Adaptive Importance Sampling with Automatic Model Selection in Value Function Approximation

Hirotaka Hachiya† **Takayuki Akiyama**† **Masashi Sugiyama**† and **Jan Peters**‡
†Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan
{hachiya@sg., akiyama@sg., sugi@}cs.titech.ac.jp
‡Max-Planck Institute for Biological Cybernetics, 72076 Tuebingen, Germany
jan.peters@tuebingen.mpg.de

Abstract

Off-policy reinforcement learning is aimed at efficiently reusing data samples gathered in the past, which is an essential problem for physically grounded AI as experiments are usually prohibitively expensive. A common approach is to use importance sampling techniques for compensating for the bias caused by the difference between data-sampling policies and the target policy. However, existing off-policy methods do not often take the variance of value function estimators explicitly into account and therefore their performance tends to be unstable. To cope with this problem, we propose using an adaptive importance sampling technique which allows us to actively control the trade-off between bias and variance. We further provide a method for optimally determining the trade-off parameter based on a variant of cross-validation. We demonstrate the usefulness of the proposed approach through simulations.

Introduction

Policy iteration is a reinforcement learning setup where the optimal policy is obtained by iteratively performing policy evaluation and improvement steps (Sutton & Barto 1998). When policies are updated, many popular policy iteration methods require the user to gather new data samples following the updated policy and the new samples are used for value function approximation. However, this approach is inefficient particularly when the sampling cost is high since previously gathered data samples are simply discarded; it would be more efficient if we could reuse the data collected in the past. A situation where the sampling policy (a policy used for gathering data samples) and the current policy are different is called *off-policy* reinforcement learning (Sutton & Barto 1998) with few notable exceptions such as Q-learning (Watkins 1989) and policy search by dynamic programming (Bagnell *et al.* 2003).

In the off-policy situation, simply employing a standard policy iteration method such as *least-squares* policy iteration (Lagoudakis & Parr 2003) does not lead to the optimal policy as the sample distribution is determined by the policies. Therefore, the sampling policy can introduce bias into the sample distribution. This distribution mismatch problem could be eased by the use of *importance sampling* techniques (Fishman 1996), which cancel the bias asymptotically. However, the approximation error is not necessarily small when the bias is reduced by importance sampling; the variance of estimators should also be taken into account since the approximation error is the sum of squared bias and variance. Due to large variance, existing importance sampling techniques tend to be unstable (Sutton & Barto 1998; Precup, Sutton, & Singh 2000).

To overcome the instability problem, we propose using an *adaptive importance sampling* technique used in statistics (Shimodaira 2000). The proposed adaptive method, which smoothly bridges the ordinary estimator and the importance-weighted estimator, allows us to control the trade-off between bias and variance. Thus, given that the trade-off parameter is determined carefully, the optimal performance can be achieved in terms of both bias and variance. However, the optimal value of the trade-off parameter is heavily dependent on data samples and policies. For this reason, using a prefixed parameter value may not be always effective in practice.

In order to optimally choose the value of the trade-off parameter, we reformulate the value function approximation problem as a supervised regression problem and propose using an automatic model selection method based on a variant of cross-validation (Sugiyama, Krauledat, & Müller 2007). The method called *importance-weighted cross-validation* enables us to estimate the approximation error of value functions in an almost unbiased manner even under off-policy situations. Thus we can actively determine the trade-off parameter based on data samples at hand. We demonstrate the usefulness of the proposed approach through simulations.

Background and Notation

In this section, we formulate the reinforcement learning problem.

Markov Decision Processes. Let us consider a Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, P_{\mathrm{T}}, R, \gamma)$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $P_{\mathrm{T}}(s'|s,a)$ $(\in [0,1])$ is the transition probability density from state s to next state s' when action a is taken, $R(s,a,s')$ $(\in \mathbb{R})$ is a reward for

transition from s to s' by taking action a, and γ ($\in [0,1)$) is the discount factor for future rewards. Let $\pi(a|s)$ ($\in [0,1]$) be a stochastic policy which is the conditional probability density of taking action a given state s. Let $Q^\pi(s,a)$ ($\in \mathbb{R}$) be a state-action value function for policy π which is the expected discounted sum of rewards the agent will obtain when taking action a in state s and following policy π thereafter.

$$Q^\pi(s,a) \equiv \mathop{\mathbb{E}}_{\pi,P_{\mathrm{T}}}\left[\sum_{n=1}^{\infty}\gamma^{n-1}R(s_n,a_n)\Big|s_1=s,a_1=a\right],$$

where $\mathbb{E}_{\pi,P_{\mathrm{T}}}$ denotes the expectation over $\{s_n,a_n\}_{n=1}^{\infty}$ following $\pi(a_n|s_n)$ and $P_{\mathrm{T}}(s_{n+1}|s_n,a_n)$, and $R(s,a)$ is the expected reward when the agent takes action a in state s:

$$R(s,a) = \mathop{\mathbb{E}}_{P_{\mathrm{T}}(s'|s,a)}\left[R(s,a,s')\right].$$

The goal of reinforcement learning is to obtain the policy which maximizes the sum of future rewards; the optimal policy may be expressed as

$$\pi^*(a|s) \equiv \delta(a - \arg\max_{a'} Q^*(s,a')),$$

where $\delta(\cdot)$ is the delta function and $Q^*(s,a)$ is the *optimal* state-action value function defined by

$$Q^*(s,a) \equiv \max_{\pi} Q^\pi(s,a).$$

Policy Iteration. Computing the value function $Q^\pi(s,a)$ is called *policy evaluation*. Using $Q^\pi(s,a)$, we can find a better policy $\pi'(a|s)$ by

$$\pi'(a|s) = \delta(a - \arg\max_{a'} Q^\pi(s,a')).$$

This is called (greedy) *policy improvement*. It is known that repeating policy evaluation and policy improvement results in the optimal policy $\pi^*(a|s)$ (Sutton & Barto 1998). This entire process is called *policy iteration*. For technical reasons, we assume that all policies are strictly positive (i.e., all actions have non-zero probabilities). In order to guarantee this, we update a policy by softmax:

$$\pi'(a|s) = \frac{\exp(Q^\pi(s,a)/\tau)}{\int_{\mathcal{A}}\exp(Q^\pi(s,a')/\tau)da'}, \tag{1}$$

where τ is a positive parameter which determines the randomness of new policy π'.

Value Function Approximation. Although policy iteration is guaranteed to produce the optimal policy, it is often computationally intractable since $|\mathcal{S}|\times|\mathcal{A}|$ is very large; $|\mathcal{S}|$ or $|\mathcal{A}|$ becomes infinity when the state space or action space is continuous. To overcome this problem, the references (Sutton & Barto 1998; Precup, Sutton, & Dasgupta 2001; Lagoudakis & Parr 2003) proposed approximating the state-action value function $Q^\pi(s,a)$ using the following linear model:

$$\widehat{Q}^\pi(s,a;\boldsymbol{\theta}) \equiv \sum_{b=1}^{B}\theta_b\phi_b(s,a) = \boldsymbol{\theta}^\top\boldsymbol{\phi}(s,a),$$

where $\boldsymbol{\phi}(s,a) = (\phi_1(s,a),\phi_2(s,a),\ldots,\phi_B(s,a))^\top$ are the fixed basis functions, $\top$ denotes the transpose, B is the number of basis functions, and $\boldsymbol{\theta} = (\theta_1,\theta_2,\ldots,\theta_B)^\top$ are model parameters. Note that B is usually chosen to be much smaller than $|\mathcal{S}|\times|\mathcal{A}|$. For N-step transitions, we ideally want to learn the parameters $\boldsymbol{\theta}$ so that the approximation error G is minimized:

$$\begin{aligned}\boldsymbol{\theta}^* &\equiv \arg\min_{\boldsymbol{\theta}} G,\\ G &\equiv \mathop{\mathbb{E}}_{P_1,\pi,P_{\mathrm{T}}}\left[\frac{1}{N}\sum_{n=1}^{N} g(s_n,a_n;\boldsymbol{\theta})\right], \qquad (2)\\ g(s,a;\boldsymbol{\theta}) &\equiv \left(\widehat{Q}^\pi(s,a;\boldsymbol{\theta}) - Q^\pi(s,a)\right)^2,\end{aligned}$$

where $g(s,a;\boldsymbol{\theta})$ is a cost function corresponding to the approximation error of $\widehat{Q}^\pi$ for each pair (s,a) and $\mathbb{E}_{P_1,\pi,P_{\mathrm{T}}}$ denotes the expectation over $\{s_n,a_n\}_{n=1}^N$ following the initial probability density $P_1(s_1)$, the policy $\pi(a_n|s_n)$, and the transition probability $P_{\mathrm{T}}(s_{n+1}|s_n,a_n)$. A fundamental problem of the above formulation is that the target function $Q^\pi(s,a)$ cannot be observed directly. To cope with this problem, we use the Bellman residual cost function (Schoknecht 2003; Lagoudakis & Parr 2003) defined by

$$\begin{aligned}&g_{\mathrm{BR}}(s,a;\boldsymbol{\theta})\\ &\equiv \left(\widehat{Q}^\pi(s,a;\boldsymbol{\theta}) - R(s,a) - \gamma \mathop{\mathbb{E}}_{\substack{\pi(a'|s')\\ P_{\mathrm{T}}(s'|s,a)}}\left[\widehat{Q}^\pi(s',a';\boldsymbol{\theta})\right]\right)^2.\end{aligned}$$

On-policy vs. Off-policy. We suppose that a data set consisting of M episodes of N steps is available. The agent initially starts from a randomly selected state s_1 following the initial probability density $P_1(s)$ and chooses an action based on a *sampling policy* $\widetilde{\pi}(a_n|s_n)$. Then the agent makes a transition following $P_{\mathrm{T}}(s_{n+1}|s_n,a_n)$ and receives a reward r_n ($= R(s_n,a_n,s_{n+1})$). This is repeated for N steps—thus the training data $\mathcal{D}^{\widetilde{\pi}}$ is expressed as

$$\mathcal{D}^{\widetilde{\pi}} \equiv \{d_m^{\widetilde{\pi}}\}_{m=1}^M,$$

where each episodic sample $d_m^{\widetilde{\pi}}$ consists of a set of 4-tuple elements as

$$d_m^{\widetilde{\pi}} \equiv \{(s_{m,n}^{\widetilde{\pi}}, a_{m,n}^{\widetilde{\pi}}, r_{m,n}^{\widetilde{\pi}}, s_{m,n+1}^{\widetilde{\pi}})\}_{n=1}^N.$$

We use two types of policies which have different purposes: the *sampling policy* $\widetilde{\pi}(a|s)$ for collecting data samples and the *current policy* $\pi(a|s)$ for computing the value function $\widehat{Q}^\pi$. If $\widetilde{\pi}(a|s)$ is equal to $\pi(a|s)$, just replacing the expectation contained in the error G by sample averages gives a *consistent* estimator (i.e., the estimated parameter converges to the optimal value as the number M of episodes goes to infinity):

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{\mathrm{NIW}} &\equiv \arg\min_{\boldsymbol{\theta}} \widehat{G}_{\mathrm{NIW}},\\ \widehat{G}_{\mathrm{NIW}} &\equiv \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\widehat{g}_{m,n},\\ \widehat{g}_{m,n} &\equiv \widehat{g}_{\mathrm{BR}}(s_{m,n}^{\widetilde{\pi}}, a_{m,n}^{\widetilde{\pi}}, r_{m,n}^{\widetilde{\pi}};\boldsymbol{\theta},\mathcal{D}),\end{aligned}$$

$$\widehat{g}_{\mathrm{BR}}(s,a,r;\boldsymbol{\theta},\mathcal{D})$$
$$\equiv\Big(\widehat{Q}^{\pi}(s,a;\boldsymbol{\theta})-r-\frac{\gamma}{|\mathcal{D}_{(s,a)}|}\sum_{s'\in\mathcal{D}_{(s,a)}}\mathbb{E}_{\pi(a'|s')}[\widehat{Q}^{\pi}(s',a';\boldsymbol{\theta})]\Big)^2,$$

where $\mathcal{D}_{(s,a)}$ is a set of 4-tuple elements containing state s and action a in the training data $\mathcal{D}$, and 'NIW' denotes 'No Importance Weight' (explained later). This situation is called *on-policy*.

However, $\widetilde{\pi}(a|s)$ is usually different from $\pi(a|s)$ since the current policy is updated in policy iteration. The situation where $\widetilde{\pi}(a|s)$ is different from $\pi(a|s)$ is called *off-policy*. In the off-policy setting, $\widehat{\boldsymbol{\theta}}_{\mathrm{NIW}}$ is no longer consistent. This inconsistency problem could be avoided by gathering new samples, i.e., when the current policy is updated, new samples are gathered following the updated policy and the new samples are used for policy evaluation. However, when the data sampling cost is high, this is not cost-efficient—it would be more cost-efficient if we could reuse the previously gathered samples.

In the following sections, we address the issue of efficient sample reuse in the off-policy setting.

Importance Weighting Techniques

In this section, we review existing off-policy reinforcement learning techniques.

Importance Sampling. *Importance sampling* is a general technique for dealing with the off-policy situation. Suppose we have i.i.d. (*independent and identically distributed*) samples $\{x_m\}_{m=1}^M$ from a strictly positive probability density function $\widetilde{P}(x)$. Using these samples, we would like to compute the expectation of a function $g(x)$ over another probability density function $P(x)$. A consistent approximation of the expectation is given by the *importance-weighted* average as follows:

$$\frac{1}{M}\sum_{m=1}^{M} g(x_m)\frac{P(x_m)}{\widetilde{P}(x_m)} \xrightarrow{M\to\infty} \mathbb{E}_{\widetilde{P}(x)}\left[g(x)\frac{P(x)}{\widetilde{P}(x)}\right]$$
$$=\int g(x)\frac{P(x)}{\widetilde{P}(x)}\widetilde{P}(x)dx=\int g(x)P(x)dx=\mathbb{E}_{P(x)}\left[g(x)\right].$$

However, applying the importance sampling technique in off-policy reinforcement learning is not that straightforward since our training samples of state s and action a are not i.i.d. due to the sequential nature of MDPs. Below, we review existing importance weighting techniques in MDPs.

Episodic Importance Weights. In a setting with episodic restarts, the independence of the episodes can be employed resulting into the episodic importance weight (EIW) (Sutton & Barto 1998). More specifically, the error G defined by (2) can be rewritten as

$$G=\mathbb{E}_{P_{\mathrm{I}},\widetilde{\pi},P_{\mathrm{T}}}\left[\frac{1}{N}\sum_{n=1}^{M} g(s_n,a_n;\boldsymbol{\theta})\frac{P_{\pi}(d)}{P_{\widetilde{\pi}}(d)}\right],$$

where $P_{\pi}(d)$ is the probability density of observing episodic data d under policy π:

$$P_{\pi}(d)\equiv P_{\mathrm{I}}(s_1)\prod_{n=1}^{N}\pi(a_n|s_n)P_{\mathrm{T}}(s_{n+1}|s_n,a_n).$$

We note that the importance weights can be computed without explicitly knowing P_{I} and P_{T} as

$$\frac{P_{\pi}(d)}{P_{\widetilde{\pi}}(d)}=\frac{\prod_{n=1}^{N}\pi(a_n|s_n)}{\prod_{n=1}^{N}\widetilde{\pi}(a_n|s_n)}.$$

Using the training data $\mathcal{D}^{\widetilde{\pi}}$, we can construct a consistent estimator of G as

$$\widehat{G}_{\mathrm{EIW}}\equiv\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\widehat{g}_{m,n}w_{m,N},$$

where

$$w_{m,N}\equiv\frac{\prod_{n'=1}^{N}\pi(a^{\widetilde{\pi}}_{m,n'}|s^{\widetilde{\pi}}_{m,n'})}{\prod_{n'=1}^{N}\widetilde{\pi}(a^{\widetilde{\pi}}_{m,n'}|s^{\widetilde{\pi}}_{m,n'})}.$$

Per-decision Importance Weights. The paper (Precup, Sutton, & Singh 2000) proposed a more efficient importance sampling technique called the *per-decision importance weight* (PDIW) method. A crucial observation in PDIW is that the error at the n-th step does not depend on the samples after the n-th step, i.e., G can be rewritten as

$$G=\mathbb{E}_{P_{\mathrm{I}},\widetilde{\pi},P_{\mathrm{T}}}\left[\frac{1}{N}\sum_{n=1}^{N} g(s_n,a_n;\boldsymbol{\theta})w_{m,n}\right].$$

Using the training data $\mathcal{D}^{\widetilde{\pi}}$, we can construct a consistent estimator as

$$\widehat{G}_{\mathrm{PDIW}}\equiv\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\widehat{g}_{m,n}w_{m,n}.$$

Based on this, the parameter $\boldsymbol{\theta}$ is estimated by

$$\widehat{\boldsymbol{\theta}}_{\mathrm{PDIW}}\equiv\arg\min_{\boldsymbol{\theta}}\widehat{G}_{\mathrm{PDIW}}.$$

Policy Iteration with Adaptive Importance Weights

The importance weighted estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{PDIW}}$ is guaranteed to be consistent. However, it is not *efficient* in the statistics sense (Shimodaira 2000), i.e., it can have large variance in finite sample cases and therefore learning with PDIW could be unstable in practice. In this section, we propose a new importance-weighting method that is more stable than existing methods.

Adaptive Per-Decision Importance Weights. In order to improve the estimation accuracy, it is important to control the trade-off between consistency and efficiency (or similarly bias and variance) based on the training data. Here, we introduce a *flattening parameter* ν $(0\leq\nu\leq 1)$ to control the trade-off by slightly 'flattening' the importance weights (Shimodaira 2000; Sugiyama, Krauledat, & Müller 2007):

$$\widehat{G}_{\mathrm{APDIW}}\equiv\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\widehat{g}_{m,n}\Big(w_{m,n}\Big)^{\nu}, \qquad (3)$$

where APDIW means Adaptive PDIW. Based on this, the parameter $\boldsymbol{\theta}$ is estimated by

$$\widehat{\boldsymbol{\theta}}_{\mathrm{APDIW}} \equiv \arg\min_{\boldsymbol{\theta}} \widehat{G}_{\mathrm{APDIW}}. \quad (4)$$

When $\nu = 0$, $\widehat{\boldsymbol{\theta}}_{\mathrm{APDIW}}$ is reduced to the ordinary estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{NIW}}$. Therefore, it has large bias but has relatively small variance. On the other hand, when $\nu = 1$, $\widehat{\boldsymbol{\theta}}_{\mathrm{APDIW}}$ is reduced to the importance-weighted estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{PDIW}}$. Therefore, it has small bias but has relatively large variance. Generally, the best ν tends to be large (small) when the number of training samples is large (small). However, this general trend is not enough to fine-tune the flattening parameter since the best value of ν would depend on training samples, sampling and current policies etc.

Automatic Selection of the Flattening Parameter. Here, we show how we determine the value of the flattening parameter ν automatically from the training data and policies. We estimate the approximation error G using *importance-weighted cross validation* (IWCV) (Sugiyama, Krauledat, & Müller 2007). A basic idea of IWCV is to divide the training data $\mathcal{D}^{\widetilde{\pi}}$ into a 'training part' and a 'validation part'. Then the parameter $\boldsymbol{\theta}$ is learned from the training part and the approximation error is estimated using the validation part. Below we explain in more detail how we apply IWCV to the selection of the flattening parameter ν in the current context.

Let us divide M episodic training data $\mathcal{D}^{\widetilde{\pi}}$ into K subsets $\{\mathcal{D}^{\widetilde{\pi}}_k\}_{k=1}^K$ of the same size (typically $K = 5$). For simplicity, we assume that M/K is an integer. Let $\widehat{\boldsymbol{\theta}}^k_{\mathrm{APDIW}}$ be the parameter learned from $\{\mathcal{D}^{\widetilde{\pi}}_{k'}\}_{k' \neq k}$ with APDIW (3). Then, the approximation error G is estimated by

$$\widehat{G}_{\mathrm{IWCV}} = \frac{1}{K} \sum_{k=1}^{K} \widehat{G}^k_{\mathrm{IWCV}}, \quad (5)$$

where

$$\widehat{G}^k_{\mathrm{IWCV}} = \frac{K}{MN} \times \sum_{d^{\widetilde{\pi}}_m \in \mathcal{D}^{\widetilde{\pi}}_k} \sum_{n=1}^{N} \widehat{g}_{\mathrm{BR}}(s^{\widetilde{\pi}}_{m,n}, a^{\widetilde{\pi}}_{m,n}, r^{\widetilde{\pi}}_{m,n}; \widehat{\boldsymbol{\theta}}^k_{\mathrm{APDIW}}, \mathcal{D}^{\widetilde{\pi}}_k) w_{m,n}. \quad (6)$$

We estimate the approximation error by the above K-fold IWCV method for all model candidates (in the current setting, several different values of the flattening parameter ν) and choose the best one that minimizes the estimated error:

$$\widehat{\nu}_{\mathrm{IWCV}} = \arg\min_{\nu} \widehat{G}_{\mathrm{IWCV}}. \quad (7)$$

One may think that, for model selection, $\widehat{G}_{\mathrm{APDIW}}$ could be directly used, instead of $\widehat{G}_{\mathrm{IWCV}}$. However, it can be proved that $\widehat{G}_{\mathrm{APDIW}}$ is heavily biased (or in other words, *over-fitted*) since the same training samples are used twice for learning the parameters and estimating the approximation error. On the other hand, we can prove that $\widehat{G}_{\mathrm{IWCV}}$ is an almost unbiased estimator of G, where 'almost' comes from the fact that the number of training samples is reduced due to data splitting in the cross validation procedure (Sugiyama, Krauledat, & Müller 2007). Note that ordinary CV is heavily biased due to the off-policy setting.

Sample-Reuse Policy Iteration (SRPI). Finally we show how the above methods can be applied to policy iteration. Let us denote the policy at the l-th iteration by π_l and the maximum number of iterations by L. In general policy-iteration methods, after the current policy is improved, new data samples $\mathcal{D}^{\pi_l}$ are collected following the new policy π_l to evaluate the policy π_l. Thus, previously-collected data samples $\{\mathcal{D}^{\pi_1}, \mathcal{D}^{\pi_2}, ..., \mathcal{D}^{\pi_{l-1}}\}$ are not used:

$$\pi_1 \overset{E:\{\mathcal{D}^{\pi_1}\}}{\longrightarrow} \widehat{Q}^{\pi_1} \overset{I}{\longrightarrow} \pi_2 \overset{E:\{\mathcal{D}^{\pi_2}\}}{\longrightarrow} \widehat{Q}^{\pi_2} \overset{I}{\longrightarrow} \cdots \overset{I}{\longrightarrow} \pi_L,$$

where E:$\{\mathcal{D}\}$ and I indicate policy evaluation using the data sample $\mathcal{D}$ and policy improvement respectively. It would be more efficient if we could reuse previously collected data samples to perform policy evaluation as:

$$\pi_1 \overset{E:\{\mathcal{D}^{\pi_1}\}}{\longrightarrow} \widehat{Q}^{\pi_1} \overset{I}{\longrightarrow} \pi_2 \overset{E:\{\mathcal{D}^{\pi_1},\mathcal{D}^{\pi_2}\}}{\longrightarrow} \widehat{Q}^{\pi_2} \overset{I}{\longrightarrow} \cdots \overset{I}{\longrightarrow} \pi_L.$$

However, reusing previously-collected data samples causes the *off-policy* situation because previous policies and the current policy are different unless the current policy is converged to the optimal one. Here we propose using APDIW and IWCV in policy iteration. To this purpose, we extend the definition of $\widehat{G}_{\mathrm{APDIW}}$ (3) so that multiple sampling policies $\{\pi_1, \pi_2, \ldots, \pi_l\}$ are taken into account:

$$\widehat{\boldsymbol{\theta}}^l_{\mathrm{APDIW}} \equiv \arg\min_{\boldsymbol{\theta}} \widehat{G}^l_{\mathrm{APDIW}}, \quad (8)$$

$$\widehat{G}^l_{\mathrm{APDIW}} \equiv \frac{1}{lMN} \sum_{l'=1}^{l} \sum_{m=1}^{M} \sum_{n=1}^{N} \widehat{g}_{\mathrm{BR}}(s^{\pi_{l'}}_{m,n}, a^{\pi_{l'}}_{m,n}, r^{\pi_{l'}}_{m,n}; \boldsymbol{\theta}, \{\mathcal{D}^{\pi_{l'}}\}_{l'=1}^{l}) \left(\frac{\prod_{n'=1}^{n} \pi_l(a^{\pi_{l'}}_{m,n'} | s^{\pi_{l'}}_{m,n'})}{\prod_{n'=1}^{n} \pi_{l'}(a^{\pi_{l'}}_{m,n'} | s^{\pi_{l'}}_{m,n'})} \right)^{\nu_l},$$

where $\widehat{G}^l_{\mathrm{APDIW}}$ is the approximation error estimated at the l-th policy evaluation with APDIW. The flattening parameter ν_l is chosen based on IWCV before performing policy evaluation.

Experiments

In this section, we evaluate the performance of our proposed method.

Chain walk. First, we illustrate how the flattening parameter ν influences the estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{APDIW}}$. The MDP (see Fig.1) consists of 10 states $\mathcal{S} = \{s^{(i)}\}_{i=1}^{10}$ and two actions $\mathcal{A} = \{L, R\}$. The reward $+1$ is given when visiting $s^{(1)}$ and $s^{(10)}$. The transition probability P_{T} is indicated by the numbers attached to the arrows in the figure. The Gaussian kernel with standard deviation $\sigma = 10$ is used as a basis function and kernel centers are located at $s^{(1)}$, $s^{(5)}$ and $s^{(10)}$. We note that the total number of basis functions is 6 since each kernel is copied over the action space. The discount factor is set at 0.9. $\widehat{\boldsymbol{\theta}}_{\mathrm{APDIW}}$ is computed from the training samples $\mathcal{D}_{\widetilde{\pi}}$ using (4) with several different numbers M of episodes; the number N of steps is fixed at 10. The sampling policy $\widetilde{\pi}(a|s)$ and the target policy $\pi(a|s)$ are chosen randomly at every trial such that $\widetilde{\pi} \neq \pi$. Fig.2 depicts the

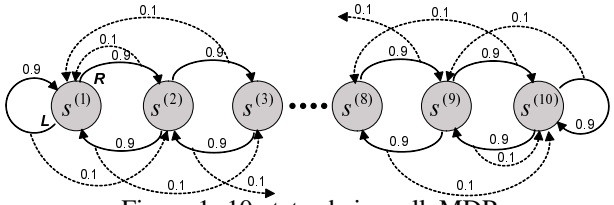
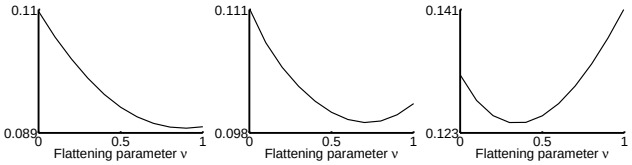
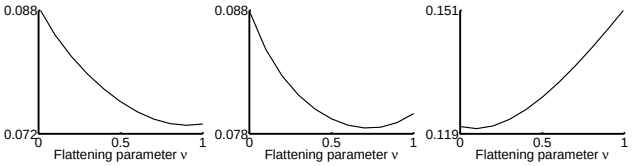


Figure 1: 10-state chain walk MDP.



(a) 200 episodes (b) 50 episodes (c) 10 episodes

Figure 2: Average true error G over 50 trials as a function of the flattening parameter ν in the 10-state chain walk problem. The trend of G differs depending on the number of episodes.



(a) 200 episodes (b) 50 episodes (c) 10 episodes

Figure 3: Average error estimated by 5-fold IWCV ($\widehat{G}_{\mathrm{APDIW}}$) as a function of the flattening parameter ν in the 10-state chain walk problem. IWCV nicely captures the trend of the true error G in Fig.2

true error G averaged over 50 trials as functions of the flattening parameter ν. Note that the true error G is used only for illustration purposes and is not available in reality.

Fig.2(a) shows that when the number of episodes is large ($M = 200$), the true error G tends to decrease as the flattening parameter increases. This would be a natural result due to consistency of $\widehat{\boldsymbol{\theta}}_{\mathrm{APDIW}}$ when $\nu = 1$. On the other hand, Fig.2(b) shows that when the number of episodes is not large ($M = 50$), $\nu = 1$ performs rather poorly. This implies that the consistent estimator tends to be unstable when the number of episodes is not large enough; $\nu = 0.7$ works the best in this case. Fig.2(c) shows the results when the number of episodes is further reduced ($M = 10$). This illustrates that the consistent estimator with $\nu = 1$ is even worse than the ordinary estimator ($\nu = 0$) because the bias is dominated by large variance. In this case, the best ν is even smaller and is achieved at $\nu = 0.3$.

Next, we illustrate how IWCV works. Fig.3 depicts the error estimated by 5-fold IWCV averaged over 50 trials as a function of the flattening parameter ν. The figures show that IWCV nicely captures the trend of the true error G for all three cases (cf Fig.2). Fig.4 describes, as functions of the number M of episodes, the average true error G obtained by NIW ($\nu = 0$), PDIW ($\nu = 1$), and APDIW+IWCV ($\nu \in \{0.0, 0.1, 0.2, ..., 0.9, 1.0\}$ is adaptively selected in each trial using IWCV). This result shows that the improvement of the performance by NIW saturates when $M \geq 30$, implying that the bias caused by NIW is not negligible.

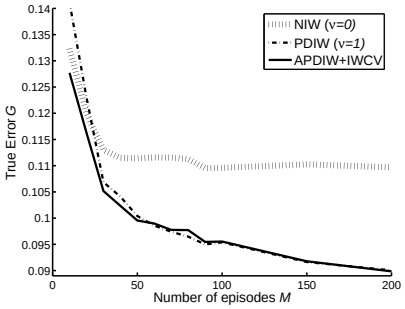


Figure 4: Average true error G over 50 trials in the 10-state chainwalk MDP.

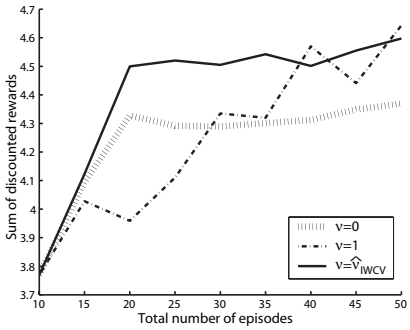


Figure 5: The performance of policies measured by the average sum of discounted rewards over 30 trials as a function of the total number of episodes in the 10-state chain walk problem.

The performance of PDIW is worse than the baseline NIW method when $M \leq 30$ which is caused by the large variance of PDIW. On the other hand, APDIW+IWCV consistency gives the best performance for all M, illustrating the high adaptation ability of the proposed method.

Finally, we illustrate how SRPI works. We consider three scenarios: ν is fixed at 0, ν is fixed at 1 and SRPI (ν is chosen by IWCV). The agent collects samples $\mathcal{D}^{\pi_l}$ ($M = 5$ and $N = 10$) at every policy iteration following the current policy π_l and computes $\widehat{\boldsymbol{\theta}}^{l}_{\mathrm{APDIW}}$ from all collected samples $\{\mathcal{D}^{\pi_1}, \mathcal{D}^{\pi_2}, \ldots, \mathcal{D}^{\pi_l}\}$ by (8). We use 3 Gaussian kernels with standard deviation $\sigma = 10$ as basis functions and place the center of kernels on a randomly selected state ($\in \mathcal{S}$). Fig.5 depicts the average sum of discounted rewards computed from the test samples. The initial policy π_1 is chosen randomly. Policy improvement is carried out by softmax (1) with $\tau = 2l$. Fig.5 shows that SRPI can outperform the cases with $\nu = 0$ and $\nu = 1$. This indicates that SRPI effectively reuse previously collected samples throughout the iterations by choosing the flattening parameter appropriately.

Mountain Car. We consider the *mountain car task* (Sutton & Barto 1998) illustrated in Fig.6, consisting of a car and landscape described by $\sin(3x)$. The goal of the task is to guide the car to the goal. The action space $\mathcal{A}$ consists of three different values of the force $\{0.2, -0.2, 0\}[kg \cdot m/s^2]$. We note that the force itself is not strong enough for the car to climb the right hill up. The state space $\mathcal{S}$ is continuous and consists of the horizontal position $x[m]$ ($\in [-1.2, 0.5]$)

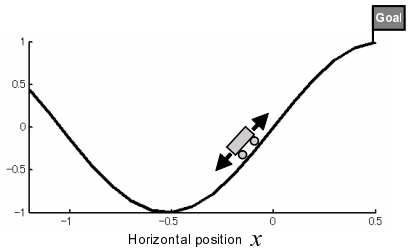


Figure 6: Illustration of the mountain car task.

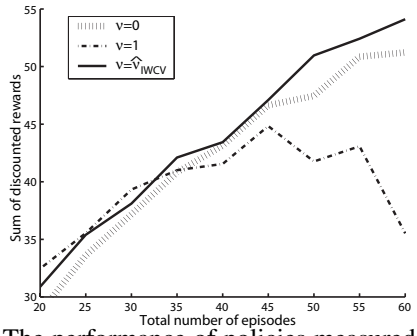


Figure 7: The performance of policies measured by the average sum of discounted rewards over 30 trials as a function of the total number of episodes in the mountain car task.

and the velocity $\dot{x}[m/s]$ ($\in [-1.5, 1.5]$). The equation of motion of the car is

$$\ddot{x} = \Big(-9.8w\cos(3x) + \frac{a}{w} - k\dot{x} \Big), \qquad (9)$$

where a is an action ($\in \mathcal{A}$), w ($= 0.2[kg]$) is the mass of the car, k ($= 0.3$) is the friction coefficient. We set the time step at $\Delta t = 0.1[s]$. We define the reward function $R(s, a, s')$ as

$$R(s, a, s') = \begin{cases} 1 & \text{if } x_{s'} \geq 0.5, \\ -0.01 & \text{otherwise.} \end{cases}$$

We use 10 Gaussian kernels with standard deviation $\sigma = 1$ as basis functions and place kernel centers randomly on the state space ($\in \mathcal{S}$). The initial policy $\pi_1(a|s)$ and the initial-state probability density $P_1(s)$ are set to be uniform. The agent collects data samples $\mathcal{D}^{\pi_l}$ ($M = 5$ and $N = 50$) following the current policy π_l. The discounted factor γ is set at 0.9 and the policy is improved by softmax (1) with $\tau = l$. Fig.7 describes the performance of learned policies measured by the average sum of discounted rewards using independent test samples. The graph shows that SRPI works very well; the performance of the policy learned with $\nu = 1$ improves quickly in the beginning but saturates in the middle. $\nu = 0$ performs relatively well, but progress of learning tends to be behind SRPI. Thus the proposed method SRPI is shown to efficiently reuse the previously-collected samples.

Conclusions and Outlook

Instability has been one of the critical limitations of importance sampling techniques, which often makes off-policy methods impractical. To overcome this weakness, we introduced an *adaptive* importance sampling technique for controlling the trade-off between consistency and efficiency in value function approximation. We further provided an automatic model selection method for actively choosing the flattening parameter. We also proposed using the adaptive importance sampling technique in policy iteration for efficiently reusing previously collected data samples. The experimental results showed that the proposed method compares favorably with existing approaches.

The method presented in this paper can easily be extended to policy gradient methods where the need for reusing past experience is even more urgent as small gradient-descent steps result into an underutilization of the data. While importance sampling has been applied in the settings of policy gradient methods (Shelton 2001; Peshkin 2002), policy gradient methods tend to be unstable when used with standard importance-sampling methods (Kakade 2002)—the proposed methods would offer an alternative.

Acknowledgments

The authors acknowledge financial support from JSPS Global COE program "Computationism as a Foundation for the Sciences", MEXT (17700142 and 18300057), the Okawa Foundation, the Microsoft CORE3 Project and the IBM Faculty Award.

References

Bagnell, J. A.; Kakade, S.; Ng, A. Y.; and Schneider, J. 2003. Policy search by dynamic programming. In *NIPS 16*.

Fishman, G. S. 1996. *Monte Carlo: Concepts, Algorithms, and Applications*. Berlin: Springer-Verlag.

Kakade, S. 2002. A natural policy gradient. In *NIPS 14*.

Lagoudakis, M. G., and Parr, R. 2003. Least-squares policy iteration. *JMLR* 4:1107–1149.

Peshkin, L. Shelton, C. 2002. Learning from scarce experience. In *Proc. of ICML*.

Precup, D.; Sutton, R. S.; and Dasgupta, S. 2001. Off-policy temporal-difference learning with function approximation. In *Proc. of ICML*.

Precup, D.; Sutton, R. S.; and Singh, S. 2000. Eligibility traces for off-policy policy evaluation. In *Proc. of ICML*.

Schoknecht, R. 2003. Optimality of reinforcement learning algorithms with linear function approximation. In *NIPS 15*.

Shelton, C. R. 2001. Policy improvement for pomdps using normalized importance sampling. In *Proc. of UAI*.

Shimodaira, H. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90(2):227–244.

Sugiyama, M.; Krauledat, M.; and Müller, K.-R. 2007. Covariate shift adaptation by importance weighted cross validation. *JMLR* 8:985–1005.

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

Watkins, C. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation, King's College, University of Oxford.