit Feedback. In *CONSEQUENCES+REVEAL Workshop – RecSys.*

[15] A. Rupam Mahmood, Hado van Hasselt, and Richard Sutton. 2014. Weighted importance sampling for off-policy learning with linear function approximation. In *Neural Information Processing Systems.*

[16] Rishabh Mehrotra. 2021. Algorithmic Balancing of Familiarity, Similarity, & Discovery in Music Recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 3996–4005. https://doi.org/10.1145/3459637.3481893

[17] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *International Conference on Machine Learning.*

[18] Alex Strehl, John Langford, Lihong Li, and Sham Kakade. 2010. Learning from Logged Implicit Exploration Data. In *Neural Information Processing Systems.*

[19] Yi Su, Lequn Wang, Michele Santacatterina, and Thorsten Joachims. 2019. CAB: Continuous Adaptive Blending for Policy Evaluation and Learning. In *International Conference on Machine Learning.*

[20] Adith Swaminathan and Thorsten Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *Neural Information Processing Systems.*

[21] Phillip Thomas and Emma Brunskill. 2016. Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning. In *International Conference on Machine Learning.*

[22] Nikos Vlassis, Aurelien Bibaut, Maria Dimakopoulou, and Tony Jebara. 2019. On the Design of Estimators for Bandit Off-Policy Evaluation. In *International Conference on Machine Learning.*

[23] Yu-Xiang. Wang, Alekh Agarwal, and Miroslav Dudík. 2017. Optimal and Adaptive Off-policy Evaluation in Contextual Bandits. In *International Conference on Machine Learning.*

A Author biographies

Chelsea Weaver is a Senior Applied Scientist at Amazon Music, where she specializes in natural language understanding and music recommendation. She has a Ph.D. in mathematics with a focus on face recognition from the University of California, Davis, where she was advised by Naoki Saito.

Arvind Balasubramanian is a Senior Applied Scientist at Amazon Music, focusing on personalized recommendations and dynamic content sequencing. He previously developed fraud prevention models at Amazon, and pricing algorithms at Expedia. He holds a Ph.D. in Computer Science from the University of Texas at Dallas, where his research focused on time series pattern mining and machine learning applications in the healthcare domain.

Juan Martin Borgnino is a Senior Applied Scientist at Amazon Music, based in Los Angeles, California. He has worked on causal inference, natural language understanding, and personalization at both Amazon Prime Video and Amazon Music. He holds a master's degree in machine learning from Columbia University.

Ben London is a Principal Scientist at Amazon Music. His research explores machine learning theory and algorithms, with a focus on generalization guarantees, recommendation, and learning from logged bandit feedback. He was co-organizer of the NeurIPS 2019 Workshop on ML with Guarantees; an area chair for ICML (2020, 2022), NeurIPS (2020, 2021, 2025) and ICLR (2024); a senior PC member for IJCAI (2020); and Industry Co-chair for RecSys (2024). He earned his Ph.D. in 2015 at the University of Maryland.