DOI: 10.1609/aaai.12014

Recommendations as treatments

Thorsten Joachims[1] | **Ben London**[2] | **Yi Su**[1] | **Adith Swaminathan**[3] | **Lequn Wang**[1]

[1] Cornell University

[2] Amazon

[3] Microsoft Research

Correspondence
Thorsten Joachims, Cornell University, Department of Computer Science, 418 Gates Hall, 58453 Ithaca, NY, USA.
Email: thorsten.joachims@cornell.edu

Funding information
NSF, Grant/Award Numbers: IIS-1901168, IIS-2008139; Bloomberg Fellowship

Abstract

In recent years, a new line of research has taken an *interventional* view of recommender systems, where recommendations are viewed as actions that the system takes to have a desired effect. This interventional view has led to the development of *counterfactual* inference techniques for evaluating and optimizing recommendation policies. This article explains how these techniques enable unbiased offline evaluation and learning despite biased data, and how they can inform considerations of fairness and equity in recommender systems.

INTRODUCTION

What do recommender systems have in common with your favorite medical drama, or with the trusted physician you see for your real-life medical problems?

Dr. House: She has gone from the 25th weight percentile to the 3rd in one month. Now, I'm not a baby expert, but I'm pretty sure they're not supposed to shrink.
Mother: Well there's this diet we put her on when she stopped breast feeding …
Father: But it's healthy, um, raw food. We're vegans. Almond milk, tofu, uh, vegetables …
Dr. House: Raw food … If only her ancestors had mastered the secret of fire. Babies need fat, proteins, calories. Less important: sprouts and hemp. Starving babies is bad and illegal in many cultures. I'm having her admitted. (Fox 2004)

If you are a fan of "House MD", the commonality is not the witty dialogue or the charismatic characters, but (hopefully) your real-life doctor does not aspire to those standards either. Instead, at an abstract level, both recommender systems and medical professionals need to reason about interventions and counterfactuals. The interventions can be under our control (e.g., admitting the baby, or recommending a movie) or they can be chosen by others (e.g., the baby's diet, or a user-selected movie), but in either case we only see the factual outcome under the chosen treatment; we do not see the counterfactual outcomes, namely what would have happened under a different treatment. It is this reasoning about treatments and counterfactuals that is common to both settings, and we argue that it provides a formal basis for recommender systems. In particular, this viewpoint crystallizes why recommendation is fundamentally different from regular supervised learning, and highlights the fact that recommendation is primarily about *acting* — not prediction. Accordingly, recommender systems should be viewed as *policies* that decide what interventions to make in order to optimize a desired outcome.

In this article, we explain why an interventional view of recommendation provides a rigorous framework for thinking about recommender systems — enabling new insights both at a technical level for evaluation and learning, as well as at a conceptual level when we reason about the future of recommender systems. In some respects, the view of recommender systems as autonomous systems that act through their recommendations is already part of common industry practice. For example, A/B tests are widely recognized as the gold standard for evaluating recommender systems, and they are functionally equivalent to a controlled randomized trial in medicine. However, we argue

that the connection between recommender systems and causal inference runs much deeper, leading to a rigorous foundation for the field that produces new algorithms with provable guarantees.

One such area is **offline A/B testing,** which is also known as *off-policy* evaluation in the literature. The offline A/B testing methodology developed over the past years can overcome the main drawbacks of online A/B tests; namely, that they are slow, expensive, and pose a risk to the user experience. Through the development of counterfactual estimators for off-policy evaluation (Agarwal et al. 2017; Gilotte et al. 2018; Li et al. 2011; Su et al. 2019), it has become possible to quickly test new recommendation policies using data that were logged in the past. We outline in the following section how, and under which conditions, these estimators provide unbiased estimates of how a new policy would have performed in an online A/B test, if it had been used instead of the policy that logged the historical data.

A second area is **machine learning for recommendation** using logged data, also called off-policy learning. Once we know how to obtain unbiased estimates for new policies using logged interaction data from some historic policy—which are often in abundance—we can design learning algorithms that search for the best policy in hindsight. Such counterfactual learning (i.e., which policy would have performed best, if we had used it instead of the policy that logged the data) is a new form of empirical risk minimization. Most importantly, counterfactual learning methods (Bottou et al. 2013b; Joachims, Swaminathan, and Schnabel 2017; Swaminathan and Joachims 2015a) overcome one of the key problems of offline learning for recommender systems; namely, that conventional offline methods optimize an objective that is different from what is measured in an online A/B test. In contrast, counterfactual learning methods directly optimize the same online performance measure, and we explain in the following how they bridge the gap between online and offline.

Finally, let us come back to the introductory example. If your initial reaction was that doctors make life-altering decisions while recommender systems do not, we urge you to think again. Granted, an individual recommendation for a movie, a track, or a product pales in comparison to the decisions doctors need to make every day. But many small decisions can add up to have a considerable impact. Furthermore, the technologies and methods used in recommender systems are being adapted to a wide array of new applications, where they inform potentially life-altering decisions in criminal justice (Angwin et al. 2016), the allocation of government benefits (Nezik 2019), and university admission (Waters and Miikkulainen 2014). There is now growing discussion around **fairness in recommender systems**, although no agreed-upon definition or standard of fairness exists yet. We argue below that an interventional view of recommender systems provides a promising framework for reasoning about fairness and for addressing selection biases in some applications. We also discuss what this means for the future of recommender systems in the final section of this article, as we need to engage with a broad range of social sciences and methodologies.

OFFLINE A/B TESTING

The most straightforward way to evaluate a recommendation policy is to simply expose it to users and see how it performs. Typically, there will already be a policy in place—a so-called *baseline*—with which we wish to compare; that is, we want to know whether, and how much, the new policy is better than the existing one. To measure this *treatment effect*, we can conduct an A/B test in which we randomly partition the user base into two disjoint populations, expose one population to the *control* (baseline policy), and the other to the *treatment* (new policy).

Online A/B testing is generally considered the gold standard for evaluating a recommendation policy. By letting the policy interact directly with users, we can observe feedback for any decision it makes, and (with proper randomization of treatment groups) obtain an unbiased measurement of treatment effect by aggregating feedback within each group. Because of its simplicity and statistical accuracy, many of today's large technology companies use online A/B testing as a final vetting process for new features and improvements.

Though online testing is appealing from a statistical perspective, it is costly from almost every other perspective. Conducting an online test requires developing an idea into a robust, "production-ready" system that can handle web-scale traffic. This process takes time and resources, even for the simplest of ideas. Beyond the immediate financial cost of preparing an online test, there is also an opportunity cost to running a test; by exposing users to an experimental policy, which could potentially be worse than the existing policy, we risk providing an inferior user experience. This risk is compounded by the fact that online tests generally need to run for a matter of days or weeks to achieve statistical significance; during this time, we may provide a suboptimal experience to users if the test is ultimately unsuccessful.

The above costs can be mitigated by testing new ideas in *offline* A/B tests. In an offline setting, we can try risky ideas without affecting the user experience. Moreover, we can work faster and cheaper because the overhead of offline experimentation is lower; we do not need to develop production-ready code or deal with the complexity of an online service. Of course, we need data to evaluate on.

Fortunately, we may have already collected millions (or even billions) of user interactions with an existing system, or from previous online A/B tests. By recycling these data to evaluate new treatments offline, we effectively amortize the cost of running online tests. Furthermore, accurate offline estimates would allow us to employ traditional machine learning methodologies, such as *cross-validation* or *backtesting*.

However, getting unbiased evaluations in the offline setting requires more care than in the online setting. In an online A/B test, a simple average of observations is an unbiased estimator; while in an offline A/B test, a simple average would be biased. The source of this bias comes from the fact that we are estimating the performance of a *target policy* using data that were collected under a different policy, the so-called *logging policy*. The distribution of actions that the logging policy took can influence any estimate we make using the logged data. If the logging policy selected actions uniformly at random, then it would not introduce any bias. Unfortunately, this is an unrealistic expectation, since exposing users to such a policy could have significant negative consequences. A more likely scenario is that the logging policy is non-uniformly stochastic; that it favors certain actions over others—hopefully, in a way that is beneficial to users—but still with some randomness. Because the distribution of actions is non-uniform, we observe outcomes for certain actions more than others; and when we use these observations to evaluate a new policy, our estimates—and the decisions we make using these estimates—will be biased by whichever actions the logging policy favored.

To make this issue more concrete, consider a simple movie recommender for a video streaming service. Imagine that, at each visit, the recommender selects a single movie to suggest to the users after they log in. Assume that we have collected data from this recommender using a stochastic policy whose distribution is weighted towards more popular movies, such as blockbuster superhero movies. When we use these data offline to evaluate new policies, we may mistakenly conclude that policies that favor superhero movies are better, since these movies are over-represented. This is the so-called "rich-get-richer" effect, wherein things that are already very successful become more so due to their ubiquity. At the same time, we may miss opportunities to provide more personalized recommendations due to insufficient data in certain niches. For instance, consider a user whose favorite genre is Scandinavian thrillers. They might also appreciate superhero movies, so recommending them an installment from the *Avengers* franchise is a safe bet. Yet that would be suboptimal; they would much prefer *Midsommar*, a horror film set in Sweden.

We pause here to note that traditional metrics used for supervised learning—such as the accuracy of click prediction—do not account for these inherent biases. We could bolster our offline metrics with ad hoc measures—such as diversity or popularity of recommended content—to ameliorate failure modes like in the example above. However, offline comparisons made with these metrics may not reflect the outcome of online A/B tests. In the next section, we will revisit this issue and derive unbiased metrics that are composable with supervised learning.

We now describe a principled approach, borrowed from causal inference, that can correct the biases that occur during data collection, and thereby yield unbiased estimates of the metrics that we care about. This technique addresses the counterfactual question of how well a new policy would have performed if it had been used instead of the policy that logged the historical data. For this reason, such estimators are often called *counterfactual estimators*.

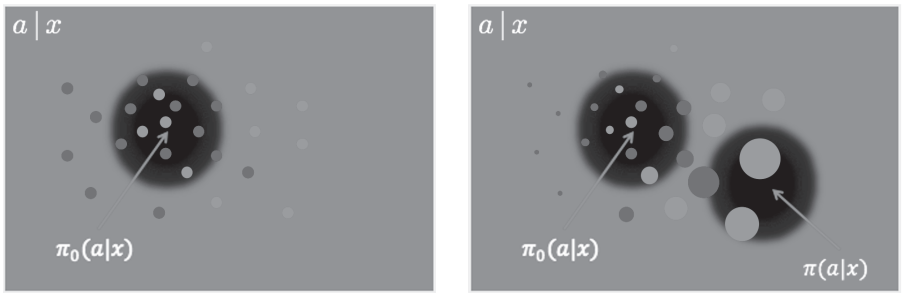
To present the technique, it will be helpful to first introduce some light notation and terminology. Suppose each moment in time (indexed by i) is represented by some *contextual attributes*, which we denote by x_i. These could describe the current user, time of day, etc. The logging policy, denoted as π_0, receives x_i and responds by sampling an *action*, a_i; for example, a movie that it presents to the user. The user in turn responds by either accepting or rejecting the recommendation; we quantify this response with a variable, $r_i = r(x_i, a_i)$, which is typically called a *reward*. Given a logged dataset of n such interactions, our goal is to evaluate the expected *utility*,

$$U(\pi) = \mathbb{E}_x \mathbb{E}_{a\sim\pi(\cdot|x)}[r(x,a)] \tag{1}$$

of a new *target* policy, π. This policy induces a conditional probability distribution over actions (given contexts), denoted as $\pi(a|x)$, and in the simplest case it picks one particular action for each context with probability 1. A flawed estimator for $U(\pi)$ would be to simply multiply the logged rewards by the probability under the target policy: $\pi(a_i|x_i)r_i$. This weighting suffers from the bias issues described above, since the distribution of logged actions is already weighted by the logging policy, π_0. To correct this bias, we need to divide each reward by the *propensity*$\pi_0(a_i|x_i)$ (i.e., the probability of the logged action, a_i, under the logging policy). This yields the following unbiased, counterfactual estimator:

$$\hat{U}(\pi) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)} r_i. \tag{2}$$

This *inverse propensity score* (IPS) estimator (Horvitz and Thompson 1952; Rosenbaum and Rubin 1983) can be



(A) Dataset before IPS weighting (B) Dataset after IPS weighting

FIGURE 1 The effect of IPS weighting. The gray rectangle represents the action space (conditioned on a given context), wherein each dot is a logged action, whose color indicates "good" (green) or "bad" (red) reward. Dark gray shading represents the logging and target policies. Dots in the figure on the right are resized to represent the importance weights, $\pi(a_i|x_i)/\pi_0(a_i|x_i)$

viewed as a form of *Monte Carlo simulation* (i.e., approximating an integral by sampling) wherein the goal is to estimate an expectation under a target distribution using samples from a (different) source distribution. It is essentially like tossing a biased coin to estimate the expected number of heads under a differently biased coin.

Figure 1 illustrates the effect of IPS weighting. On the left is a visualization of the dataset, wherein the action space (conditioned on some context) is represented by a two-dimensional plane. Each dot represents an action that was sampled by the logging policy, which is represented by a shaded circle. Dots that are closer to the center of the logging policy are more likely under its sampling distribution. Each dot is colored based on how the user responded; green indicates "good" reward (e.g., the user watched the movie to completion), while red indicates "bad" reward (e.g., the user rejected the recommendation). On the right, we visualize the re-weighted dataset using IPS with a given target policy. Each dot has been resized based on the ratio $\pi(a_i|x_i)/\pi_0(a_i|x_i)$, which is referred to as an *importance weight*. Dots that are close to the logging policy but far from the target policy are small, while dots that are far from the logging policy but close to the target policy are large. Thus, IPS emphasizes events that were rare under the logging policy but are likely under the target policy. These "under-explored" actions are the most important, because they could have a large impact on the target policy's expected reward.

If the logging policy is sufficiently randomized, such that it has a non-zero probability of selecting any action that has a non-zero probability under the target policy, then the IPS estimator is statistically *unbiased*; meaning, its expected value (over realizations of the dataset) is equal to the target policy's true expected utility. This is precisely the quality that we want from an estimator because it tells us how a new policy will perform online, thereby allowing us to make decisions about which policies we deploy. Unfortunately, this advantage comes with several challenges. First, the logging policy needs to be randomized, which requires a trade-off between exploration (better data for offline A/B tests) and exploitation (less risk for the user experience). Second, IPS requires us to compute and log propensities, which necessitates additional logging infrastructure and logging policies for which the propensities are easy to compute. Alternatively, it may be possible to *estimate* the propensities—either by Monte Carlo simulation or supervised learning—but this approximation usually introduces some bias. Last but not least, the primary issue with IPS is that very small propensities result in very large importance weights, so the estimator can have a large variance. High variance reduces the *effective sample size*; meaning, more data are needed to achieve a reliable estimate.

The IPS estimator's variance problem has received much attention, and many solutions have been proposed. Arguably, the simplest solution is to truncate the propensities (or importance weights), such that they are "clipped" to a reasonable range (Ionides 2008). Alternatively, one could simply omit data records for which the propensities are too small (Bottou et al. 2013a). Other methods use multiplicative *control variates* to normalize the importance weights such that they sum to one (Swaminathan and Joachims 2015b). There are also methods that combine IPS with a direct estimation of the reward (utility) function, so that IPS is only used to correct mistakes in the predicted reward (Dudík, Langford, and Li 2011; Liu et al. 2019; Vlassis et al. 2019, Su et al. 2019). All of these methods effectively reduce the variance of the estimator — however, this usually comes at the cost of introducing bias. This trade-off between bias and variance is central to offline evaluation. Formal analyses have quantified this trade-off, and we can use these to derive confidence intervals (Strehl et al. 2010a; Thomas, Theocharous, and Ghavamzadeh 2015) to account for statistical error.

With unbiased (or minimally biased) utility estimators, we can obtain reliable offline estimates of how a new

policy will perform when deployed online. Yet, where does the new policy come from? In the next section, we discuss how to *learn* policies from logged data, so as to maximize expected online utility. We can also optimize auxiliary criteria, such as ensuring that the recommendations are equitable to both users and content providers. We discuss these fairness considerations in a later section of this article. Finally, we also discuss practical considerations and challenges associated with offline evaluation and learning.

OFFLINE LEARNING FOR RECOMMENDATION

The offline A/B tests and counterfactual estimators discussed in the previous section give us a way to estimate the performance of many policies without having to field them. This leads to a natural way to compare various policies and search for the best one in hindsight.

To formalize this strategy for *policy learning*, suppose we are given a class of policies, Π, in which we can search for a new, better policy. This class could be that of simple linear models or complicated neural networks. We want to find a specific policy, $\pi \in \Pi$, that gives us the highest expected utility, $U(\pi)$. Unfortunately, $U(\pi)$ is unknown, but we can obtain an unbiased estimate, $\hat{U}(\pi)$, using a counterfactual estimator, such as the IPS estimator described in the previous section. By optimizing this estimate, we perform *counterfactual risk minimization* (CRM), which is analogous to the idea of *empirical risk minimization* from supervised machine learning. Specifically, we search for the policy that gives us the highest estimated utility, measured by $\hat{U}(\pi)$:

$$\arg\max_{\pi \in \Pi} \hat{U}(\pi). \tag{3}$$

While the IPS estimator (Equation (1)) is a common choice for $\hat{U}(\pi)$, other counterfactual estimators can be used as well, such as *self-normalized IPS* (Swaminathan and Joachims 2015b) or *doubly-robust* (Dudík, Langford, and Li 2011). It is easy to see that improved counterfactual estimators allow more reliable comparisons between different policies in Π, and thus better learning performance (Strehl et al. 2010b).

Let us illustrate the CRM approach by considering a movie recommendation problem in which we want to recommend one movie in the top banner of a website. In this setting, the context x is a feature vector encoding all the personal information and viewing history of the current user, while each action a (i.e., movie recommendation) is also described by a feature vector. For simplicity, we assume that the feedback r is a binary indicator (e.g., whether the user watched the recommended movie to the end), but it can be any real-valued reward. Here the policy class we consider is that of *softmax* policies (Swaminathan and Joachims 2015a), Π_{SM}, where each $\pi_w(x) \in \Pi_{\text{SM}}$ is defined as

$$\pi_w(a|x) = \frac{f_w(\phi(x,a))}{\sum_{a' \in A} f_w(\phi(x,a'))}, \tag{4}$$

for a scoring function f (e.g., a neural network) with parameters w (e.g., network weights) and a feature representation $\Phi(x, a)$. The features could be any embedding we have for the context-action pairs, or it can be a simple concatenation of the context and action features. This policy is a "soft" version of the arg-max function, and the Euclidean length of w determines how random its selections are; larger w typically yields a more deterministic policy, whereas smaller w yields a more uniformly random policy. The CRM learning objective in this specific setting is:

$$\arg\max_{\pi_w \in \Pi_{\text{SM}}} \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_w(a_i|x_i)}{\pi_0(a_i|x_i)} r_i. \tag{5}$$

This optimization can be solved using standard optimization techniques (e.g., *stochastic gradient descent*). There are several improvements one could make to this learning objective, such as adding a *regularizer* to penalize the complexity of the policy, or adding a variance penalization term, both of which reduce *overfitting* (Swaminathan and Joachims 2015a). Various advanced CRM methods have also been studied recently (Chen et al. 2019b; Faury et al. 2020; Joachims, Swaminathan, and de Rijke 2018; Kallus 2020; London and Sandler 2019; Ma, Wang, and Narayanaswamy 2019; Wu and Wang 2018).

Note that the CRM approach to policy learning is fundamentally different from a "Model the World" approach to learning, in which the logged data are used to learn a reward predictor. The reward predictor can be learned using regression or matrix factorization; then, given the learned predictor, the resulting policy picks the action that has the highest predicted reward. While straightforward at first glance, there are a few drawbacks.

First, the performance of this learning algorithm depends strongly on the estimation accuracy of the reward predictor. Since we train the reward predictor using logged data, it is expected that the model will be estimated more accurately on the actions that the logging policy sampled more often, and less accurately on the actions it sampled infrequently. Second, unless the reward model is properly specified, approximation errors in learning the reward predictor will translate to modeling bias; hence, a potentially poor policy. Finally, and maybe most importantly, the

"Model the World" approach is rather indirect; instead of directly learning the optimal policy, we first solve a harder problem, which is to get an accurate reward predictor for all context-action pairs. A good reward estimator will usually translate to a good policy. However, to derive a good policy, we do not need to get an equally accurate estimate in magnitude for all actions; we only need to get the correct maximizer of the reward for each context to provide the optimal recommendation. The CRM approach avoids this indirection by directly optimizing an unbiased estimate of the desired online metric.

In summary, the CRM approach and the "Model the World" approach to learning are two standard approaches used in off-policy learning. On one hand, the CRM approach gives us an unbiased learning objective that is based directly on the online metric. However, its high variance may affect its generalization performance. On the other hand, the "Model the World" approach has much smaller variance by avoiding the use of importance sampling weights, and it is a reasonable alternative when the CRM approach gives us unstable estimates.

However, the bias is a severe issue for the "Model the World" approach and it exists even when we have an infinite amount of data. Recently, there are more advanced learning objectives that combine these two components more efficiently, such as *doubly-robust* (Dudík, Langford, and Li 2011) and *continuous adaptive blending* (Su et al. 2019), which aim to leverage the benefits from both approaches.

FAIRNESS IN RECOMMENDATION

So far we have considered the utility (i.e., expected reward) as our main objective, which ideally reflects how useful the system is to an average user. Optimizing utility has a long history in information retrieval (Robertson 1977), and it is still the key objective of most learning algorithms. Yet, it is now widely recognized that utility is not the only objective that a recommender system should optimize, especially when there are multiple stakeholders with potentially divergent interests. Many systems can be viewed as mediating a two-sided market in which the recommendation policy affects the users, the items, the platform and its dynamics, and the dynamics of the recommender system itself (Abdollahpouri et al. 2020; Evans and Schmalensee 2016; Singh and Joachims 2018; Wang and Joachims 2021). We argue that a causal view of recommendation—one that understands a recommendation policy in terms of the effects its actions have on all stakeholders—adds a rigorous theoretical basis for reasoning in this complex space.

Once we move away from the conventional objective of optimizing utility, one quickly arrives at trade-offs, and the question of how to make these trade-offs in a fair way. How does maximizing utility to the users relate to fairness considerations for the items? How should high average utility be traded off against more uniform utility for all subgroups of users? How should short-term objectives (e.g., clicks) be traded against long-term objectives (e.g., avoiding polarization)? Clearly, there is no universal technical answer for how to make these trade-offs (Holstein et al. 2019). However, we can provide technical answers for how to design systems that guarantee certain trade-offs, help reason about what can and what cannot be implemented, and help understand the dynamics of such systems. As designers of recommendation systems, this gives us a crucial role in collaboration with social scientists, legal scholars, and domain experts to determine appropriate fairness goals for different applications (Abebe et al. 2020).

To start, a prerequisite for fairness is our ability to measure the effect of actions in an unbiased way. This is probably where the counterfactual estimation approach introduced above is most directly helpful, since it remedies the effect of *selection biases* (i.e. biases introduced into the data through the actions selected by the logging policy). Consider the scenario of a movie recommender system, where we want to estimate the merit of each movie from past user feedback (e.g., completed streams) given the rankings our system presented. If we cannot estimate merit accurately, there is little hope that we can fairly allocate *exposure* (e.g. how often we recommend a movie, where to position a movie in a ranking, etc.) to the movies based on their merit. Unfortunately, the policy we used to present rankings has an influence on the feedback we receive, and it creates a *position bias*, which creates a form of selection bias (Joachims et al. 2007). Movies higher in the ranking will be discovered more easily, and are thus more likely to be streamed more often. This implies that movies (or job candidates, college applicants, etc.) that were historically disadvantaged in the ranking have little chance of rising to the top based solely on their merit, thus amplifying past inequities and leading to undesirable system dynamics (Joachims, Swaminathan, and Schnabel 2017; Mehrotra et al. 2018; Morik et al. 2020). Fortunately, estimators similar to Equation (1) can provide unbiased estimates, despite the selection biases in the feedback data (Agarwal et al. 2019; Joachims, Swaminathan, and Schnabel 2017; Wang et al. 2018), thereby disrupting the rich-get-richer dynamics.

But even if we manage to eliminate all exogenous sources of biases, of which selection bias is only one, there are still design choices endogenous to the recommender systems that affect the various stakeholders (Abdollahpouri et al. 2020; Singh and Joachims 2018; Wang and Joachims 2021). The counterfactual estimators and regularizers discussed in this article can again be useful in

TABLE 1 Ranking by probability of relevance can lead to disparate or undesirable allocation of exposure

Rank	Article	P(relevant)	Exposure
1	L_1	50.99	High
2	L_2	50.98	High
3	L_3	50.97	High
…	…	…	…
10	R_1	49.03	Low
11	R_2	49.02	Low
12	R_3	49.01	Low
…	…	…	…

addressing these endogenous sources of bias. Consider, for example, fairness to the items in a recommender system that serves a two-sided market (Biega, Gummadi, and Weikum 2018; Singh and Joachims 2018). It was shown that maximizing the utility to the users can lead to an undesirable winner-takes-all dynamic for the items (Singh and Joachims 2018). The following example illustrates this. It builds on a key result from information retrieval, named the *probability ranking principle* (Robertson 1977), which shows that ranking the items by their probability of relevance maximizes utility for many commonly used utility metrics. Let's consider a non-personalized "top news" panel on a news aggregation website, where we managed to get highly accurate and unbiased estimates of relevance probabilities—the probability that an incoming user will want to read an article. If we follow the *probability ranking principle* and rank the articles by their probability of relevance, we get the ranking in Table 1. This ranking reflects that around 51% of the visitors prefer the left-leaning articles from newspaper L, and around 49% prefer the right-learning articles from newspaper R. The problem is that even though the two news sources have almost equal probability of relevance, the ranking gives disproportionately more exposure to the articles from L than from R. Not only may newspaper R object to this ranking as unbalanced, it can also lead to undesirable polarization dynamics where right-leaning users abandon the platform. Such scenarios apply not only to news rankings, but to most other settings where the items or their providers gain utility from the recommendation policy. In short, while maximizing utility to the users may be the only objective in some settings, in others it does not necessarily lead to a desirable allocation of exposure to the items, and recent work on merit-based fairness-of-exposure augments utility maximization with additional fairness constraints (Beutel et al. 2019; Biega, Gummadi, and Weikum 2018; Celis, Straszak, and Vishnoi 2017; Morik et al. 2020; Yao and Huang 2017; Singh and Joachims 2018, 2019; Wang and Joachims 2021; Yadav, Du, and Joachims 2021; Zehlike et al. 2017) or social-welfare objectives (Mehrotra et al. 2018; Vondrať k 2008; Wang and Joachims 2021; Xiao et al. 2017; Yue and Guestrin 2011).

Finally, the example also illustrates the point that a system can provide inequitable utility to subgroups of the user population, even if we optimize a user-centric notion of utility in the form of an average reward. Because we are optimizing an average, this can unfairly marginalize minority user groups, decreasing how useful the recommender system is to them in order to better serve the majority (Wang and Joachims 2021; Xiao et al. 2017). All these additional fairness requirements make the problem of recommendation more complex, since we move from maximizing utility to a more complex objective that includes fair allocation of exposure to items and fair utility trade-offs for users. Counterfactual estimators are very promising for evaluating these objectives and the "what-if" questions implied by different design choices, since they provide a path to unbiased offline estimates of online impact.

While the view of recommendations as interventions and its corresponding causal inference framework provide novel opportunities for implementing fairness objectives, we also have to be mindful about their impact. As already mentioned above, a key ingredient of counterfactual evaluation and learning is the collection of logged data using a logging policy that is stochastic enough to explore the potential outcomes when we take different actions. This exploration comes at an immediate cost to the users, and we have to make sure that this cost is equitably distributed. However, even if the short-term cost is balanced between stakeholders, insufficient exploration for some groups can put them at a disadvantage in the future. Less exploration means larger variance when using IPS-style (Dudík, Langford, and Li 2011; Horvitz and Thompson 1952; Su et al. 2019) estimators for offline evaluation and learning, and thus may lead to policies that are sub-optimal for under-explored items or user groups. Designing logging policies that balance these short-term and long-term costs is an important question for future work.

While our understanding of fairness in recommender systems is growing, there is still no consensus on what exactly fairness means for a recommender system. Appropriate definitions of fairness might be different for different applications (Holstein et al. 2019). Furthermore, we should be careful when we incorporate fairness into a recommender system, since an incomplete understanding of the system dynamics may lead to undesirable long-term impact (Liu et al. 2018). This calls for collaboration with social scientists, legal scholars, and domain experts to determine appropriate fairness goals for particular recommendation applications.

PRACTICAL CONSIDERATIONS AND CHALLENGES

While there are many advantages of offline evaluation and learning, there are also some inherent challenges to adopting these methodologies in a realistic setting. Notably, the importance-weighting estimators introduced above are only unbiased if the logging policy is sufficiently randomized, and the propensities (i.e., probabilities) of the selected actions must be known. These requirements pose two fundamental problems, which we now discuss.

Randomized data collection is risky from a business perspective because it can harm the user experience. Randomization typically increases the probability that users receive irrelevant recommendations — which, if excessive, erodes user satisfaction, and could lead to attrition. Due to this risk, data collection must strike a balance between *exploring* the user's preferences (via randomization) and *exploiting* what is known about their preferences so far. Fortunately, this trade-off is precisely what multi-armed bandit algorithms are designed for; so if a stochastic bandit algorithm (such as Thompson sampling (Thompson 1933), Boltzmann exploration (Cesa-Bianchi et al. 2017) or EXP3 (Auer et al. 2002; Seldin and Slivkins 2014)) is already being used to optimize recommendations, one can log its user interactions for future offline analysis. Since bandits usually converge to a deterministic policy, one can either "freeze" updating after a certain amount of time, or modify the algorithm to always explore a little bit (e.g., $\in$-greedy).

Unfortunately, the requirement of knowing the propensities of logged actions disqualifies some bandit algorithms from data collection. For example, the posterior distributions of some Bayesian bandits (such as linear Thompson sampling (Agrawal and Goyal 2013)) do not allow for efficient computation of propensities. They can be approximated using Monte Carlo methods, but this can bias offline evaluation.

Even if the logging policy is sufficiently randomized and supports efficient, exact propensity calculation, the logging policy may be part of a larger system that filters the recommendations, thereby complicating data collection. Indeed, most recommender systems apply some form of guardrailing to prevent "catastrophic" failures, such as recommending content that is blatantly inappropriate for the users. While these precautions are necessary, they introduce a bias that may be difficult to quantify — and hence, difficult to compensate for.

The challenges posed by offline evaluation have inspired many estimators and algorithms, but even this abundance of solutions poses a challenge to practitioners, since they must choose the best method for their application. This may require intimate knowledge of the intricacies of the application, such as: how stochastic the logging policy is; how much data can be collected versus how much is needed for confident estimates; and whether the target policy is appropriate for the estimator. If feasible, one can empirically validate offline estimators via online experimentation; for instance, by comparing online metrics to their offline estimates.

Clearly, there are many challenges to deploying offline evaluation and learning in practice. Yet, there is already growing adoption in real-world systems, which demonstrates that, despite the challenges, the approach is feasible and robust (Agarwal et al. 2016; Chen et al. 2019a; Gruson et al. 2019).

OUTLOOK AND EMERGING TOPICS

In the previous sections, we showed that viewing recommendations as interventions provides a promising framework for evaluating and learning new interaction policies. In particular, the machine learning models driving these policies are correcting for the bias that the system's actions induce in the logged data. This principled approach based on causal inference allowed us to evaluate machine learning models using offline data *as if* they were deployed in A/B tests. Online A/B testing however is *not* the be-all and end-all of recommenders. It can be prohibitively expensive when measuring very long-term system effects, or identifying how users might co-adapt to a new system. Hence, we need to carefully interpret and extend A/B testing to build reliable online systems (Kohavi et al. 2012). The interventional framing of recommenders can prove to be useful in reasoning beyond what A/B tests can measure, and provide insights on how recommendation policies should be evaluated and trained.

When we introduced counterfactual estimators earlier in this article, we assumed that only a single item is recommended during each interaction. However, most practical recommendation interfaces display rankings or slates of recommended items. Directly applying the IPS estimator is impractical in these situations because the variance of the estimator typically scales with the number of possible rankings or slates. There are combinatorially many rankings or slates and IPS would require unreasonable amounts of randomization and logged data to return a reliable estimate. Several practical counterfactual estimators have been developed that exploit the combinatorial structures in these interfaces (McInerney et al. 2020; Swaminathan et al. 2017). Ongoing research uses ideas from doubly-robust estimation and non-parametric statistics (Bibaut et al. 2019; Su et al. 2020; Yin and Wang 2020) to further tune the bias-variance trade-off of these kinds of estimators and provide a plug-and-play practical solution for realistic recommendation interfaces (Dimakopoulou et al. 2019; Ma et al. 2020a).

The study of recommender systems has crucially relied on good online metrics and we now have techniques that can estimate them offline and even reliably optimize them. When we discussed learning, we saw that counterfactual estimators are directly composable with standard machine learning principles, and that they lead to new loss functions and regularizers for off-policy training. However, the exposition in this article assumed that the metrics are well-modeled as bandit feedback, which can be rather myopic. There are several session-based metrics that have been developed for recommender systems which aim to capture long-term user utility (Jannach, Mobasher, and Berkovsky 2020; Ludewig and Jannach 2018). Sequence-based modeling techniques (inspired by the language modeling literature) have also been studied to optimize them (Ma et al. 2020b). However, there is still a disconnect between the offline objectives optimized by these techniques and the online session-based metrics. Developing counterfactual estimators for longer-term metrics, and developing reliable training paradigms to optimize them remains an open problem. Borrowing techniques from reinforcement learning appears to be a promising direction to approach this problem (Chen et al. 2019a).

When we discussed fairness, we observed that it is useful to view recommenders as mediating mechanisms (rather than stationary interaction policies) and require that they be robust for many different kinds of objectives and manipulations. Fairness is only one dimension of the vast landscape of robustness research. Like fairness, robustness can take many forms across different applications, and the following types of robustness may be useful in several settings.

Item manipulation: Item creators know that recommender systems must perform some amount of exploration to determine the quality of new items (see (Choi and Sayedi 2019) for an example in advertising platforms). Rather than improving the quality of the items they create, the creators can add spurious "duplicate" items to inventory that the recommendation platforms need to explore afresh.

User manipulation: When recommenders transfer insights across user populations, a content producer can create fake users who strongly prefer their items and make them "appear like" a subpopulation to target.

Strategic behavior: Users interacting with personalized recommenders over a period of time can co-adapt in unpredictable ways. Consider a loan recommendation scenario: a user might strategically alter their behavior in many different ways to achieve the outcomes they want (e.g., intervene to improve their credit score); some of these interventions (e.g., repaying old debts) may be aligned with the system's goals, while others (e.g., spurious manipulations of credit history) are not.

Building robust recommender systems has been an ad hoc exercise so far. It is unclear how recommenders can be made provably robust to various manipulations; however, the first step will still require reasoning about the counterfactuals following different interventions. We anticipate future studies to build on the interventional view and establish a firm foundation for robust recommender systems.

A fundamental requirement for employing the techniques discussed in this article is the availability of large quantities of logged data. The bias-variance trade-off that IPS estimators make essentially requires a large data regime where the variance is low enough to detect a reliable signal. Due to privacy concerns and GDPR regulations, a recommender system may not store data for long enough (and thus, not aggregate enough data to reach such a regime). With the advent of *federated learning*—an edge-computing framework in which training data stays on-device—there is an opportunity to harness massive amounts of data to create highly personalized experiences, while protecting the user's data security.

Moreover, differential privacy provides mechanisms to ensure that federated learning does not reveal too much about any particular user. Thus, combining counterfactual techniques with federated learning and differential privacy (e.g., as in (Agarwal et al. 2018; Geyer, Klein, and Nabi 2017; McMahan et al. 2018)) may yield a new class of counterfactual learning techniques that produce recommendation algorithms complying with data storage regulations and offering provable privacy guarantees.

Taking stock of the journey of recommendation research so far, we have become very good at learning black-box models and recommending items to user populations. However we are beginning to apply these techniques to much more complex problems involving major societal functions. Mediating the job markets of the future has much higher stakes than recommending entertainment. Nonetheless, we are optimistic that the next generation of recommender and decision-support systems will bring transparency into many processes that are currently human-driven. In the words of Isaac Asimov, "I could not bring myself to believe that if knowledge presented danger, the solution was ignorance. To me, it always seemed that the solution had to be wisdom." The interventional view we have espoused in this article begins by recognizing that recommendations have consequences in the world, and is a step towards realizing that next generation of wise recommender systems that actively consider the impact of their actions.

ACKNOWLEDGEMENTS

This research was supported in part by NSF Awards IIS-1901168 and IIS-2008139, as well as a Bloomberg Fellowship. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

Abdollahpouri, H., G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, and L. Pizzato. 2020. "Multistakeholder Recommendation: Survey and Research Directions." *User Modeling and User-Adapted Interaction* 30(1): 127–58.

Abebe, R., S. Barocas, J. Kleinberg, K. Levy, M. Raghavan, and D. G. Robinson. 2020. "Roles for Computing in Social Change." In *Conference on Fairness, Accountability, and Transparency*, 252–60.

Agarwal, A., S. Bird, M. Cozowicz, L. Hoang, J. Langford, S. Lee, J. Li, et al. 2016. Making contextual decisions with low technical debt. *ArXiv Preprint arXiv:1606.03966*.

Agarwal, A., S. Basu, T. Schnabel, and T. Joachims. 2017. "Effective Evaluation Using Logged Bandit Feedback from Multiple Loggers." In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

Agarwal, N., A. Suresh, F. Yu, S. Kumar, and B. McMahan. 2018. "cpSGD: Communication-efficient and Differentially-private Distributed SGD." In *Advances in Neural Information Processing Systems*.

Agarwal, A., I. Zaitsev, X. Wang, C. Li, M. Najork, and T. Joachims. 2019. "Estimating Position Bias Without Intrusive Interventions." In *International Conference on Web Search and Data Mining*.

Agrawal, S., and N. Goyal. 2013. "Thompson Sampling for Contextual Bandits with Linear Payoffs." In *International Conference on Machine Learning*.

Angwin, J., J. Larson, S. Mattu, and L. Kirchner 2016. Machine bias. *ProPublica, May* 23:2016.

Auer, P., N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. 2002. "The Non-Stochastic Multiarmed Bandit Problem." *SIAM Journal of Computing* 32(1): 48–77.

Beutel, A., J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, et al. 2019. "Fairness in Recommendation Ranking Through Pairwise Comparisons." In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2212–20.

Bibaut, A., I. Malenica, N. Vlassis, and M. Van Der Laan. 2019. "More Efficient Off-Policy Evaluation Through Regularized Targeted Learning." In *International Conference on Machine Learning*, 654–3.

Biega, A. J., K. P. Gummadi, and G. Weikum. 2018. "Equity of Attention: Amortizing Individual Fairness in Rankings." In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 405–14.

Bottou, L., J. Peters, J. Q. Nonero Candela, D. Charles, D. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson. 2013a. "Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising." *Journal of Machine Learning Research* 14: 3207–60.

Bottou, L., J. Peters, J. Quiñonero-Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson. 2013b. "Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising." *Journal of Machine Learning Research* 14(1): 3207–60.

Celis, L. E., D. Straszak, and N. K. Vishnoi. 2017. Ranking with fairness constraints. *ArXiv Preprint arXiv:1704.06840*.

Cesa-Bianchi, N., C. Gentile, G. Neu, and G. Lugosi. 2017. "Boltzmann Exploration Done Right." In *Advances in Neural Information Processing Systems*.

Chen, M., A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi. 2019a. "Top-k Off-Policy Correction for a Reinforce Recommender System." In *International Conference on Web Search and Data Mining*, 456–64.

Chen, M., R. Gummadi, C. Harris, and D. Schuurmans. 2019b. "Surrogate Objectives for Batch Policy Optimization in One-Step Decision Making." In *Advances in Neural Information Processing Systems*, 8827–37.

Choi, W. J., and A. Sayedi. 2019. "Learning in Online Advertising." *Marketing Science* 38(4): 584–608.

Dimakopoulou, M., N. Vlassis, and T. Jebara. 2019. "Marginal Posterior Sampling for Slate Bandits." In *International Joint Conference on Artificial Intelligence*, 2223–9.

Dudík, M., J. Langford & L. Li 2011. "Doubly Robust Policy Evaluation and Learning." In International Conference on Machine Learning.

Evans, D. S., and R. Schmalensee. 2016. *Matchmakers: The New Economics of Multi-Sided Platforms*. Harvard Business Review Press.

Faury, L., U. Tanielian, E. Dohmatob, E. Smirnova, and F. Vasile. 2020. Distributionally robust counterfactual risk minimization. In *AAAI Conference on Artificial Intelligence*, volume 34: 3850–7.

Fox. 2004. House M.D. (Season 1, Episode 18). According to https://en.wikiquote.org/wiki/House (Season 1).

Geyer, R., T. Klein, and M. Nabi. 2017. "Differentially Private Federated Learning: A Client Level Perspective." *CoRR* abs/1712.07557.

Gilotte, A., C. Calauzénes, T. Nedelec, A. Abraham, and S. Dollé. 2018. "Offline a/b testing for recommender systems." In *International Conference on Web Search and Data Mining*, 198–206.

Gruson, A., P. Chandar, C. Charbuillet, J. McInerney, S. Hansen, D. Tardieu, and B. Carterette. 2019. "Offline Evaluation to Make Decisions about Playlist Recommendation Algorithms." In *ACM International Conference on Web Search and Data Mining*, 420–8.

Holstein, K., J. Wortman Vaughan, H. Daumé III, M. Dudík & H. Wallach 2019. "Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?." In ACM CHI Conference on Human Factors in Computing Systems, 1–16.

Horvitz, D. G., and D. J. Thompson. 1952. "A Generalization of Sampling without Replacement from a Finite Universe." *Journal of the American statistical Association* 47(260): 663–85.

Ionides, E. 2008. "Truncated Importance Sampling." *Journal of Computational and Graphical Statistics* 17(2): 295–311.

Jannach, D., B. Mobasher, and S. Berkovsky. 2020. "Research Directions in Session-based and Sequential Recommendation." *User Modeling and User-Adapted Interaction* 30(4): 609–16.

Joachims, T., L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. 2007. "Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations In Web Search." *ACM Transactions on Information Systems* 25(2).

Joachims, T., A. Swaminathan, and M. de Rijke. 2018. "Deep Learning with Logged Bandit Feedback." In *International Conference on Learning Representations*.

Joachims, T., A. Swaminathan, and T. Schnabel. 2017. "Unbiased Learning-to-Rank with Biased Feedback." In *ACM Conference on Web Search and Data Mining*, 781–9.

Kallus, N. 2020. "More Efficient Policy Learning Via Optimal Retargeting." *Journal of the American Statistical Association* 116(534): 1–34.

Kohavi, R., A. Deng, B. Frasca, R. Longbotham, T. Walker, and Y. Xu. 2012. "Trustworthy Online Controlled Experiments: Five Puzzling Outcomes Explained." In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 786–94.

Li, L., W. Chu, J. Langford, and X. Wang. 2011. "Unbiased Offline Evaluation of Contextual-Bandit-Based News Article Recommendation Algorithms." In *International Conference on Web Search and Data Mining*, 297–306.

Liu, L. T., S. Dean, E. Rolf, M. Simchowitz, and M. Hardt. 2018. "Delayed Impact of Fair Machine Learning." In *International Conference on Machine Learning*.

Liu, A., H. Liu, A. Anandkumar, and Y. Yue. 2019. "Triply Robust Off-Policy Evaluation." *CoRR* abs/1911.05811.

London, B., and T. Sandler. 2019. "Bayesian Counterfactual Risk Minimization." In *International Conference on Machine Learning*.

Ludewig, M., and D. Jannach. 2018. "Evaluation of Session-Based Recommendation Algorithms." *User Modeling and User-Adapted Interaction* 28(4-5): 331–90.

Ma, J., Z. Zhao, X. Yi, J. Yang, M. Chen, J. Tang, L. Hong, and E. H. Chi. 2020a. "Off-policy Learning in Two-Stage Recommender Systems." In *The Web Conference*, 463–73.

Ma, Y., B. Narayanaswamy, H. Lin, and H. Ding. 2020b. "Temporal-contextual Recommendation in Real-Time." In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2291–9.

Ma, Y., Y. - X. Wang, and B. Narayanaswamy. 2019. "Imitation-regularized Offline Learning." In *International Conference on Artificial Intelligence and Statistics*, 2956–65.

McInerney, J., B. Brost, P. Chandar, R. Mehrotra, and B. Carterette. 2020. "Counterfactual Evaluation of Slate Recommendations With Sequential Reward Interactions." In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1779–88.

McMahan, H., D. Ramage, K. Talwar, and L. Zhang. 2018. "Learning Differentially Private Recurrent Language Models." In *International Conference on Learning Representations*.

Mehrotra, R., J. McInerney, H. Bouchard, M. Lalmas, and F. Diaz. 2018. "Towards a Fair Marketplace: Counterfactual Evaluation of The Trade-Off Between Relevance, Fairness & Satisfaction in Recommendation Systems." In *ACM International Conference on Information and Knowledge Management*, 2243–51.

Morik, M., A. Singh, J. Hong, and T. Joachims. 2020. "Controlling Fairness and Bias in Dynamic Learning-To-Rank." In *ACM SIGIR Conference on Research and Development in Information Retrieval*.

Azure personalizer service. Available at https://docs.microsoft.com/en-in/azure/cognitive-services/personalizer/.

Nezik, A. - K. 2019. "Wenn maschinen kalt entscheiden." *Die Zeit* 44. https://www.zeit.de/2019/44/algorithmen-software-diskriminierung-arbeitsmarkt-assistenz-system.

Robertson, S. E. 1977. "The Probability Ranking Principle In IR." *Journal of Documentation*.

Rosenbaum, P., and D. Rubin. 1983. "The Central Role of the Propensity Score in Observational Studies for Causal Effects." *Biometrika* 70: 41–55.

Seldin, Y., and A. Slivkins. 2014. "One Practical Algorithm for Both Stochastic and Adversarial Bandits." In *International Conference on Machine Learning*.

Singh, A., and T. Joachims. 2018. "Fairness of Exposure in Rankings." In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Singh, A., and T. Joachims. 2019. "Policy Learning for Fairness in Rankings." In *Advances in Neural Information Processing Systems*.

Strehl, A., J. Langford, L. Li, and S. Kakade. 2010a. "Learning from Logged Implicit Exploration Data." In *Advances in Neural Information Processing Systems*.

Strehl, A., J. Langford, L. Li, and S. M. Kakade. 2010b. "Learning from logged implicit exploration data." In *Advances in Neural Information Processing Systems*, 2217–25.

Su, Y., L. Wang, M. Santacatterina, and T. Joachims. 2019. "CAB: Continuous Adaptive Blending for Policy Evaluation And Learning." In *International Conference on Machine Learning*.

Su, Y., M. Dimakopoulou, A. Krishnamurthy & M. Dudík 2020. "Doubly robust off-policy evaluation with shrinkage." In International Conference on Machine Learning, 9167–76.

Swaminathan, A., and T. Joachims. 2015a. "Counterfactual Risk Minimization: Learning From Logged Bandit Feedback." In *International Conference on Machine Learning*, 814–23.

Swaminathan, A., and T. Joachims. 2015b. "The Self-Normalized Estimator for Counterfactual Learning." In *Advances in Neural Information Processing Systems*.

Swaminathan, A., A. Krishnamurthy, A. Agarwal, M. Dudík, J. Langford, D. Jose, and I. Zitouni. 2017. "Off-policy Evaluation for Slate Recommendation." In *Advances in Neural Information Processing Systems*, 3635–45.

Thomas, P., G. Theocharous, and M. Ghavamzadeh. 2015. "High-confidence Off-policy Evaluation." In *AAAI Conference on Artificial Intelligence*.

Thompson, W. 1933. "On the Likelihood that One Unknown Probability Exceeds Another in View of The Evidence of Two Samples." *Biometrika* 25: 285–94.

Vlassis, N., A. Bibaut, M. Dimakopoulou, and T. Jebara. 2019. "On the Design of Estimators for Bandit Off-Policy Evaluation." In *International Conference on Machine Learning*.

Vondrák, J. 2008. "Optimal Approximation for the Submodular Welfare Problem in the Value Oracle Model." In *ACM Symposium on Theory of Computing*, 67–74.

Wang, L. & T. Joachims 2021. "User Fairness, Item Fairness, and Diversity for Rankings in Two-Sided Markets." ACM International Conference on the Theory of Information Retrieval.

Wang, X., N. Golbandi, M. Bendersky, D. Metzler, and M. Najork. 2018. "Position bias Estimation for Unbiased Learning to Rank in Personal Search." In *ACM International Conference on Web Search and Data Mining*, 610–8.

Waters, A., and R. Miikkulainen. 2014. "Grade: Machine Learning Support for Graduate Admissions." *AI Magazine* 35(1): 64–75.

Wu, H., and M. Wang. 2018. "Variance Regularized Counterfactual Risk Minimization Via Variational Divergence Minimization." In *International Conference on Machine Learning*, 5353–62.

Xiao, L., Z. Min, Z. Yongfeng, G. Zhaoquan, L. Yiqun, and M. Shaoping. 2017. "Fairness-aware Group Recommendation with Pareto-Efficiency." In *ACM Conference on Recommender Systems*, 107–5.

Yadav, H., Z. Du, and T. Joachims. 2021. "Fair Learning-to-Rank From Implicit Feedback." In *ACM SIGIR Conference on Research and Development in Information Retrieval*.

Yao, S., and B. Huang. 2017. "Beyond Parity: Fairness Objectives for Collaborative Filtering." In *Advances in Neural Information Processing Systems*, 2921–30.

Yin, M., and Y. - X. Wang. 2020. "Asymptotically Efficient Off-Policy Evaluation for Tabular Reinforcement Learning." In *International Conference on Artificial Intelligence and Statistics*, 3948–58.

Yue, Y., and C. Guestrin. 2011. "Linear Submodular Bandits and their Application to Diversified Retrieval." In *Advances in Neural Information Processing Systems*, 2483–91.

Zehlike, M., F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates. 2017. "Fa* ir: A fair top-k ranking algorithm." In *ACM Conference on Information and Knowledge Management*, 1569–78.

AUTHOR BIOGRAPHIES

Thorsten Joachims is a Professor in the Department of Computer Science and in the Department of Information Science at Cornell University, and he is an Amazon Scholar.

Ben London is a Senior Scientist at Amazon.

Yi Su is a PhD student in the Department of Statistics and Data Science at Cornell University.

Adith Swaminathan is a Senior Researcher at Microsoft Research.

Lequn Wang is a PhD student in the Department of Computer Science at Cornell University.

How to cite this article: Joachims, T., B. London, Y. Su, A. Swaminathan, and L. Wang. 2021. "Recommendations as treatments." *AI Magazine*. 42: 19–30. https://doi.org/10.1609/aaai.12014