Predictive Off-Policy Policy Evaluation for Nonstationary Decision Problems, with Applications to Digital Marketing

Philip S. Thomas
philipt@cs.cmu.edu
Carnegie Mellon University

Georgios Theocharous
theochar@adobe.com
Adobe Research

Mohammad Ghavamzadeh
ghavamza@adobe.com
Adobe Research

Ishan Durugkar
idurugkar@cs.umass.edu
University of Massachusetts Amherst

Emma Brunskill
ebrun@cs.cmu.edu
Carnegie Mellon University

Abstract

In this paper we consider the problem of evaluating one digital marketing policy (or more generally, a policy for an MDP with unknown transition and reward functions) using data collected from the execution of a different policy. We call this problem *off-policy policy evaluation*. Existing methods for off-policy policy evaluation assume that the transition and reward functions of the MDP are stationary—an assumption that is typically false, particularly for digital marketing applications. This means that existing off-policy policy evaluation methods are *reactive* to nonstationarity, in that they slowly correct for changes after they occur. We argue that off-policy policy evaluation for nonstationary MDPs can be phrased as a time series prediction problem, which results in *predictive* methods that can anticipate changes before they happen. We therefore propose a synthesis of existing off-policy policy evaluation methods with existing time series prediction methods, which we show results in a drastic reduction of mean squared error when evaluating policies using real digital marketing data set.

Introduction

We assume that the reader is familiar with reinforcement learning (Sutton and Barto 1998). In this paper we study the problem of evaluating a policy for a *Markov decision process* (MDP) with unknown transition and reward functions using historical data collected from another (usually different) policy. We call this problem *off-policy policy evaluation* (OPE). Methods for OPE are important because they can tell a practitioner what to expect if a new policy produced by a reinforcement learning algorithm were used, without requiring the new policy to actually be used. This is particularly valuable for high-risk applications where the use of a bad policy could be costly or dangerous. For example, we focus on the application of OPE methods to digital marketing policies, where significant over or under-predictions of performance can be costly.

Several powerful OPE algorithms have been developed both within the reinforcement learning and bandit communities (Precup, Sutton, and Singh 2000; Dudík, Langford, and Li 2011; Bottou et al. 2013; Jiang and Li 2016; Thomas and Brunskill 2016). However, these methods assume that the environment is stationary—that the transition and reward functions of the MDP do not change between episodes. There has been increasing concern among practitioners in industry that the predictions of these OPE methods are invalid because the environment is really nonstationary.

In this paper we study methods for OPE when the environment is nonstationary—when the transition and reward functions of the MDP can change between episodes. After providing background and the problem setting, we further motivate the necessity for such methods using data from a real problem. We then propose a new approach to off-policy policy evaluation—phrasing it as a time series prediction problem. This results in methods that are *predictive* rather than reactive. We evaluate one such predictive method using real data from a digital marketing application and find that it drastically reduces the mean squared error of predictions relative to existing methods. Moreover, we find that the primary source of error when using current standard approaches to evaluate digital marketing policies appears to be nonstationarity, which highlights the importance of applying methods that directly mitigate nonstationarity.

Nonstationary Markov Decision Processes

We use the notational standard MDPNv1 (Thomas 2015) for MDPs. We consider only finite-horizon MDPs, i.e., we assume that L is finite. We define a *nonstationary MDP* to be an MDP where the transition function, P, and the reward function, R, change over time. Changes to P and R within an episode can be modeled by including the time step as part of the state. However, the standard definition of an MDP does not allow P and R to change across episodes. To allow this, we replace the transition function, P, and the reward function, R, in the definition of an MDP with sequences of transition and reward functions: $(P^\iota)_{\iota=0}^\infty$ and $(R^\iota)_{\iota=0}^\infty$. The transition and reward functions for the ι^{th} episode are then P^ι and R^ι. Throughout this paper, we reserve the symbol ι to denote episode numbers.

The way that the environment changes might depend on the actions chosen by the agent. For example, if the MDP models an automobile, then the amount that the tires wear over time may be impacted by the policy that is used (a policy that uses the brakes more may wear down the brake pads faster). So, P^ι may depend on actions that are chosen during episode $\hat\iota$ if $\hat\iota < \iota$. To model this, we make each P^ι and R^ι a random variable that may depend on events from past episodes.

So, a nonstationary MDP is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, (P^\iota)_{\iota=0}^{\infty}, (R^\iota)_{\iota=0}^{\infty}, d_0, \gamma)$, where $\mathcal{S}, \mathcal{A}, \mathcal{R}, d_0$, and γ are as defined by MDPNv1 for stationary MDPs, and P^ι and R^ι are random variables that denote the transition and reward functions for the ι^{th} episode of the nonstationary MDP. We write S_t^ι, A_t^ι, and R_t^ι to denote the state, action, and reward at time t during the ι^{th} episode.[1] Let

$$J(\pi, \iota) := \mathbf{E}\left[\sum_{t=0}^{L-1} \gamma^t R_t^\iota \middle| \pi, P^\iota, R^\iota\right]$$

be the expected return during the ι^{th} episode if the policy π is used.[2] If the MDP is stationary, then we suppress the dependencies on ι and write, for example, $J(\pi)$. Lastly, let $H^\iota := (S_0^\iota, A_0^\iota, R_0^\iota, S_1^\iota, \dots)$ denote the ι^{th} *trajectory*, or the *history of the ι^{th} episode* for alliteration.

Off-Policy Policy Evaluation (OPE)

For conventional OPE, we assume that the environment is a stationary MDP. Let π_e be a policy called the *evaluation policy*. We are given historical data, D, which consists of $n \in \mathbb{N}_{\geq 1}$ trajectories and the policies that generated them: $D := (H^\iota, \pi^\iota)_{\iota=0}^{n-1}$, where H^ι was generated by running the *behavior policy*, π^ι, on the ι^{th} episode. Our goal is to estimate the performance of π_e, $J(\pi_e)$, using the historical data, D.

One popular method for OPE is *importance sampling* (Kahn 1955; Precup 2000). The importance sampling estimator of $J(\pi_e)$ produced from a trajectory, $H := (S_0, A_0, R_0, S_1, \dots)$, that was generated by the policy π_b is given by:

$$\text{IS}(\pi_e | H, \pi_b) := \prod_{t=0}^{L-1} \frac{\pi_e(A_t|S_t)}{\pi_b(A_t|S_t)} \sum_{t=0}^{L-1} \gamma^t R_t.$$

The IS estimator is an unbiased estimator of $J(\pi_e)$. That is, $\mathbf{E}[\text{IS}(\pi_e|H, \pi_b)|H \sim \pi_b] = J(\pi_b)$, where $H \sim \pi_b$ denotes that the trajectory H was generated using the behavior policy π_b.

More advanced methods exist for OPE, including weighted and per-decision importance sampling (Precup 2000), as well as variants that use control variates (Jiang and Li 2016). The question of which method of OPE to use as a subroutine for the methods that we propose is tangent to the message of this work. We therefore write $\text{OPE}(\pi_e, \iota|D)$ to denote any estimate of $J(\pi_e, \iota)$ created from D, and we make no assumptions about the veracity of this estimate. For example, one might select $\text{OPE}(\pi_e, \iota|D) := \text{IS}(\pi_e|H^\iota, \pi^\iota)$ in order to use ordinary importance sampling, or

$$\text{OPE}(\pi_e, \iota|D) := \frac{\text{IS}(\pi_e|H^\iota, \pi^\iota)}{\frac{1}{n}\sum_{\hat\iota=0}^{n-1} \prod_{t=0}^{L-1} \frac{\pi_e(A_t^{\hat\iota}|S_t^{\hat\iota})}{\pi_b(A_t^{\hat\iota}|S_t^{\hat\iota})}},$$

to use *weighted importance sampling* (Precup 2000). Notice that the weighted importance sampling estimate of $J(\pi_e, \iota)$ depends on all trajectories in D, not just H^ι, and so we make OPE take as input D rather than just a single trajectory, H.

Despite the fact that nearly all real-world problems are nonstationary, the current standard method used for off-policy evaluation assumes that the environment is stationary. The standard approach is simple: the estimate, $\hat J(\pi, n)$ of $J(\pi, n)$, the performance during the next episode, is the (sometimes weighted) average the OPE estimates from each trajectory:

$$\hat J(\pi, n) := \frac{1}{n}\sum_{\iota=0}^{n-1} \text{OPE}(\pi_e, \iota|\mathcal{D}). \tag{1}$$

Variants of this scheme have recently been proposed for digital marketing applications (Thomas, Theocharous, and Ghavamzadeh 2015; Jiang and Li 2016), and have been used in a wide variety of application areas including state-space models (finance, signal-tracking), evolutionary models (molecular physics and biology, genetics) and others (Liu 2001, Section 3).

The problem with this standard scheme is that it produces *reactive* behavior. That is, consider what happens in a particularly simple example: if $J(\pi_e, \iota)$ is a linear function of ι. For example, let $J(\pi_e, \iota)$ have a negative slope, so that the performance of π_e decreases as more episodes pass. The simple scheme in (1) will estimate $J(\pi_e, \iota)$ using the mean estimate from *previous* episodes, which will tend to be too large, as depicted in Figure 1. In the next section we show how we can formalize the problem of off-policy policy evaluation for nonstationary Markov decision problems before proposing our new predictive approach.

Nonstationary Off-Policy Policy Evaluation

Nonstationary Off-Policy Policy Evaluation (NOPE) is simply OPE for nonstationary MDPs. In this setting, the goal is to use D to estimate $J(\pi_e, n)$—the performance of π_e during the *next episode*.

Notice that we have not made assumptions about how the transition and reward functions of the nonstationary MDP change. For some applications, they may drift slowly, making $J(\pi_e, \iota)$ change slowly with ι. For example, this sort of drift may occur due to mechanical wear in a robot. For other applications, $J(\pi_e, \iota)$ may be fixed for some number of episodes, and then make a large jump. For example,

[1] Although the reward function, R^ι, is denoted by symbols similar to the reward R_t^ι, it should be clear from context which is intended. Furthermore, rewards always have subscripts, and the reward function never has a subscript.

[2] To avoid overly-complex notation, we suppress the dependency of J (and later other terms) on the particular nonstationary MDP that we are considering. Furthermore, notice that $J(\pi, \iota)$ is a random variable until P^ι and R^ι are sampled.

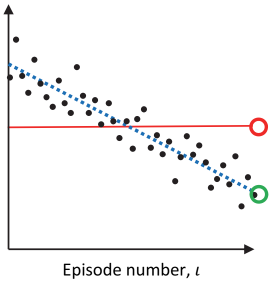

Figure 1: This illustration depicts an example of how the existing standard OPE methods produce *reactive* behavior, and is hand-drawn to provide intuition (often the variance of the black points will be much higher). Here the dotted blue line depicts $J(\pi_e, \iota)$ for various ι. The black dots denote $\text{OPE}(\pi_e, \iota|D)$ for various ι. Notice that each $\text{OPE}(\pi_e, \iota|D)$ is a decent estimate of $J(\pi_e, \iota)$, which changes with ι. Our goal is to estimate $J(\pi_e, n)$—the performance of the policy during the *next* episode. That is, our goal is to predict the vertical position of the green circle. However, by averaging the OPE estimates, we get the red circle, which is a reasonable prediction of performance in the past. As more data arrives (n increases) the predictions will decrease, but will always remain behind the target value of $J(\pi_e, n)$.

this sort of jump may occur in digital marketing applications (Theocharous, Thomas, and Ghavamzadeh 2015) due to media coverage of a relevant topic rapidly changing public opinion of a product. In yet other applications, the environment may include both large jumps and smooth drift.

Notice that NOPE can range from trivial to completely intractable. If the MDP has few states and actions, changes slowly between episodes, and the evaluation policy is similar to the behavior policy, then we should be able to get accurate off-policy estimates. On the other extreme, if for each episode the MDP's transition and reward functions are drawn randomly (or adversarially) from a wide distribution, then producing accurate estimates of $J(\pi_e, n)$ may be intractable.

Motivating Example

Digital marketing is a major use of reinforcement learning algorithms in industry. When a person visits the website of a company, she is often shown a list of current promotions. In order for the display of these promotions to be effective, it must be properly targeted based on the known information about the person (e.g., her interests, past travel behavior, or income). The problem of automatically deciding which promotion (sometimes called a *campaign*) to show to the visitor of a website is typically treated as a bandit problem (Li et al. 2010) or a reinforcement learning problem (Theocharous, Thomas, and Ghavamzadeh 2015).

Each visitor of the website corresponds to an episode, the known information about the visitor is the *state* or *observation*, and the decision of which promotions to show is an *action*. If the visitor clicks on a promotion, then the system is provided with a reward of $+1$, and if the visitor does not click, then the system is provided with a reward of 0. Each trajectory corresponds to one visit by one user in the bandit setting, and each trajectory corresponds to one user's sequence of interactions with the website in the reinforcement learning setting.

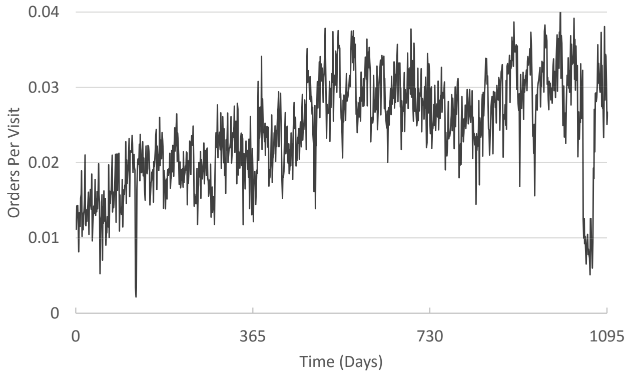

Figure 2: Moving average plot of $\text{OPE}(\pi_e, \iota|D)$ for various ι, on the real-world digital marketing data. This data spans three years. Notice that the performance of the policy climbs significantly over time, so naïve predictions will tend to be too low.

The system's goal is to determine how to select actions (select promotions to display) based on the available observations (the known information of the visitor) such that the reward is maximized (the number of clicks is maximized). In the bandit setting $J(\pi_e, \iota)$ is the expected number of clicks *per visit*, called the *click through rate* (CTR), while in the reinforcement learning setting it is the expected number of clicks *per user*, called the *life-time value* (LTV). We consider a setup that differs slightly from the standard problem formulation: we measure performance in terms of the number of actual orders, rather than clicks, since a click that does not result in an order (sale) is not inherently beneficial. We refer to this metric as the *orders per visit* (OPV) metric.

In order to determine how much of a problem nonstationarity really is, we collected three years of data from the website of one of Adobe's digital marketing solutions customers. For simplicity here, we selected the evaluation policy to be equal to the behavior policy, so that all of the importance weights are one. Figure 2 summarizes the resulting data.

In this data it is evident that there is significant nonstationarity—the OPV varied drastically over the span of the plot. This is also not just an artifact of high variance: using Student's t-test we can conclude that the expected return during the first 500 and second 500 days was different with $p < 0.005$. This is compelling evidence that we cannot ignore nonstationarity in our customers' data when providing predictions of the expected future performance of our digital marketing algorithms, and is compelling real-world motivation for developing NOPE algorithms.

Predictive Off-Policy Evaluation using Time Series Methods

The primary contribution of this paper is an observation that, in retrospect, is obvious: **NOPE is a time series prediction problem, and is particularly important for digital marketing.** Let $x_\iota = \iota$ and $Y_\iota = \text{OPE}(\pi_e, \iota|D)$ for $\iota \in \{1, \ldots, n-1\}$. This makes x an array of n times (each episode corresponds to one unit of time) and y an array of the corresponding n observations. Our goal is to predict the expected value of the next point in this time series, which will occur at $x_n = n$. Pseudocode for this *time series prediction* (TSP) approach is given in Algorithm 1.

Algorithm 1 Time Series Prediction (TSP)

```
Input: Evaluation policy, $\pi_e$, historical data, $D := (H^\iota, \pi^\iota)_{\iota=0}^{n-1}$, and a time-series prediction algorithm (and its hyper-parameters).
Create arrays $x$ and $y$, both of length $n$.
for $\iota = 0$ to $n-1$ do
    $x_\iota \leftarrow \iota$
    $y_\iota \leftarrow \text{OPE}(\pi_e, \iota|D)$
end for
Train a time-series prediction algorithm on $x, y$.
return the time-series prediction algorithm's prediction for time $n$.
```

When considering using time-series prediction methods for off-policy policy evaluation, it is important that we establish that the underlying process is actually nonstationary. One popular method for determining whether a process is stationary or nonstationary is to report the sample *autocorrelation function* (ACF):

$$\text{ACF}_h := \frac{\mathbf{E}[(X_{t+h} - \mu)(X_t - \mu)]}{\mathbf{E}[(X_t - \mu)^2]},$$

where h is a parameter called the *lag* (which is selected by the researcher), X_t is the time series, and μ is the mean of the time series. For a stationary time series, the ACF will drop to zero relatively quickly, while the ACF of nonstationary data decreases slowly.

ARIMA models are models of time series data that can capture many different sources of nonstationarity. The time series prediction algorithm that we use in our experiments is the R forecast package for fitting ARIMA models, as described by Hyndman and Khandakar (2008).

Empirical Studies

In this section we show that, despite the lack of theoretical results about using TSP for NOPE, it performs remarkably well on real data. Because our experiments use real-world data, we do not know ground truth—we have $\text{OPE}(\pi_e, \iota|D)$ for a series of ι, but we do not know $J(\pi_e, \iota)$ for any ι. This makes evaluating our methods challenging—we cannot, for example, compute the true error or mean squared error of estimates. We therefore estimate the mean squared error directly from the data as follows.

For each $\iota \in \{1, \ldots, n-1\}$ we compute each method's output, $\hat{y}_\iota$, given all of the previous data, $D_{\iota-1} := (H^{\hat{\iota}}, \pi^{\hat{\iota}})_{\hat{\iota}=0}^{\iota-1}$. That is, $D_{\iota-1}$ denotes the data set D, but truncated so that it only includes the first ι episodes of data—episodes 0 through $\iota - 1$. We then compute the observed next value, $y_\iota = \text{OPE}(\pi_e, \iota|D_\iota)$ and the TSP prediction, $\hat{y}$, computed using $D_{\iota-1}$. From these, we compute the squared error, $(\hat{y}_\iota - y_\iota)^2$, and we report the (root) mean squared error over all ι. We perform this experiment using both the current standard approach defined in (1) and Algorithm 1, TSP.

Notice that this scheme is not perfect. Even if an estimator perfectly predicts $J(\pi_e, \iota)$ for every ι, it will be reported as having non-zero mean squared error. This is due to the high variance of OPE (which is used to compute the target values, y), which gets conflated with the variance of $\hat{y}$ in our estimate of mean squared error. Although this means that the mean squared errors that we report are not good estimates of the mean squared error of the estimators, $\hat{y}$, the variance-conflation problem impacts all methods nearly equally. So, in the absence of ground truth knowledge, the reported mean squared error values are a reasonable measure of how accurate the methods are relative to each other.

Nonstationary Mountain Car

For this domain we modified the canonical mountain car domain (Sutton and Barto 1998) to include nonstationarity. Specifically, we simulated mechanical wear (e.g., on the tire treads) by decreasing the car's acceleration with each episode. Even more specifically, in the update equation for mountain car, we multiplied the acceleration caused by the agent's action by a decay term:

$$\delta = 1.0 - \left(\frac{\iota}{1.8m}\right)^2,$$

where m is the maximum number of episodes, 20,000. We used a near-random behavior policy and a mediocre evaluation policy (an optimal policy for the ordinary mountain car domain has an expected return around -150). A moving-average plot of the resulting data is provided in Figure 3. Notice that a moving average with $k = 1$ is just a direct plot of $\text{OPE}(\pi_e, \iota|D)$ for all ι, which has high variance. For this domain, and all others, we used ordinary importance sampling for OPE.

Real-World Data

The next domain that we consider is digital marketing, as described previously, using data from the websites of three large companies. We refer to these three data sets as DM1, DM2, and DM3, which use a mixture of on and off-policy data. As we will see, DM1 has large amounts of nonstationarity, DM2 has mild nonstationarity, and DM3 has little nonstationarity. Still, across all three domains, we find that TSP produces lower (root) mean squared errors.

Results

We applied our TSP algorithm for NOPE, described in Algorithm 1, to the nonstationary mountain car and digital marketing data sets. The following plots all take the same form:

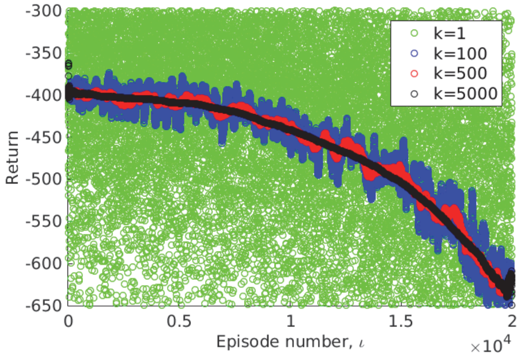

Figure 3: Moving average plot of the importance weighted returns (the OPE estimates) for the nonstationary mountain car domain, where k denotes the number of points used by the moving average.

the first plots for each domain are autocorrelation plots that show whether or not there appears to be nonstationarity in the data. As a rule of thumb, if the ACF values are within the dotted blue lines, then there is not sufficient evidence to conclude that there is nonstationarity. However, if the ACF values lie outside the dotted blue lines, it suggests that there is nonstationarity.

The subsequent plots for each domain depict the expected return (which is the expected OPV for the digital marketing data sets) as predicted by several different methods. The black curves are the target values—the moving average of the OPE estimates over a small time interval. For each episode number, our goal is to compute the value of the black curve given all of the previous values of the black curve. The blue curve does this using the standard method, which simply averages the previous black points. The red curve is our newly proposed method, which uses ARIMA to predict the next point on the black curve—to predict the performance of the evaluation policy during the next episode. Above some of the plots we report the sample *root mean squared error* (RMSE) for our method, *tsp*, and the standard method, *standard*.

First consider the results on the mountain car domain, which are provided in Figure 4. Notice that the autocorrelation plot suggests that there is nonstationarity—decaying the acceleration of the car does impact the performance of the evaluation policy. Notice that the black curve (the performance of the behavior policy) decreases over time from around -400 to around -650. The predictions of the standard method (blue curve) lag behind the true values due to their reactive nature, while our approach (red curve) accurately tracks the target values. This is further verified by the sample RMSE values: our method achieves a RMSE of 17.201 while the standard only achieves a RMSE of 68.152. This is compelling evidence that treating the problem as a time series prediction problem yields significantly better predictions than the standard approach.

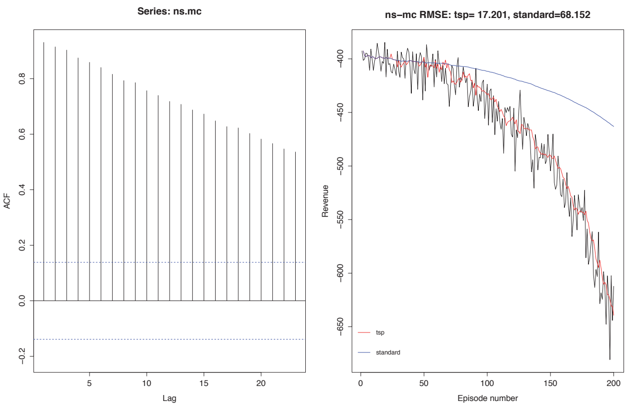

Figure 4: Results on the nonstationary mountain car domain. The left plot shows the autocorellation for the time series, where it is obvious the signal is nonstationary. The right plot compares the TSP approach with the standard. TSP outperforms the standard approach, since the series is nonstationary. The time series was aggregated at every 100 observations.

Next consider the results on the digital marketing data sets. As shown in Figure 5, DM1 shows significant nonstationarity, which TSP handles much better than the standard method. The sample RMSE for TSP on DM1 is 0.00334 while the sample RMSE for the standard approach is 0.006811. As shown in 6, DM2 shows mild nonstationarity, which TSP is still able to leverage to get an RMSE of 0.000248, which is much lower than the RMSE of the standard approach, 0.00029. Finally, as shown in Figure 7, in DM3, there is little nonstationarity, and both approaches perform similarly. This shows that using TSP methods does not tend to be detrimental when nonstationarity is not present.

Conclusion

In summary, we have proposed a new approach to off-policy policy evaluation for applications where the environment may be nonstationary. Our entire contribution can be summarized by the following statement: **off-policy policy evaluation for nonstationary MDPs, and particularly for digital marketing applications, should be treated as a time series prediction problem**. We showed empirically, using real and synthetic data, that an off-the-shelf time series prediction algorithm (ARIMA) can produce more accurate estimates of the performance of an evaluation policy from historical data than the existing standard approach.

It is our hope that this simple observation will encourage other practitioners to apply time series methods to off-policy policy evaluation problems, and theoreticians to develop additional theory that brings together these two previously disparate fields to produce new and more reliable methods.

References

Bottou, L.; Peters, J.; Quiñonero-Candela, J.; Charles, D. X.; Chickering, D. M.; Portugaly, E.; Ray, D.; Simard, P.; and Snelson, E. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research* 14:3207–3260.

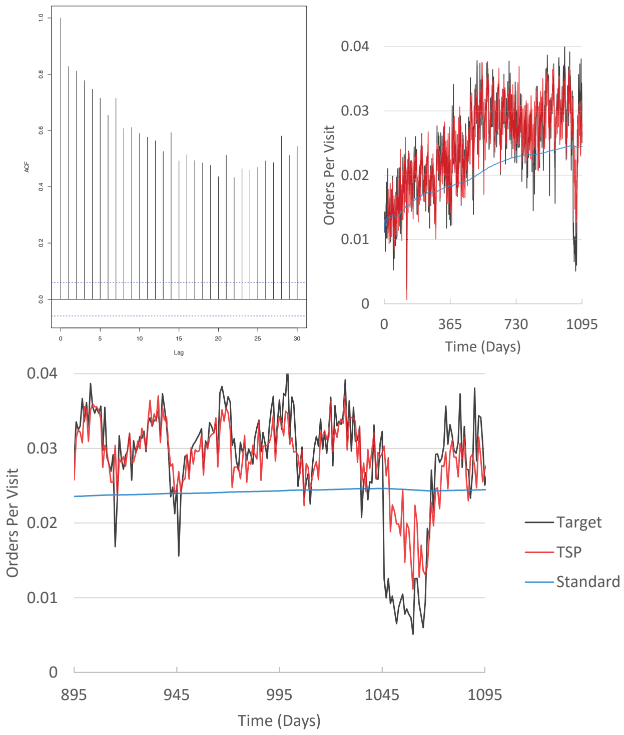

Figure 5: Results using the DM1 data set. The third plot is the same as the second, but zoomed in on approximately the last 200 days.

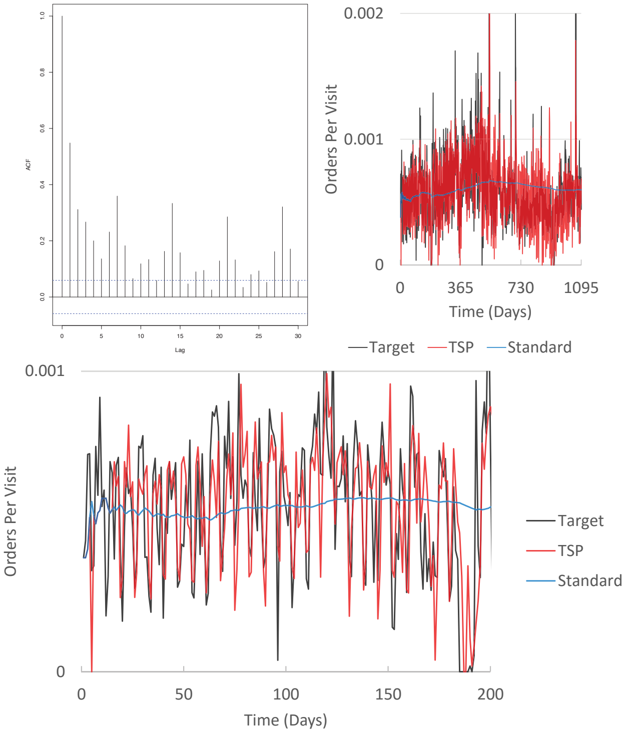

Figure 6: Results using the DM2 data set. The third plot is the same as the second, but zoomed in on the first 200 days.

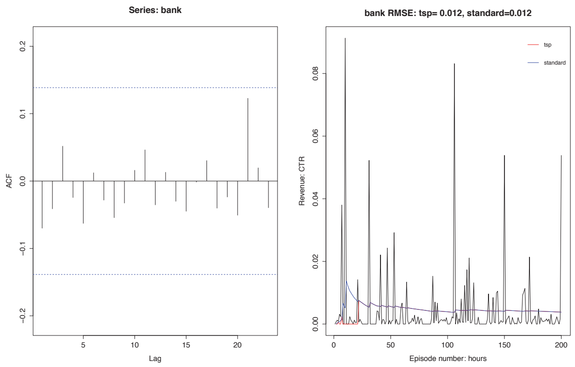

Figure 7: Results using the DM3 data set.

Dudík, M.; Langford, J.; and Li, L. 2011. Doubly robust policy evaluation and learning. In *Proceedings of the Twenty-Eighth International Conference on Machine Learning*, 1097–1104.

Hyndman, R. J., and Khandakar, Y. 2008. Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software* 26(3):1–22.

Jiang, N., and Li, L. 2016. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*.

Kahn, H. 1955. Use of different Monte Carlo sampling techniques. Technical Report P-766, The RAND Corporation.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW*, 661–670.

Liu, J. S. 2001. *Monte Carlo strategies in scientific computing*. Springer.

Precup, D.; Sutton, R. S.; and Singh, S. 2000. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning*, 759–766.

Precup, D. 2000. *Temporal Abstraction in Reinforcement Learning*. Ph.D. Dissertation, University of Massachusetts Amherst.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Theocharous, G.; Thomas, P. S.; and Ghavamzadeh, M. 2015. Personalized ad recommendation systems for lifetime value optimization with guarantees. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Thomas, P. S., and Brunskill, E. 2016. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*.

Thomas, P. S.; Theocharous, G.; and Ghavamzadeh, M. 2015. High confidence off-policy evaluation. In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence*.

Thomas, P. S. 2015. A notation for Markov decision processes. *ArXiv* arXiv:1512.09075v1.