# DualDICE: Behavior-Agnostic Estimation of Discounted Stationary Distribution Corrections

**Michael Benitah**
260187637
michael.benitah@umontreal.ca

**Marc-André Ruel**
260962515
ruel.marcandre@gmail.com

**Louis-François Preville-Ratelle**
110227536
louis-francois.preville-ratelle@umontreal.ca

## 1 Introduction

In this project, we do an overview of the article Nachum et al. [2019], clarify some of the mathematical development of optimization problem using Fenchel conjugates and apply the DualDICE algorithm to an infinite-horizon version of the tabular Taxi environment (so that we can compare our results with CartPole and Reacher environments presented in the article).[1]

Policy learning depends on data, but it is often difficult to gather such data because in many situations one does not have direct access to the environment. By recurring to off-policy data, one can apply importance sampling to estimate a per step reward, but this technique requires knowledge of the state-action probabilities for both the target and behaviour policies and may present high variance if the behaviour policy is too far off from the target policy.

Nachum et al. [2019] propose an algorithm that can be applied to any given data set of trajectories, with no requirement of any specific behaviour policy. The trajectories may be obtained by one or many behaviour policies or at random by any possible mean. DualDICE is generally more robust if compared to other algorithms used for estimating stationary distribution ratios precisely because it does not use importance ratios and is, thus, less prone to variance which may affect the stability of such algorithms.

## 2 Description of the algorithm and mathematical details

### 2.1 Background

The article uses convex analysis and also uses the Fenchel conjugates and the Fenchel Duality which are both central to their approach. The latter is also known as the Fenchel–Rockafellar duality. Since we had shallow knowledge about convex analysis, we looked for references with the goal of gaining an intuitive understanding of Fenchel duality. A well-known reference is the book of Rockafellar (see Rockafellar [1970]), but it takes a lot of time to get to the Fenchel duality. With the time constraints, we decided to use a more recent book Mordukhovich and Nam [2013], which we read most of the first half. The current version does not include Fenchel duality, but some videos on Youtube by one of the authors do (Nguyen [2020]). According to the authoer, an extended version of this book will appear soon that includes more material like the Fenchel duality.

[1] Our GitHub repository: https://github.com/marued/RL-dualDICE

Preprint. Under review.

This report follows quite closely the main article and its notation. Some additional comments are presented when more explanations are helpful and adding a bit more mathematical background when it seems appropriate.

First, let's start by defining the notation. Fallowing the same notation used in class, let $\mathcal{M} = \langle S, A, R, T, \beta, \gamma \rangle$, where $S$ is a state space, $A$ an action space, $R$ a reward function, $T$ a transition probability function, $\beta$ an initial state distribution and a discount factor $\gamma \in [0, 1)$. The symbols $\pi$ and $\mu$ represent some policies. Similarly, $\mathcal{D}$ represents a dataset that contains transitions of the form $(s, a, r, s')$ made from trajectories from potentially multiple policies or more complicated recipes. However, it seems important to mention that these come from whole trajectories and not only tuples. An infinite horizon environment is assumed, although some finite horizon can be seen as infinite by adding at the end a new terminal state which loops onto itself.

The main variable of interest is the normalized expected per step reward:

$$\rho(\pi) := (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 \sim \beta, \forall t, a_t \sim \pi(s_t), r_t \sim R(s_t, a_t), s_{t+1} \sim T(s_t, a_t)\right]. \tag{1}$$

If the data is obtained following $\mu$ with trajectories of the form $\tau = (s_0, a_0, r_0, \ldots, s_H)$, then diverse methods using the importance sampling ratios $\prod_{t=0}^{H-1} \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}$ can be used to give an expected formula for $\rho(\pi)$, where the data is collected following $\mu$. These methods that use these ratios can have exponential variance, which can be a big issue (see Sutton and Barto [2018] for more about this subject). A similar formula for $\rho(\pi)$ that avoids these long-term ratios is given by:

$$\rho(\pi) = \mathbb{E}_{(s,a)\sim d^\pi, r\sim R(s,a)}[r] \tag{2}$$

where

$$d^\pi(s,a) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \text{Pr}(s_t = s, a_t = t | s_0 \sim \beta, \forall t, a_t \sim \pi(s_t), r_t \sim R(s_t, a_t), s_{t+1} \sim T(s_t, a_t)) \tag{3}$$

is the normalized discounted stationary distribution. Similar to the importance sampling framework, if the data comes from the dataset $\mathcal{D}$, then

$$\rho(\pi) = \mathbb{E}_{(s,a,r)\sim d^\mathcal{D}}[w_{\pi/\mathcal{D}}(s,a) r], \tag{4}$$

where $d^\mathcal{D}$ is similarly defined as $d^\pi$ and $w_{\pi/\mathcal{D}}(s,a) := \frac{d^{\pi(s,a)}}{d^{\mathcal{D}(s,a)}}$ are the main coefficients that the presented method tries to estimate. The authors only assume that $d^{\pi(s,a)} > 0$ implies $d^\mathcal{D} > 0$ and that $||w_{\pi/\mathcal{D}}(s,a)||_\infty < C$ for some constant $C$.

### 2.2 DualDICE Algorithm

Just as the minimum of the function $\frac{1}{2}mx^2 - nx$ is attained at $x^* = \frac{n}{m}$, the optimizer of the following problem:

$$\min_{v: S\times A \to \mathbb{R}} \frac{1}{2}\mathbb{E}_{(s,a)\sim d^\mathcal{D}}\left[x(s,a)^2\right] - \mathbb{E}_{(s,a)\sim d^\pi}\left[x(s,a)\right] \tag{5}$$

is given by

$$x^*(s,a) = \frac{d^{\pi(s,a)}}{d^{\mathcal{D}(s,a)}} = w_{\pi/\mathcal{D}}(s,a). \tag{6}$$

In the previous equation, access to the data points $(s,a) \in d^\mathcal{D}$ is given, but not access to the whole distribution $d^\pi$. Only the initial distribution $\beta$ is accessible. To fix this issue, they perform a change of variables:

$$\nu(s,a) = x(s,a) + \gamma \mathbb{E}_{s'\sim T(s,a), a'\sim\pi(s')}\left[\nu(s',a')\right], \forall (s,a) \in S \times A. \tag{7}$$

Using this, the authors calculate some simple cancellations and prove that Equation (5) is equivalent to

$$\min_{\nu: S\times A \to \mathbb{R}} \frac{1}{2}\mathbb{E}_{(s,a)\sim d^\mathcal{D}}\left[(\nu - \mathcal{B}^\pi \nu)(s,a)^2\right] - (1-\gamma)\mathbb{E}_{s_0\sim\beta, a_0\sim\pi(s_0)}\left[\nu(s_0,a_0)\right], \tag{8}$$

where $\mathcal{B}^\pi$ is the expected Bellman operator with zero reward given by
$\mathcal{B}^\pi \nu(s,a) = \gamma \mathbb{E}_{s'\sim T(s,a), a'\sim\pi(s')}\left[\nu(s',a')\right]$. The change of variables was probably guessed with the

goal of dropping $d^\pi$ from the equation in favor of the initial state distribution $\beta$. But as it will be explained later, there is a lot more behind this change of variables. Please note from above that

$$(\nu^* - \mathcal{B}^\pi)(s,a) = \nu^*(s,a) = w_{\pi/\mathcal{D}}(s,a), \tag{9}$$

where the RHS ($\forall s, a \in S \times A$) are precisely the coefficients that their algorithms will try to estimate. Equation (8) might seem like a good function to optimize, unfortunately as the authors mentioned, the quantity $(\nu - \mathcal{B}^\pi\nu)(s,a)^2$ contains a conditional expression inside a square, and getting a good gradient in this case is apparently difficult. We know very little about these optimization issues, but we accepted their comments that an environment that is stochastic with a large or continuous action space might be hard to optimize with standard stochastic techniques (this topic seems well known, see Mohamed et al. [2019] for more).

We can dodge from those two difficulties by applying Fenchel conjugates. The Fenchel conjugate of a function $f(x)$ is defined as $f^*(\zeta) = \max_x \zeta x - f^*(x)$. It is a theorem that when $f(x)$ is convex, then $f^{**}(x) = \max_\zeta \; x\zeta - f^*(\zeta) = f(x)$. For $f(x) = \frac{1}{2}x^2$, it is easy to derive that $f^*(\zeta) = \frac{1}{2}\zeta^2$. Therefore Equation (8) is equivalent to

$$\min_{v:S\times A\to\mathbb{R}} \mathbb{E}_{(s,a)\sim d^\mathcal{D}}\left[\max_\zeta \left(v - \mathcal{B}^\pi\nu\right)(s,a)\zeta - \frac{1}{2}\zeta^2\right] - \left[(1-\gamma)\mathbb{E}_{s_0\sim\beta, a_0\sim\pi(s_0)}\left[\nu(s_0,a_0)\right]\right], \tag{10}$$

which in turn is equivalent to

$$\min_{v:S\times A\to\mathbb{R}} \max_{\zeta:S\times A\to\mathbb{R}} \Bigg( \mathbb{E}_{(s,a,s')\sim d^\mathcal{D}, a'\sim\pi(s')}\left[(\nu(s,a) - \gamma\nu(s',a'))\zeta(s,a) - \frac{1}{2}\zeta(s,a)^2\right] - (1-\gamma)\mathbb{E}_{s_0\sim\beta,a_0\sim\pi(s_0)}\left[\nu(s_0,a_0)\right] \Bigg). \tag{11}$$

Skipping some details, the authors mention that using KKT conditions and other results of optimization, the saddle point solution $(\nu^*, \zeta^*)$ of the previous minimax optimization problem is equal to $\zeta^*(s,a) = (\nu^* - \mathcal{B}^\pi\nu^*)(s,a) = w_{\pi/\mathcal{D}}(s,a)$.

By the form of the objective function and since we have direct access to $\zeta(s,a)$ (no conditional inside a square anymore), they conclude that Equation (11) is easy to optimize. At the end of their article, they prove with this optimization output $(\hat{\nu}, \hat{\zeta})$ over several pages that $\mathbb{E}_{d^\mathcal{D}}[\hat{\zeta}(s,a)r]$ approximates well $\rho(\pi)$ under some required conditions. We will not discuss these proofs in this report. Nor will we include the pseudocode of the DualDICE algorithm, which is presented on page 12 of their article. Intuitively, the algorithm is simple. It uses stochastic gradient descent directly on the objective function in Equation (11): it alternates between minimizing it using $\nu_{\theta_1}(s,a)$ and maximizing it using $\zeta_{\theta_2}(s,a)$. They also include a section that proves that the algorithm is applicable for other convex functions than $f(x) = \frac{1}{2}x^2$.

It is not hard to accept intuitively that this new objective function is more stable. The main thing missing in this article are some words about an important tool mentioned. The authors write "Exploiting Fenchel Duality" as a subtitle for a section just before using Fenchel conjugates. Unfortunately, the Fenchel duality does not explicitly appear anywhere in the article. We did not clearly understand why until we read a very recent review paper by two of the authors (see Nachum and Dai [2020]). In the last two paragraphs on page 14, the authors claim that the change of variables trick is essentially equivalent to applying the Fenchel duality. Without these explanations, we would never have found this. We believe this should have been included since it constitutes one of the most important contributions in the DualDICE paper. On a positive note, although we read only about half of that recent review paper (Nachum and Dai [2020]), it is really well written and it covers many interesting mathematical foundations related to similar optimization techniques. In particular, the reader learns that these techniques introduced in DualDICE came after similar methods in RL were developed around LP duality and Lagrange duality (Fenchel duality generalizes these two). This helps to put their work in perspective and better understand the origin, the foundations and the applications of these methods.

## 3 Experimental results

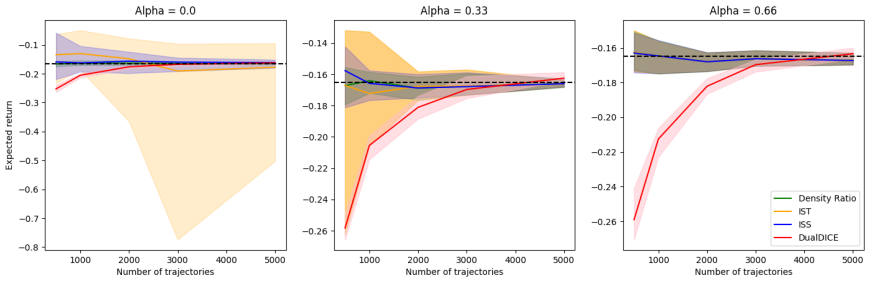


Figure 1: The behaviour and target policies are similar.

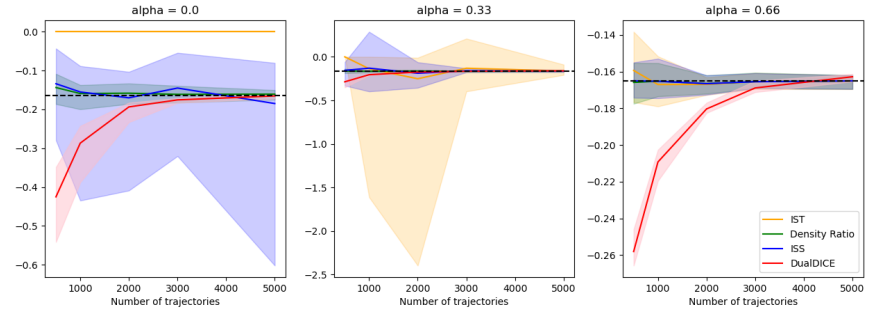


Figure 2: Using behaviour policy that is very different then the target policy .

We executed DualDICE on a modified infinite-horizon version of the Taxi environment. The trajectory does not end when a customer is dropped at a proper location. Instead, the episode continues indefinitely with new customers randomly appearing on the map. The code and experiments come from Liu et al. [2018] and we integrated DualDICE to it. We estimated the target expected return value, represented by the black dashed line, by running the target policy on 100 000 trajectories. In all our experiments, every trajectory ends at 400 steps. The different policies are obtained by running Q-Learning on the environment and periodically saving the policy during its learning process. We then use a thoroughly trained policy as our target policy and a policy with fewer training steps as an arbitrary policy $pi_+$. The behaviour policy is defined as a combination of the policies according to some mixture coefficient $\alpha$.

Figure 1 shows our results for the Taxi environment when the policy $pi_+$ is quite similar to the target policy for three different values of $\alpha$. Those results are very similar to the charts shown in Figure 2 of Nachum et al. [2019] for the environments CartPole and Reacher. Since DualDICE is supposed to behave well on any dataset, irrespective of the behaviour policy, we decided to execute the same test but with a very different behaviour policy. As expected, we see in Figure 2 that a behaviour policy which is very different from the target policy generates a high variance for importance sampling methods such as ISS or IST, even causing IST to collapse. DualDICE, however, at least in this particular setting, stays robust and maintains its typical behaviour with a small variance throughout the experiments, completely under control.

## 4 Conclusion

The downside for this infinite-horizon version of the Taxi environment is that DualDICE is more data hungry than the other methods and this seems to be the case for other environments as well. In cases where IST and ISS are well-behaved, it takes up to 2000 or 3000 trajectories before DualDICE is able to get similar results in the infinite-horizon Taxi environment. In spite of that, DualDICE seem to be

more robust in our experiments and does not require a precise behaviour policy to be able to evaluate the target policy, simply needing a dataset of given trajectories. To paint any specific behavioural outlines this algorithm has in any practical setting, we would need to test it in different environments.

## Statement of contribution

We discussed Nachum et al. [2019] ideas amongst us to get a thorough understanding of the paper. The code for this project is largely based on the code provided by `https://github.com/google-research/google-research/tree/master/dual_dice` and `https://github.com/zt95/infinite-horizon-off-policy-estimation`. Marc-André Ruel integrated the two pieces of code and implemented charts for the Taxi environment. Louis-François Preville-Ratelle worked on the mathematical aspect of the Fenchel duality. Michael Benitah made the bridge between code and theory and worked on the video presentation. The report was co-authored by all three of us.

## External References

Our Github repository can be found at `https://github.com/marued/RL-dualDICE`. The video presentation can be found at `https://youtu.be/no-JKqfD0zw`.

## Reproducibility Checklist

For all models and algorithms presented, check if you include:

- **A clear description of the mathematical setting, algorithm, and/or model.** See Section 2 and especially Equation (11) for the main objective function.
- **A clear explanation of any assumptions.** See the end of Section 2.1.

For any theoretical claim, check if you include:

- **A clear statement of the claim.** See Section 2 and Section 3.4 of the original paper Nachum et al. [2019] for the statement of convergence of the algorithm.
- **A complete proof of the claim.** See the Appendix D of the original paper Nachum et al. [2019] for detailed proofs.

For all datasets used, check if you include:

- **The relevant statistics, such as number of examples.** Each trajectories are 400 steps generated by a saved behavior policy available in the Github. The number of trajectories is part of the X axis of our results.
- **The details of train/validation/test splits.** We are testing on data generated from behaviour policy on different seeds.
- **An explanation of any data that were excluded, and all pre-processing step.** We have a saved behavior policy to generate the data needed for the experiments.
- **A link to a downloadable version of the dataset or simulation environment.** https://github.com/marued/RL-dualDICE
- **For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control.** Data collection is part of the experiments provided in the GitHub.

For all shared code related to this work, check if you include:

- **Specification of dependencies.** A requirement.txt is provided for python dependencies.
- **Training code.** Training code is part of infinite-horizon-off-policy-estimation in a method called Q_learning. Refer to Liu et al. [2018]
- **Evaluation code.** Part of run_graphs_compare_both.py
- **Pre-trained model(s).** The pretrained policies are part of the repository and are the same ones part of Liu et al. [2018]
- **README file includes table of results accompanied by precise command to run to produce those results.** Command are part of the README file.

For all reported experimental results, check if you include:

- **The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results.** These are all provided with the code if some are missing in our document.
- **The exact number of training and evaluation runs.** All experiments were run on 12 different seeds.
- **A description of results with central tendency (e.g. mean) and variation (e.g. error bars).** Part of this document.
- **The average runtime for each result, or estimated energy cost.** Depends on the type of machine since our experiments are run in multi-process (one for each seed). Can take up to 24 hours to run all experiments.
- **A description of the computing infrastructure used.** AMD Ryzen 6 core processor with 16 GB of ram. No need for graphics card.

## References


Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *arXiv preprint arXiv:1810.12429*, 2018.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning, 2019.

Boris S Mordukhovich and Nguyen Mau Nam. An easy path to convex analysis and applications. *Synthesis Lectures on Mathematics and Statistics*, 6(2):1–218, 2013.

Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality, 2020.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections, 2019.

Mau Nam Nguyen. Convex Analysis and Optimization. `https://sites.google.com/pdx.edu/convex-analysis-optimization`, 2020.

R Tyrrell Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1970.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. 2018.