

Biases in Data-Driven Networking, and What to Do About Them

Mihovil Bartulovic
Carnegie Mellon University

Junchen Jiang
Microsoft Research/Carnegie
Mellon University

Sivaraman Balakrishnan
Carnegie Mellon University

Vyas Sekar
Carnegie Mellon University

Bruno Sinopoli
Carnegie Mellon University

ABSTRACT

Recent efforts highlight the promise of data-driven approaches to optimize network decisions. Many such efforts use trace-driven evaluation; i.e., running offline analysis on network traces to estimate the potential benefits of different policies before running them in practice. Unfortunately, such frameworks can have fundamental pitfalls (e.g., skews due to previous policies that were used in the data collection phase and insufficient data for specific subpopulations) that could lead to misleading estimates and ultimately suboptimal decisions. In this paper, we shed light on such pitfalls and identify a promising roadmap to address these pitfalls by leveraging parallels in causal inference, namely the Doubly Robust estimator.

1 INTRODUCTION

Driven by the opportunity to collect and analyze data (e.g., application quality measurement from end users), many recent proposals have demonstrated the promise of using data-driven optimization of networked systems (e.g., [1, 3, 4, 15, 16, 29, 30]). At a high-level, these proposals build data-driven prediction models to capture the relationships between observable features (e.g., client IP, location, device type) and performance indices, and use these predictive models to guide the decision making for future sessions. For instance, recent work [1, 3, 4, 15, 16, 29, 30] demonstrates the promise in several applications such as optimizing quality in video and VoIP as well as classical problems in routing and traffic engineering.

For such data-driven decision making to be effective, we need some way to compare and contrast different *policies*, where a policy specifies the decision making outcomes (e.g., a mapping of clients to CDNs based on their attributes). A

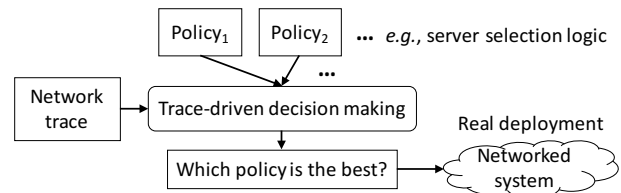


Figure 1: Trace-driven evaluation predicts the best policies using traces of empirical measurement data.

common approach is *trace-driven evaluation* as depicted in Figure 1. Here, we evaluate different policies (e.g., a server selection policy) in an offline fashion by using network traces, before deploying them. Since operators may be reluctant to carry out real-traffic experiments to avoid disruption and SLA violations [24], this workflow offers a practical alternative. Moreover, this approach can capture real-world complex interactions that can occur in networked systems which might be intractable to simulate analytically [10].

Unfortunately, such trace-driven evaluation if performed without care could lead to inaccurate and suboptimal decision making. One source of error stems from potential skews in the trace collection process and consequently how the prediction models process the trace to capture the relationships between decisions and performance. For instance, if we use a trace of packet loss rate between only WiFi clients and a server to estimate the packet loss rate between wired clients and the server, we would have an overestimation of the packet loss rate. Furthermore, there could be inherent sources of variance and noise that may affect the statistical validity of the evaluation process, especially if our measurement coverage for certain subpopulations is sparse (e.g., set of clients in city X using server Y in CDN Z).

Seen in a broader context, such trace-driven evaluation frameworks and their potential pitfalls are not unique to the networking domain. Indeed, such sources of error are well documented in the machine learning literature and in other domains such as ad recommendation [28] and medical treatment [5, 6]. Drawing on this parallel, we see an opportunity to build a principled networking trace-driven evaluation framework based on advances in these domains.

In particular, we identify a promising connection with work on *Doubly Robust (DR)* estimation techniques [5, 9, 21, 32,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

HotNets-XVI, November 30–December 1, 2017, Palo Alto, CA, USA

© 2017 Association for Computing Machinery.

ACM ISBN 978-1-4503-5569-8/17/11...\$15.00

<https://doi.org/10.1145/3152434.3152448>

34]. To understand DR it is useful to visit two natural approaches for building such an evaluation framework: (1) *direct method* (DM) estimator and (2) *inverse propensity score* (IPS) estimator. Intuitively, the DM estimator uses a reward model (i.e., a model for the performance/utility of different decisions) based on the collected trace while the IPS estimator avoids the need for this model and estimates the observed performance on the collected trace. Both methods have inherent tradeoffs; the IPS estimator works well when decisions taken by the old policy have high probability under the new policy but can have high variance when this is not the case. On the other hand, the DM avoids this coverage problem by using all the available trace data, but relies crucially on the ability to generate an accurate reward model. Indeed, as we will see, many of the current approaches we observe in the networking literature to try and address these aforementioned pitfalls can be viewed as manifestations of one of these techniques.

Perhaps surprisingly, the literature on DR estimation shows that it is possible to combine these two methods to construct estimators that perform well with high-probability, when at least one of DM/IPS would have produced accurate estimates. Moreover, using DR allows us to make fewer assumptions about the nature of the workload, the trace collection process and the policy used when the trace was collected.

While DR is a promising starting point, there are several key challenges in applying it in networking contexts. First, simple applications of DR assume that both the old and new policy under evaluation are agnostic to the history of previous outcomes. Second, the theory implicitly assumes that the new policy is being evaluated under the same “system states” as when the trace was collected. However, this is not the case in networking where load and background traffic can affect outcomes. Third, in many prior uses for DR, there is an implicit notion of independence between decision making processes and the rewards. In a network setting, there are confounding factors due to load and congestion “self-induced” by previous decisions. For instance, if we assign many clients to a server then the performance for future clients using that server may be degraded.

Roadmap: We highlight use cases and pitfalls of trace-driven evaluation in §2. We provide background on DR estimation in §3. We highlight challenges in extending DR to networking and sketch preliminary ideas to address these in §4.

2 MOTIVATION

We begin by formalizing the notion of trace-driven evaluation and describe many use cases in networking where offline trace-driven evaluation is widely used. Then, we highlight the pitfalls of this process with illustrative examples.

2.1 Preliminaries

Definitions: We begin by formally defining the goal of a trace-driven evaluation framework and introduce notation that we use throughout this paper.

At a high-level, we consider a network application that seeks to use data-driven approaches to optimize some specific performance metric. To elaborate, we have a set of many *client-contexts*¹ $c \in \mathbb{C}$. We use the terms client or client-context to refer to a featurized summary of relevant client and contextual information. We also have a set of possible *decisions* $d \in \mathbb{D}$, and our goal is to optimize the (average) *reward* r (i.e., some performance metric).

We assume that this decision making system has access to a *trace*—a set of tuples $T = \{(c_k, d_k, r_k)\}_{k=1,2,\dots,n}$, of clients, specific per-client decisions, and performance outcomes (rewards). We define a *policy* as a function that maps clients to possible decisions; i.e., a policy returns $\mu(d|c)$, the probability of choosing the decision d for client c , and $\sum_{d \in \mathbb{D}} \mu(d|c) = 1$.

Now, a trace-driven evaluator takes as input a new policy μ_{new} (e.g., a new strategy we want to try), a *trace* consisting of n tuples $T = \{(c_k, d_k, r_k)\}_{k=1,2,\dots,n}$, and an old policy μ_{old} . That is, d_k is the decision made by the old policy μ_{old} on c_k , and r_k is the observed *reward* of d_k on c_k . We assume that the policy μ_{old} is known, i.e. we assume knowledge of the probability with which the old policy chose the decision d_k . In practice, it may be necessary to estimate this probability from the trace. The evaluator then returns as output an estimate $\hat{V}(\mu_{\text{new}}, \mu_{\text{old}}, T)$ of the expected reward: $V(\mu_{\text{new}}, T) = \frac{1}{n} \sum_{k=1}^n \sum_{d \in \mathbb{D}} \mu_{\text{new}}(d|c_k) r(c_k, d)$, if μ_{new} was used to make decisions for the same clients in the same sequence as in the trace. Using such a trace-driven evaluator, we can then compare different policies μ_{new} to pick the best possible strategy for future clients.

Use cases: Many networked systems in practice use this trace-driven evaluation workflow as part of their decision making systems. For example:

- Video content providers often use previously observed video quality to estimate the effect of using alternative CDN and bitrate selection algorithms [15, 18] before actually deploying these new policies.
- Other content providers use measurement traces to evaluate web server selection [29] and VoIP relay server selection [14] policies in an offline fashion.
- CDNs can test new configurations, such as ISP peering or placement of edge servers, in an offline fashion, before investing resources to deploy new configurations [38].
- To compare multiple adaptive bitrate (ABR) algorithms under the same network conditions, video content providers often use traces of throughput observed by real clients to predict the quality if a new ABR algorithm were to run on the same clients [31, 37, 42].
- Prior work on TCP congestion control (e.g., [11, 43]) uses traces of packet-level events (e.g., round-trip time, packet loss) to benchmark TCP congestion control performance under same network conditions as well as to predict the impact to end-to-end performance [7].

¹Through the rest of the paper we use client and client-context interchangeably.

- Other networking problems, such as traffic engineering (e.g., [20, 36, 41]), routing (e.g., [35, 39]), and cloud configuration (e.g., [40]), also benefit from accurate trace-driven evaluation.

2.2 Pitfalls of trace-driven evaluation

Despite its promise, trace-driven evaluation can have significant sources of errors. Next, we use illustrative examples to highlight two important sources of error which roughly stem from incorrect modeling assumptions or from the curse of dimensionality.

2.2.1 Model misspecification. One approach to trace-driven evaluation is based on estimating a *reward model*, i.e. a model for the rewards of the different decisions as a function of the client-contexts, and then using this predictive model to estimate the average reward. Such modeling-based approaches can lead to inaccurate conclusions in cases when the modeling assumptions made are incorrect, or when we are unable to estimate a reliable model due to data scarcity. We provide concrete examples, in the context of networking problems, of the issues that can affect modeling-based approaches.

Let us consider the problem of adaptive bitrate (ABR) policies in video streaming. When evaluating the QoE of different decisions (i.e., what bitrate to choose for the next chunk), many prior works rely on throughput estimation based on the observed throughput of recent chunks (e.g., MPC [42], FESTIVE [17]). In this process, the throughput estimator may implicitly assume that the observed throughput is independent of the chunk’s bitrate; i.e., if a client observes throughput of 2Mbps when downloading a 720p encoded video chunk, it assumes the throughput for the same client to download a 360p chunk would also be 2Mbps. However, using lower bitrates can lead to lower observed throughput than available bandwidth [12]; e.g., if the chunk size is too small for TCP to reach steady state. This can lead to misleading performance estimates. For instance, in Figure 2, the old ABR policy (i.e., used for previous chunks) chooses a low bitrate on the second chunk and observes low throughput. This leads the data-driven evaluator to falsely assume the throughput would be low even if the new ABR policy uses a higher bitrate on the second chunk, and this could potentially result in inaccurate QoE estimates.

Figure 3 illustrates another form of model misspecification. In this example, there are important unmeasured or unused features whose exclusion leads to inaccurate policy evaluation. To estimate the performance of relying a VoIP call via certain relay path ($A_2 \rightarrow R \rightarrow B_2$), VIA [14] identifies the calls between the same source and destination AS (e.g., $A_1 \rightarrow R \rightarrow B_1$) which were selected to use the same relay path by the old policy, and uses their observed performance as the performance estimation. (To map the notions in the example to those in the problem formulation, each VoIP call is a “client”, and the path choice is a “decision”.) However, if the old policy chooses only calls between two devices behind

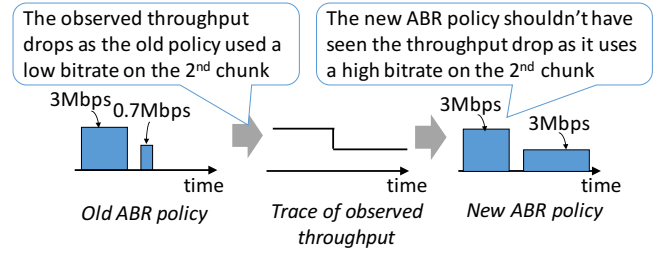


Figure 2: Adaptive Bitrate: Example where inaccurate assumptions in the reward model (observed throughput is independent of chunk bitrate) can lead to estimation errors.

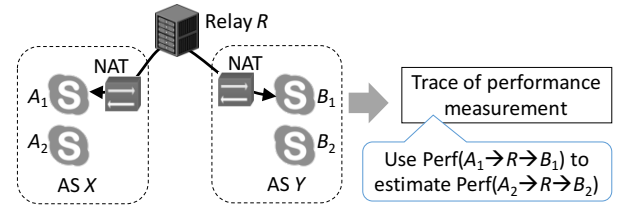


Figure 3: Relay selection: A selection bias in the old policy of only using relays for NAT-ed hosts could cause errors in the evaluation result.

NATs to use the relay path, the observed performance on these calls may not be indicative to infer the performance of relaying other calls between public IPs, since private IP users may have different last-mile network conditions than public IP users [22].

When a model is used to estimate the rewards for a new policy, similar issues can arise when the estimated model is inaccurate. For instance, given network packet traces, WISE [38] builds a Causal Bayesian Network (CBN) to capture the effect of different CDN configurations on average response time of requests. As a simple toy example in Figure 4, suppose each request from ISP-1 and ISP-2 can choose one of two frontend clusters (FE-1, FE-2) and one of two backend clusters (BE-1, BE-2). Note that in this example, a “client” is a request, and a “policy” is a CDN configuration that maps a request to a “decision” of frontend and backend. Our goal is to estimate the response time of a request X from ISP-1 using FE-1 and BE-2. The ground truth in the example is that the response time of a request from ISP-1 is high only when it uses BE-1 and FE-1; i.e., response time of X should be short. Suppose the trace input was small and WISE infers an incomplete CBN as shown; thus, WISE would incorrectly predict that request X has a long response time.

To summarize, there are two potential issues with purely modeling-based approaches: the predicted reward may be a poor estimate of the real rewards either because of inappropriate modeling choices, or because we have insufficient data to estimate a reliable model.

2.2.2 Model-free approaches and the curse of dimensionality. One approach to avoid the issues that stem

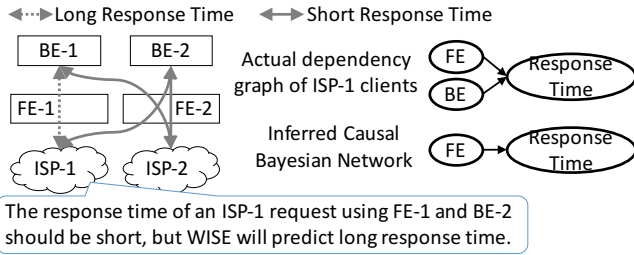


Figure 4: WISE: The causal model learned could be wrong depending on the trace

from inaccurate models is to only use the observed rewards, but to appropriately re-weight these rewards to account for the fact that the corresponding decisions are either more or less likely under the new policy (when compared to the old policy). Such approaches are only reliable when the overlap between old and new policy is high, i.e. the decision chosen by the old policy has a high probability of being chosen under the new policy. The problems of non-overlap between the old and new policy are exacerbated when the client-context vectors are moderately high-dimensional or when the decision space is sufficiently rich. As a concrete example, given the video quality of previously seen clients who have been randomly assigned to a set of available CDNs and bitrates, CFA [15] evaluates the video quality of a different client-CDN/bitrates assignment by using only the data of clients who use the same CDNs/bitrate in the old and new assignments. As shown in Figure 5, the estimate can be based only on a small amount of matches (or even no matches at all as in the figure). This can cause high variance in the evaluation results.

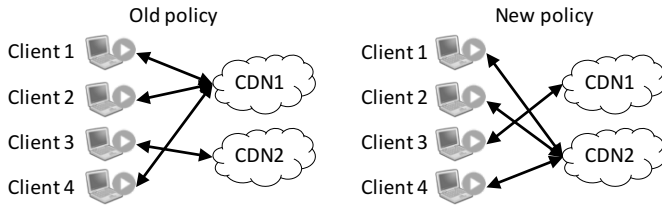


Figure 5: Matching the decisions of the old policy and the new policy is unbiased but could lead to low coverage and statistical significance.

3 DOUBLY ROBUST ESTIMATION

The main challenge of trace-driven evaluation is that the data collected with an old policy does not faithfully represent the proportions of actions of the new policy. Related issues arise in applications ranging from survey sampling [8] and causal inference [5, 21, 32, 34] to ad recommendation [28] and reinforcement learning [19]. Motivated by this past work, we see an opportunity to build a principled platform for networking trace-driven evaluation.

Specifically, we identify a natural parallel to notion of Doubly Robust (DR) estimation used in prior work. The key idea behind the DR estimator (depicted in Figure 6) is to combine two basic estimators, *Direct Method (DM)* and *Inverse*

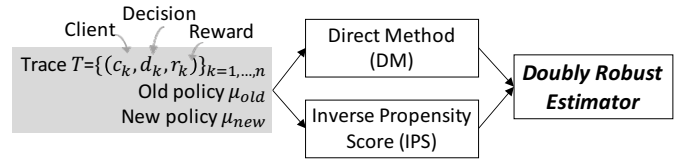


Figure 6: The DR estimator combines the estimates of two basic methods, DM and IPS.

Propensity Score (IPS), so that DR is more accurate than both DM and IPS in most cases. Concretely, we assume that the inputs to the estimators are a new policy μ_{new} , an old policy μ_{old} , and a trace $T = \{(c_k, d_k, r_k)\}_{k=1,2,\dots,n}$.

Two basic evaluation methods: At a high level, we can consider two abstract methods for policy evaluation, which the DR estimator is built up on:

- DM uses a *reward model* $\hat{r}(c, d)$ to predict the reward of any client c and decision d , and returns the average reward of a new policy μ_{new} by $\hat{V}_{\text{DM}} = \frac{1}{n} \sum_{k=1}^n \sum_{d \in \mathbb{D}} \mu_{\text{new}}(d|c_k) \hat{r}(c_k, d)$. An example of DM estimator can be found in Figure 2, in which a flow-level simulator (the reward model) is used to predict $\hat{r}(c, d)$ for any given chunk c and bitrate d .
- The main issue with directly using an old trace to evaluate a new policy is that when the old and new policy differ significantly, the proportions of actions chosen for each client in the trace do not match the new policy. IPS uses importance weighting to correct for these incorrect proportions. Concretely, the estimator is a *weighted* sum of rewards r_k actually observed by each client c_k on the decision d_k : $\hat{V}_{\text{IPS}} = \frac{1}{n} \sum_{k=1}^n \frac{\mu_{\text{new}}(d_k|c_k)}{\mu_{\text{old}}(d_k|c_k)} r_k$.

In typical scenarios, the IPS estimator is less prone to problems of bias since no model is assumed for the rewards, rather only observed rewards are used in constructing the estimate for the expected reward. The IPS estimator can however have large variance since we are inflating the influence of tuples for which $\mu_{\text{old}}(d_k|c_k)$ is small. In practice, the DM is based on an idealized reward model whose parameters are estimated using historic data and model mis-specification can lead to high bias.

DR estimator: The DR estimator combines the DM estimate \hat{V}_{DM} and the IPS estimate \hat{V}_{IPS} in the following manner:

$$\hat{V}_{\text{DR}} = \hat{V}_{\text{DM}} + \hat{V}_{\text{IPS}} - \frac{1}{n} \sum_{k=1}^n \frac{\mu_{\text{new}}(d_k|c_k)}{\mu_{\text{old}}(d_k|c_k)} \hat{r}(c_k, d_k) \quad (1)$$

In order to build some intuition for the form of the DR estimator, we consider two special cases when the DM and IPS estimators respectively are accurate and show that in these cases the DR estimator reduces to the accurate estimator. First, we observe that the DR estimator can be rewritten as an average over clients, i.e.,

$$\hat{V}_{\text{DR}} = \frac{1}{n} \sum_{k=1}^n \left[\sum_{d \in \mathbb{D}} \mu_{\text{new}}(d|c_k) \hat{r}(c_k, d) + \frac{\mu_{\text{new}}(d_k|c_k)}{\mu_{\text{old}}(d_k|c_k)} [r_k - \hat{r}(c_k, d_k)] \right] \quad (2)$$

We focus attention on the k -th tuple.

- If the new and old policy deterministically take the same action d_k the IPS estimator is accurate. In this case, we

observe that $\mu_{\text{new}}(d_k|c_k) = \mu_{\text{old}}(d_k|c_k)$ and the DR estimator for this client/tuple is equal to the IPS estimator.

- If the reward estimate from the DM is equal to the true reward for the k -th client, i.e. the reward model is accurate, then we observe that $r_k = \hat{r}(c_k, d_k)$. The DR estimator for this client/tuple is equal to the DM estimator.

To summarize, at least in these special cases, the DR estimate of the reward agrees with the IPS estimate for tuples for which we expect the IPS estimate to be accurate, and agrees with the DM estimate for tuples for which we expect the DM estimate to be accurate.

Under certain assumptions the DR estimator is well-understood [5, 9, 21, 32, 34] to possess “second-order bias”, i.e. roughly its error is upper bounded by the product of the error of the DM and IPS estimators. This property leads to its so-called double robustness, the estimator is accurate if either the DM or IPS estimators, but not necessarily both, is accurate. This property holds even when we do not know which of the DM/IPS estimators are accurate. Moreover, the second-order bias ensures that in cases when both the DM/IPS estimators are accurate, the DR estimator improves on each of them individually.

Why DR for networking: To see why these aforementioned properties of DR are appealing in a networking context, let us revisit the pitfalls from §2. Indeed, some of the existing approaches to trace-driven evaluation (e.g., CBN in WISE, CFA) are manifestations of a DM- or IPS-like strategy. For example, the CBN in WISE is effectively trying to build a reward model (i.e., it is a form of DM estimation) and the CFA technique of finding “overlaps” is a primitive form of IPS. Using DR, we can potentially address the shortcomings of an incorrect model (e.g., CBN in WISE) or the sparsity issue (e.g., in CFA). Finally, in the NAT relaying example, ideally we need to add in the relevant feature (e.g., NAT-ed host). However, this increases the dimensionality of the feature space, and consequently degrades estimation accuracy when we are building a reward model for DM or using inverse-probability weighting in IPS (a manifestation of the curse of dimensionality). In favorable settings, the “second-order bias” of DR mitigates the curse of dimensionality to some extent and allows us to add more relevant features to model richer client contexts, and estimate more accurate models.

4 CHALLENGES AND EARLY PROMISE

While DR represents a promising step ahead with respect to both DM and IPS, there are several key challenges in networking applications. In this section, we highlight some of these challenges and sketch some initial solutions.

4.1 Challenges in applying DR in networking

While DR is a promising theoretical concept, there are several practical issues in applying it in a networking context:

- *Coverage and randomness:* The DR estimator, just like IPS, assumes that the old policy is stochastic, whose distribution is known. If not enough randomness is present

decisions that occur with low probability will generate high variance as term in the denominator $\mu_{\text{old}}(d_k|c_k)$ will be very small. Unfortunately, several applications are deterministic because they are designed to optimize performance or save cost, and randomization of policies is rarely used. While it is hard to augment an existing trace, we see an opportunity to persuade network operators and protocol designers to augment policies to introduce randomness where impact on overall performance is small.

- *Stationarity of policies:* The DR estimator described in §3 is limited to policies that are “history-agnostic” or stationary; a policy’s decision only depends on the current client. Most networking policies, however, are *non-stationary*, where a policy’s decision on client c_k depends also on the history $h_k = \{(c_i, d_i, r_i)\}_{i < k}$. In this case, the decision maker adapts its action-selection policy over time based on the observed history of client-action-reward triples.
- *System state of the world:* The theory of DR implicitly assumes that the new policy is being evaluated under the same “system states” as when the trace was collected. However, this is not the case in networking contexts where load and background traffic can affect the potential outcomes. For instance, we want to evaluate the performance of a server selection logic during peak hours, but the trace we have was collected during early morning hours. Thus, the DR estimator would produce biased results, since server performance tends to be different between early morning hours (with low traffic load) and peak hours (with high traffic load).
- *Hidden decision-reward coupling:* In many prior use cases for DR, there is an implicit notion of independence between the decision making process and the rewards; e.g., the value of an ad click does not change with more clients clicking on it. In a network setting, there can be confounding factors due to load and congestion induced by our current and previous decisions. For instance, if we assign clients to a specific server (e.g., we believed it would have good performance) then the performance of future clients using that server instance may be degraded due to increased load.

4.2 Early promise

In this section we first show the basic DR estimator can be extended to handle non-stationary (i.e., history-based) policies, and then show via simulation how a DR estimator can significantly reduce evaluation error of trace-driven evaluators used in scenarios from §2.

DR for non-stationary policies: One solution (used in ad recommendation [27]) is to maintain a separate history g_k that consists of only the clients on which the decision of the old policy and new policy matches. Formally, it starts with an empty history $h_1 = \emptyset$ for the old policy, an empty history $g_1 = \emptyset$ for the new policy, and the expectation of total rewards $M = 0$. For $k = 1, \dots, n$, it repeats the following steps:

- (1) Sample a decision d' from \mathbb{D} based on probability of $\mu_{\text{new}}(d|c_k, g_k)$;
- (2) If $d' = d_k$, then $M \leftarrow M + \sum_{d \in \mathbb{D}} \mu_{\text{new}}(d|c_k, g_k) \hat{r}(c_k, d) + \frac{\mu_{\text{old}}(d_k|c_k, g_k)}{\mu_{\text{new}}(d_k|c_k, g_k)} (r_k - \hat{r}_k)$; $g_{k+1} \leftarrow g_k \oplus (c_k, d_k, r_k)$;
- (3) Else, $g_{k+1} \leftarrow g_k$;
- (4) $h_{k+1} \leftarrow h_k \oplus (c_k, d_k, r_k)$;

Finally, it returns the average expected reward of the new policy $M/|g_{n+1}|$ as output. Intuitively, for each client, if the new policy makes the same decision as the old policy did in the trace, the algorithm will update the DR estimate by Eq. 2 on a per-client base (Step 2); otherwise, it will skip the client. This extended DR estimator is identical to the basic DR under the assumption of stationary policies, and enjoys the same property as the basic one [9].

Preliminary results: Next, we use synthetic traces to compare the DR estimator with trace-driven evaluators from prior research. Due to limited space, the full details of the simulation can be found in [2]. We define *evaluation error* by relative error between actual average reward V (ground truth) and its estimate \hat{V} , $\frac{|\hat{V} - V|}{V}$. We use the metric of relative error as an indicator that there exist evaluation bias and/or variance.

Figure 7a shows that the DR estimator can reduce the evaluation error caused by the selection bias in the example of Figure 4. Recall WISE [38] builds a Causal Bayesian Network (CBN) to identify the dependency between the response time of a request and the selection of frontend and backend. We simulate 500 clients for each measurement (arrow) in Figure 4, and 5 clients for each remaining choice of backend and frontend not shown in Figure 4. The new policy uses the same traffic pattern, except that 50% of ISP-1 clients use FE-1 and BE-2. Figure 7a shows DR’s evaluation error is about 32% lower than WISE. The evaluation in WISE is a sort of DM-like as it does not use observed data. DR avoids the negative impact of the selection bias by using the empirical data of a few ISP-1 clients who used FE-1 and BE-2.

Figure 7b shows that the DR estimator can reduce the evaluation error caused by the inaccurate assumption about the reward model in the example of Figure 2. We create a video session with 100 chunks and five bitrate levels, and the available bandwidth is a constant b . To evaluate the video quality of the new ABR policy [42], we first use the old ABR policy (a buffer-based ABR policy [13]) to collect throughput traces, where the observed throughput is $b * p(r)$, $p \leq 1$ and monotonically increases with the chosen bitrate. Now, we can compare the evaluation approach from the FastMPC paper and the evaluation when we use DR. Recall that FastMPC [42] assumes that the observed throughput is independent to the chosen bitrate. In Figure 7b, we see that DR’s evaluation error is 74% lower than the original evaluator. This is due to DR correcting the bias in the original evaluation method by using the unbiased quality measurement on chunks that use the same bitrate as in the observed trace.

Figure 7c shows that the DR estimator can reduce the variance in the example of Figure 5. To estimate the video quality of a new CDN and bitrate selection policy, the original

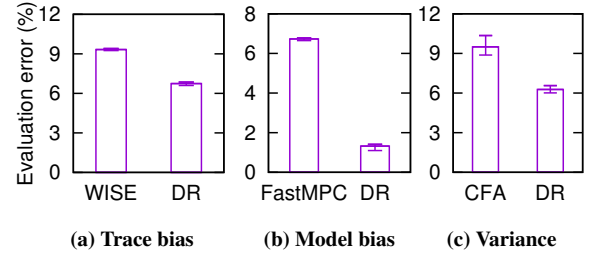


Figure 7: Simulation results showing preliminary promise of DR. The results show the mean, minimum and maximum of evaluation errors over 50 runs.

evaluator of CFA [15] uses a trace of clients with random CDN and bitrate selection, and focuses on the subset of clients who have the same decision in the new policy. We create a synthetic trace that has the same feature set and is generated by an old policy that randomly assigns clients to one of the available CDNs and bitrates, as in the original work. The DM estimates are based on a k -NN model [25] trained by the trace. Figure 7c shows DR’s evaluation error is about 36% lower than that of the original evaluator. Unlike previous examples where DR reduces the biases of evaluation models, this example illustrates the power of DR to reduce variance of evaluation results by giving *each* client an estimate using a (possibly biased) DM model.

4.3 Open questions

Modeling world state: We see an opportunity to address this using domain-specific knowledge. For instance, if we know that the peak-hour performance is on average 20% worse than morning-hour performance, we could create a new trace by degrading the performance in the trace by 20% (similar to [38]) and use the DR estimator on the new trace. That said, modeling such a “transition function” between network states may itself be error prone. We conjecture that this process can be automated by collecting a few samples from various network states, and then identifying the transition function using techniques such as transfer learning [33].

Tackling reward-decision coupling: In some sense, this is similar to the first challenge of making DR aware of network states, except that we also need to detect changes of network states during the course of running a policy. We posit that we could borrow ideas from change-point detection to infer if/when our decisions have affected the system state (e.g., [23, 26]). Specifically, we see an opportunity to detect such self-inflicted state changes by monitoring domain-specific metrics. For instance, in the previous example, if we monitor the load of each server as a proxy metric of the system states and use thresholds to decide whether the server state is “low load”, “high load”, or “overload”, then the DR estimator can use the empirical data in the trace when the network states match.

ACKNOWLEDGMENTS

This work was funded in part by NSF awards CNS-1565343, CNS-1345305, and DMS-1713003.

REFERENCES

- [1] The data-driven approach to network management: Innovation delivered. <https://goo.gl/vfLF5z>.
- [2] Simulation setup. <https://github.com/DoublyRobustEvaluation/DoublyRobustEvaluation/>.
- [3] Systems that learn initiative at csail. <http://stl.csail.mit.edu/>.
- [4] A. Akella and R. Mahajan. A call to arms for management plane analytics. In *Proc. HotNets*, page 4. ACM, 2014.
- [5] H. Bang and J. M. Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005.
- [6] J. R. Carpenter, M. G. Kenward, and S. Vansteelandt. A comparison of multiple imputation and doubly robust estimation for analyses with missing data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 169(3):571–584, 2006.
- [7] Y. Cheng, U. Hölzle, N. Cardwell, S. Savage, and G. M. Voelker. Monkey see, monkey do: A tool for tcp tracing and replaying. In *USENIX Annual Technical Conference, General Track*, 2004.
- [8] W. G. Cochran. *Sampling Techniques, 3rd Edition*. John Wiley, 1977.
- [9] M. Dudík, D. Erhan, J. Langford, L. Li, et al. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.
- [10] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on Networking (ToN)*, 9(4):392–403, 2001.
- [11] B. Han, F. Qian, L. Ji, V. Gopalakrishnan, and N. Bedminster. Mpdash: Adaptive video streaming over preference-aware multipath. In *CoNEXT*, pages 129–143, 2016.
- [12] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari. Confused, timid, and unstable: picking a video streaming rate is hard. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 225–238. ACM, 2012.
- [13] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. *ACM SIGCOMM Computer Communication Review*, 44(4):187–198, 2015.
- [14] J. Jiang, R. Das, G. Ananthanarayanan, P. A. Chou, V. Padmanabhan, V. Sekar, E. Dominique, M. Golszewski, D. Kukoleca, R. Vafin, et al. Via: Improving internet telephony call quality using predictive relay selection. In *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*, pages 286–299. ACM, 2016.
- [15] J. Jiang, V. Sekar, H. Milner, D. Shepherd, I. Stoica, and H. Zhang. Cfa: A practical prediction system for video qoe optimization. In *NSDI*, pages 137–150, 2016.
- [16] J. Jiang, V. Sekar, I. Stoica, and H. Zhang. Unleashing the potential of data-driven networking. In *Proceedings of 9th International Conference on COMMunication Systems & NETWORKS (COMSNET)*, 2017.
- [17] J. Jiang, V. Sekar, and H. Zhang. Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Streaming with Festive. In *ACM CoNEXT 2012*.
- [18] J. Jiang, S. Sun, V. Sekar, and H. Zhang. Pytheas: Enabling data-driven quality of experience optimization using group-based exploration-exploitation. In *NSDI*, pages 393–406, 2017.
- [19] N. Jiang and L. Li. Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 652–661, 2016.
- [20] X. Jin, Y. Li, D. Wei, S. Li, J. Gao, L. Xu, G. Li, W. Xu, and J. Rexford. Optimizing bulk transfers with software-defined optical wan. In *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*, pages 87–100. ACM, 2016.
- [21] J. D. Y. Kang and J. L. Schafer. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statist. Sci.*, 22(4):523–539, 11 2007.
- [22] F. Kaup, F. Michelinakis, N. Bui, J. Widmer, K. Wac, and D. Hausheer. Behind the nat-a measurement based evaluation of cellular service quality. In *Network and Service Management (CNSM), 2015 11th International Conference on*, pages 228–236. IEEE, 2015.
- [23] R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of change-points with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.
- [24] S. S. Krishnan and R. K. Sitaraman. Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. *IEEE/ACM Transactions on Networking*, 21(6):2001–2014, 2013.
- [25] D. T. Larose. K-nearest neighbor algorithm. *Discovering Knowledge in Data: An Introduction to Data Mining*, pages 90–106, 2005.
- [26] M. Lavielle. Using penalized contrasts for the change-point problem. *Signal processing*, 85(8):1501–1510, 2005.
- [27] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.
- [28] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM, 2011.
- [29] H. H. Liu, R. Viswanathan, M. Calder, A. Akella, R. Mahajan, J. Padhye, and M. Zhang. Efficiently delivering online services over integrated infrastructure. In *NSDI*, pages 77–90, 2016.
- [30] H. Mao, M. Alizadeh, I. Menache, and S. Kandula. Resource management with deep reinforcement learning. In *Proc. HotNets*, pages 50–56, 2016.
- [31] H. Mao, R. Netravali, and M. Alizadeh. Neural adaptive video streaming with pensieve. In *Proceedings of the 2016 conference on ACM SIGCOMM 2017 Conference*. ACM, 2017.
- [32] S. A. Murphy, M. J. van der Laan, J. M. Robins, and C. P. P. R. Group. Marginal mean models for dynamic regimes. *Journal of the American Statistical Association*, 96(456):1410–1423, 2001.
- [33] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.
- [34] J. M. Robins, A. Rotnitzky, and L. P. Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89(427):846–866, 1994.
- [35] M. Schapira, Y. Zhu, and J. Rexford. Putting bgp on the right path: A case for next-hop routing. In *Proc. HotNets*, page 3. ACM, 2010.
- [36] B. Schlinder, H. Kim, T. Cui, E. Katz-Bassett, H. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng. Engineering egress with edge fabric. In *SIGCOMM*. ACM, 2016.
- [37] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli. Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction. In *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*, pages 272–285. ACM, 2016.
- [38] M. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar. Answering what-if deployment and configuration questions with wise. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 99–110. ACM, 2008.
- [39] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar. Learning to route. In *Proc. HotNets*, 2017.
- [40] S. Venkataraman, Z. Yang, M. J. Franklin, B. Recht, and I. Stoica. Ernest: Efficient performance prediction for large-scale advanced analytics. In *NSDI*, pages 363–378, 2016.
- [41] K.-K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain, V. Lin, C. Rice, B. Rogan, A. Singh, B. Tanaka, M. Verma, P. Sood, M. Tariq, M. Tierney, D. Trumic, V. Valancius, C. Ying, M. Kallahalla, B. Koley, and A. Vahdat. Taking the edge off with espresso: Scale, reliability and programmability for global internet peering. In *SIGCOMM*. ACM, 2016.
- [42] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over http. *ACM SIGCOMM Computer Communication Review*, 45(4):325–338, 2015.
- [43] Y. Zaki, T. Pötsch, J. Chen, L. Subramanian, and C. Görg. Adaptive congestion control for unpredictable cellular networks. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 509–522. ACM, 2015.