Scale Invariant Solutions for Overdetermined Linear Systems with Applications to Reinforcement Learning

Rahul Madhavan 1  Gugan Thoppe 1  Hemanta Makwana 1

Abstract
Overdetermined linear systems are common in reinforcement learning, e.g., in Q and value function estimation with function approximation. The standard least-squares criterion, however, leads to a solution that is unduly influenced by rows with large norms. This is a serious issue, especially when the matrices in these systems are beyond user control. To address this, we propose a scale-invariant criterion that we then use to develop two novel algorithms for value function estimation: Normalized Monte Carlo and Normalized TD(0). Separately, we also introduce a novel adaptive stepsize that may be useful beyond this work as well. We use simulations and theoretical guarantees to demonstrate the efficacy of our ideas.

1. Introduction
Feature scaling and data normalization is a common practice in machine learning and has been shown to be effective in areas as widely disparate as deep learning (Bishop, 1995; Sola & Sevilla, 1997), nearest neighbour classifiers (Li et al., 2016; Singh & Singh, 2020), SVMs (Stolcke et al., 2008), PCA (Casella & Berger, 2001) and data mining (Han et al., 2011). Their main utility is when the norm of the input vector is not a true reflection of its importance (Bishop, 1995). Normalization is also known to often help increase the speed of learning (Ba et al., 2016) as well as reduce the dependence on outliers (Ben-Gal, 2005; Botchkarev, 2019). In this paper, we explore the idea of normalization in the context of solving an overdetermined system \( \Phi w = V \), where \( V \in \mathbb{R}^n \) and \( \Phi \in \mathbb{R}^{m \times n} \) with \( m \geq n \), and then study its applications to Reinforcement Learning (RL).

Typically, an overdetermined system \( \Phi w = V \) is solved using the least squares criterion: 
\[
\min_w \sum_{i=1}^m d_i (\phi_i^T w - V_i)^2 ,
\]
where \( \phi_i^T \) and \( V_i \) is the \( i \)-th rows of \( \Phi \) and \( V \), respectively, and \( d_i \)'s are some positive weights. In this work, we propose the alternative criterion
\[
\min_w \sum_{i=1}^m d_i (\phi_i^T w - V_i)^2 / \| \phi_i \|_2^2
\]
which is the weighted sum of squares of distances to the hyperplanes \( \phi_i^T w = V_i \), \( i = \{1, \ldots, m\} \).

We claim that our criterion has the following two advantages over least squares. Firstly, the solution under our criterion is scale invariant, i.e., it remains unchanged even if the individual equations are rescaled arbitrarily. Secondly, when \( \Phi \) has outlier rows, i.e., those with large norms, the solution obtained with our criterion may be more desirable than the one obtained via least squares (see the example below).

In the context of RL, we use the above idea to develop two novel algorithms for the value function estimation problem with linear function approximation: Normalized Monte Carlo and Normalized TD(0). These methods combine ideas of both the randomized (Strohmer & Vershynin, 2009a) as well as the stochastic Kaczmarz algorithm (Thoppe et al., 2014). In each iteration, the weight update is based on a subset of hyperplanes randomly sampled using a distribution that is proportional to the \( d_i \)'s. Further, our method allows for a noisy estimate of \( V_i \) in each hyperplane. Note that unlike (Strohmer & Vershynin, 2009a), the sampling distribution for the hyperplanes does not depend on \( \{ \| \phi_i \|_2 \} \). The main reasons for this being: i.) there may not be any additional benefits (Strohmer & Vershynin, 2009b), and ii.) the feature norms of all states may not be accessible ab-initio.

We now provide an example in the context of RL to illustrate an advantage of our criterion over the least squares idea in the presence of outliers.

Illustrating example: Consider the value function estimation problem in an \( m \) state Markov Decision Process (MDP). Under linear function approximation, this is equivalent to solving \( \Phi w = V \), where \( \Phi \) represents the feature matrix and
$V$ denotes the value function related to the given policy. Let $\gamma$ be the discount factor and let the reward on each transition be $r$. Suppose that under the given policy, the probability of transitioning between any two states is the same. Then, clearly, the value of each state is $\frac{r}{1-\gamma}$. Separately, let $\Phi$ be a column vector in $\mathbb{R}^m$ whose first $(m-1)$ values come from the normal distribution $N(\mu, \sigma)$ while the last one, arbitrarily set to $p\mu$, plays the role of an outlier.

We set $m = 20, \mu = 1, \sigma = 0.05, p = 5, r = 1, \gamma = 0.5$ and sample $\Phi$ 1000 times. Table 1 lists the mean $|\phi_i w - V_i|$ value for $w$ obtained under least squares and our proposed criteria. Clearly, the outlier has a significant influence in the least squares case (see Appendix A for further details).

1.1. Our Contributions

Our work makes the following five major contributions:

1. **Total Projections:** We introduce a novel stochastic Kaczmarz method called the Total Projections Algorithm (Section 2). This is useful for solving the overdetermined system $\Phi w = V$ under our proposed error criterion, and works even with access only to noisy samples of $V$.

2. **Scale Invariant RL methods:** Using the total projections idea, we develop two novel algorithms for value function estimation under linear function approximation: Normalized Monte-Carlo (Section 3) and Normalized TD(0) (Section 4). These methods are scale invariant, i.e., rows with large norms have minimal additional influence over the solution.

3. **Curvature based Adaptive Step Size:** We propose a novel adaptive step size (in Section 5.1) choice based on the radius of a suitably defined osculating circle, which we obtain from the Frenet-Serret equations. We show our update, in expectation, to be a contraction on the error function, and provide evidence (Figure 1) for speedup.

4. **Theoretical Convergence:** We use the theory of stochastic approximation to establish almost sure (a.s.) convergence for the two RL methods we propose. The proofs on the normalized TD(0) method involves some new tensor multiplication methods (appendix I, section 5.3).

5. **Momentum - simulations and theory:** We test various types of momentum with our algorithm using simulations and find constant-$\beta$ heavy-ball to be best. We incorporate this in our algorithm (section 5.2) and prove convergence using the theory of stochastic approximation (appendix G).

2. **Total Projections**

Consider the overdetermined linear system $\Phi w = V$, consisting of $m$ rows of the form $\phi_i^\top w = V_i$, $\phi_i, w \in \mathbb{R}^n$. Let $D$ be a diagonal weight matrix with entries $d_1, \ldots, d_m$. If we wish to solve $\min_w \sum_{i=1}^m d_i (\phi_i^\top w - V_i)^2 / \|\phi_i\|^2$, then the stochastic update (with say $\tau$ samples) takes the form

$$w_{k+1} = w_k - \alpha_k \frac{1}{\tau} \sum_{i=1}^\tau \phi_i^\top \frac{w_k - V_i}{\|\phi_i\|^2} \phi_i$$

where the rows are sampled with probability $d_i$. This update rule leads to Algorithm 1. Since each expression of the form $(\phi_i^\top w_k - V_i)\phi_i / (\|\phi_i\|^2)$ is a projection from $w_k$ onto the hyperplane $\phi_i^\top w = V_i$, we call the algorithm as Total Projections Algorithm.

For our full update rule, we need to add a momentum and our choice of step size. For the momentum part we use heavyball momentum with constant $\beta$ (reasons in Section 5.2). For our step size, we use an osculating circle based step choice, which we call the curvature step (details in Section 5.1). We provide evidence that the step size works in the section below. With these in place, we now describe our full update rule.

Let $TP_k(w_k) = \sum_{i=1}^\tau (\phi_i^\top w_k - V_i)^2 \phi_i / \|\phi_i\|^2$, and $\Delta TP_k(w_k) = TP_k([w_k - TP_k(w_k)]) - TP_k(w_k)$. Then:

$$w_{k+1} = w_k - \eta_k \frac{TP_k(w_k)}{\Delta TP_k(w_k)} + \beta (w_k - w_{k-1})$$

where $\beta \in (0, 1)$ and $\eta_k = \frac{1}{k^p} \quad p \in (0.5, 1)$

2.1. Evidence for Step Size Choice

In figure 1, we plot the errors (as measured by distance from the error minimizer $w^*$) with number of iterations for total projections with curvature step algorithm. We note the exponential convergence and that the error decreases

| State | Mean Error $\phi_i^\top w - V_i$ |
|-------|----------------------------------|
| 1     | 0.08                            | 0.91 |
| 2     | 0.08                            | 0.91 |
| 3     | 0.08                            | 0.91 |
| ...   | ...                             | ... |
| 19    | 0.07                            | 0.91 |
| 20    | -7.63                           | -3.45 |
monotonically. This shows that with the increased curvature-step size, we still have a contraction on the error function.

![Figure 1. Plot for errors with number of iterations for non-stochastic updates with curvature-step. We observe that our update is a contraction, and further note the exponential convergence.](image)

### 2.2. RL Context for Total Projections

In the context of RL, we use Equation 2 to generate two algorithms - Normalized Monte Carlo and Normalized TD(0).

In the Normalized Monte Carlo, the role of $\Phi$ is played by the feature vectors for the states. The value vector for the states takes the role of $V$. Note that here, we only have access to noise samples of $V$, which we estimate through the First-Visit Monte Carlo Algorithm. Finally, the $d_i$'s are given by the stationary distribution of the transition matrix. For the Normalized TD(0), we note that the TD error can be defined as $\delta = R_{ss'} - (\phi_s - \gamma \phi_{s'})^T w$. While algorithms often proceed to solve for $E[\delta \Phi] = 0$ (Sutton et al., 2008), we try to directly minimize $E[\delta]$. This leads us to the over-determined system: $(\phi_s - \gamma \phi_{s'})^T w = R_{ss'}$, which we solve through the methods described in Total Projections. Note that if the number of states is $|S|$, we end up with $|S|^2$ rows (which leads us to Tensor Methods to show convergence).

### 3. Normalized Monte Carlo

An adaptation of Total Projections to First Visit Monte Carlo is what we call the Normalized Monte Carlo

#### 3.1. Motivation

The traditional least squares algorithm is biased towards features which have high norm. Consider a robotic setting where features are in some continuous space (speed, acceleration, sensor readings (Sharifzadeh et al., 2017)). The least squares solution would then be dominated by features where the feature-norms are high, whereas we want to give equal importance to each sample irrespective of feature-norm.

We propose an algorithm whose output gives equal importance to states, irrespective of their feature-norm, and thus is scale invariant. Such an algorithm might be of interest in cases where features are being sampled online and one cannot guarantee all states to have similar feature-norms.

### 3.2. Notations and Problem Setup:

Let us consider an RL setting with state space $S$, where $|S| = m$. Let the states be labeled $\{1 \ldots m\}$. Consider an MDP (Markov Decision Process) given by $M = (S, A, \mathbb{P}, R)$ (Szepesvari, 2010) and a discount factor $\gamma$. Consider a deterministic stationary policy $\pi : S \rightarrow A$. This induces a transition matrix $\mathcal{P} \in \mathbb{R}^{m \times m}$. $\mathcal{P}$ gives a probability distribution over next states for each given state. The probability of transition from states $s$ to $s'$ ($s, s' \in S$) is given by $\mathcal{P}_{ss'}$. Given $s$, the vector of transition probabilities over all $s' \in S$ is given by $\mathcal{P}_s$. We will assume full mixing and ergodicity. Then let $\pi \in \mathbb{R}^m$ be the stationary distribution associated with $\mathcal{P}$, and $D \in \mathbb{R}^{m \times m}$ be the diagonal matrix associated with vector $\pi$. Let $R_{ss'}$ indicates the reward on transition between state $s$ and $s'$ ($s, s' \in S$).

Let $\phi_s \in \mathbb{R}^n$ be the set of features associated with each state and $\Phi$ be the corresponding matrix of all features.

In the value estimation problem, we want to find the value $V \in \mathbb{R}^m$, under a policy $\mu$, for each state. Then, for each state $s$ we have (Szepesvari, 2010) that $V_s = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R_{t+1} | S_0 = s \right]$. Under the linear function approximation, we estimate $V$ as $\Phi w$, where $w \in \mathbb{R}^n$ denotes the feature weights. We denote the error function as $G(\cdot)$ for Normalized Monte Carlo and $H(\cdot)$ for Normalized TD(0).

We denote the ideal weight, value vector pair in Monte Carlo as $(w^L, V^L)$, in Normalized Monte Carlo as $(w^M, V^M)$ and Normalized TD(0) as $(w^N, V^N)$. In our algorithms, we

### Algorithm 1 General Total Projections algorithm

**Input:** $\Phi$, max Iterations  
**Output:** $w^*$ - output weight vector  
Initialize $w_0$ arbitrarily  
for iterations= 1 to max-iterations do  
Sample $\tau$ states: $V_1, \phi_1, \ldots, V_T, \phi_T$  
$\alpha \leftarrow$ step size  
$\beta \leftarrow$ momentum multiplier  
$U_1 \leftarrow \frac{1}{T} \sum_{t=1}^{T} \phi_i^T w - V_t$  
$U_2 \leftarrow w_k - w_{k-1}$  
$w_{k+1} \leftarrow w_k - \alpha U_1 + \beta U_2$  
end for
Algorithm 2 Total Projections algorithm for first-visit MC

Input: $\Phi$, max iterations
Output: $w^M$, estimated ideal output weights
Initialize $w_0$ arbitrarily
for iterations $= 1$ to max-iterations do
    Get Trajectory as per policy $\mu$: $S_0, R_1, S_1, \ldots, S_{T-1}, R_T$
    $w_{k+1} \leftarrow$ MC SUBROUTINE($w_k, w_{k-1},$ Trajectory, $\Phi$)
end for

Algorithm 3 MC Subroutine for first-visit Monte Carlo

Input: $w_k, w_{k-1}$, Trajectory, $\Phi$
Output: $w_{k+1}$
initialize value estimate vector $\bar{V}$, unique states counter $\tau$ to 0
for $i = 1$ to last state do if first visit then $\tau \leftarrow \tau + 1$; $\bar{V}_\tau \leftarrow$ discounted reward sum; $\phi_\tau \leftarrow \Phi(s_i)$
end for
$\alpha \leftarrow$ step size $\beta \leftarrow$ momentum multiplier
$U_1 \leftarrow \frac{1}{\beta} \sum_{i=0}^{\tau-1} \frac{\phi_i^T w_k - \bar{V}_i}{||\phi_i||^2} \phi_i$
$U_2 \leftarrow w_k - w_{k-1}$
$w_{k+1} \leftarrow w_k - \alpha U_1 + \beta U_2$
return $w_{k+1}$

3.3. Drawbacks of the traditional algorithm

Let $\delta_s(w^L)$ represent distance of point $w^L$ from hyperplane $\phi_s^T w = V_s$. Then $\delta_s(w^L) = (\phi_s^T w - V_s)/||\phi_s||$.

The least squares weight is then given by $w^L = \arg \min_w \sum_{i=1}^m (\phi_i^T w - V_i)^2 = \arg \min_w \sum_{i=1}^m (\delta_s(w) \cdot ||\phi_s||)^2$.

Observe that $w^L$ minimizes the squared distances weighted by square of feature-norms. Thus, the least squares solution is biased towards feature vectors with large norm.

3.4. Algorithm for Normalized Monte Carlo

We outline our Total Projections (TP) method as a general method to find the scale invariant solution to an over-determined system, through repeated projections. Our main method is given in algorithm 2, where we run through a trajectory sampled from the stationary distribution. This method calls as a subroutine algorithm 3, for a one step stochastic weight update. This method is inspired by Randomized Kaczmarz (our main modifications are highlighted in appendix C). We speed up the algorithm through a novel step size method (section 5.1) and momentum (section 5.2).

3.5. Stochastic Update Equation

Theorem 1. The stochastic approximation algorithm

$$w_{k+1} = w_k - \eta_k \frac{||TP_k(w_k)||}{||\Delta TP_k(w_k)||} TP_k(w_k) + \beta(w_k - w_{k-1})$$

converges a.s. to $w^M := (\Phi^T NDN\Phi)^{-1} \Phi^T NDN V$, \hspace{1cm} (3)
where $TP_k(w_k) = \sum_{i=1}^{\tau} (\phi_i^T w - V_i)\phi_i / ||\phi_i||^2$, $N$ is diagonal with $N_{i,i} = \frac{1}{||\phi_i||^2}$, $\beta \in (0, 1)$, $\eta_k = 1/k^p$; $p \in (0,5,1]$

We will formally prove this theorem using stochastic approximation theory (Borkar, 2008)

3.6. Showing Convergence without momentum

We first show the above theorem without momentum

Theorem 2. $w_{k+1} = w_k - \alpha_k \cdot TP_k(w_k)$ without momentum converges to $w^M$ (a.s)

The major claims that we use in this proof are the following:

Fact: $V_i$’s are bounded. In other words, if $R_{max} = \max_{s,s' \in S} [R_{ss'}]$, then $V_i \leq R_{max}/(1-\gamma) \forall i$

Fact: $\tau$ is bounded as it is the number of unique states

Now let the filtration be $\mathcal{F}_k = \{w_0, \ldots, w_k\}$. For the stochastic update equation in theorem 2, let the expected update be $h_{k+1}(w_k) = \mathbb{E}[TP_k(w_k) | \mathcal{F}_k]$. Then, the update rule in standard form is $w_{k+1} = w_k - \alpha (h_{k+1}(w_k) + M_{k+1})$

Proposition 3.1. $h_{k+1}(w_k)$ is Lipschitz

Proof. Proof in Appendix D.4 and appendix E.3.1

Proposition 3.2. The step size sequence $\{\alpha_i\}_{i=1}^\infty$ satisfy

$$\sum_{i=0}^\infty \alpha_i = \infty \text{ and } \sum_{i=0}^\infty \alpha_i^2 < \infty$$

Proof Sketch. This proceeds from our construction of the step size sequence in section 5.1. See appendix E.3.2 for full proof.

Proposition 3.3. then $\{M_k\}$ is a zero-mean martingale difference noise sequence

Proof. We show this in appendix E.3.3.
**Proposition 3.4.** The iterates remain bounded almost surely, \( \sup_k w_k < \infty \) (a.s.).

**Proof.** First note that \( V_i \)s are upper-bounded. Thus the estimates for the hyperplanes are upper-bounded. Now, in a fully determined system, there is at least one, and at most \( \binom{n}{m} \) intersection points in \( \mathbb{R}^n \) of the m hyperplanes. Since each iteration brings us closer to at least one of these intersection points (by the Pythagoras theorem, as we are doing projections), and the intersection points are all bounded, the iterates are almost surely bounded.

**Proposition 3.5.** Let \( h(\cdot) \) be the function which our update equation tracks asymptotically, then the unique globally asymptotically stable equilibrium point for the limiting o.d.e given by \( \dot{w}(t) = h(w(t)) \) is given as \( w^M = [(\phi^T NDN\phi)^{-1}\phi^T NDN] V \)

**Proof.** We show this in appendix E.

**Proof of Theorem 2.** From propositions 3.1, 3.2, 3.3, 3.4, we satisfy the assumptions A1-A4 required to show convergence of a stochastic approximation equation (Borkar, 2008). Based on proposition 3.5 we converge to the unique globally asymptotically stable equilibrium point given by \( w^M = [(\phi^T NDN\phi)^{-1}\phi^T NDN] V \)

**3.7. Convergence using Momentum**

Momentum methods have been shown to converge by (Avrachenkov et al., 2020) using two time-scale methods. We thus consider one adaptation here.

**Proof of Theorem 1.** We cover the full proof in appendix G.

**Proposition 3.6.** The stochastic approximation equation with momentum can be rewritten as \( w_{k+1} - w_k = \alpha_k z_k \) \( z_i = z_{i-1} + \zeta(i,k) \left[ h(w_k) + \varepsilon(i,k) + M_{i,k} \right] \) \( \forall i \in [1,k] \) \( z_0 = [h(w_k) + M_{0,k}] \)

where \( M_{i,k} \) are martingale difference noise, coefficients \( \zeta(i,k) = \beta^\gamma \frac{\alpha_k}{\alpha_{k-1}} \) provide exponential decay, expected update \( h(\cdot) \) converges to \( w^M \) and \( \varepsilon(i,k) \) are perturbation terms.

**Proposition 3.7.** The above set of equations collapse into the stochastic equation \( w_{k+1} - w_k = \alpha_k [h(w_k) + \hat{\varepsilon}_k + \hat{M}_k] \)

where \( \hat{h}(w_k) \) converges to \( w^M \), \( \{\hat{\varepsilon}_k\} \) are perturbation terms and \( \{\hat{M}_k\} \) are martingale difference noise terms.

Now it’s easy to see convergence as in Theorem 2 as perturbation terms don’t affect convergence.

**4. Normalized TD(0)**

We now outline the Normalized TD(0) as an adaptation of the Total Projections algorithm in the TD(0) context.

**4.1. Notations and Problem Setup**

We continue the same notations as in section 3.2. Here as well, the problem is to estimate the value function using a linear function approximation, but in contrast to the Monte Carlo algorithm, we don’t have to wait for the completion of an entire episode to update the weight vector. We assume a reward of \( R_{ss'} \) on transitions from state \( s \) to \( s' \), \( (s, s') \in S \).

**4.2. Error Term and update equation**

At it’s core, the TD(0) algorithm targets minimization of the expression \( R_{ss'} + \gamma V_{s'} - V_s \). Under the value function approximation regime, we can rewrite the above expression as \( R_{ss'} + \gamma \phi_s^\top w - \phi_{s'}^\top w = R_{ss'} - (\phi_s - \gamma \phi_{s'})^\top w \), where the \( (s, s') \) are sampled from \( \pi, P_s \) respectively. We denote \( (\phi_s - \gamma \phi_{s'}) \) as \( \Delta \phi_{ss'} \) for convenience. Then we may wish to minimize the expression \( \mathbb{E}_{s \in S} \mathbb{E}_{s' \in S} \mathbb{E} \left[ \Delta \phi_{ss'}^\top w - R_{ss'} \right] \)

But based on our discussion on scale invariance in sections 1 and 3.1, we note that the minimizer of the above expression above would be biased towards large \( \Delta \phi_{ss'} \) values.

The scale invariance problem is pertinent here as may we want our state to be less affected by far away states (with lower \( \Delta \phi_{ss'} \) values). Thus we propose the error:

\[ H(w) = \mathbb{E}_{s \in S} \mathbb{E}_{s' \in S} \mathbb{E} \left[ \Delta \phi_{ss'}^\top w - R_{ss'} \right]^2 \]

Which gives us the stochastic weight update equation:

\[ w_{k+1} = w_k - \alpha \sum_{i=1}^{\tau} \left[ \frac{[\Delta \phi_i]^\top w - R_i}{||\Delta \phi_i||} \right] \frac{\Delta \phi_i}{||\Delta \phi_i||} \]

**4.3. A Differential Perspective**

Let us denote the TD(0) error with respect to a pair of states \( (s, s') \) as \( E_{ss'}(w) = (\phi_s - \gamma \phi_{s'})^\top w - R_{ss'} \). Then the error function \( H(w) = \mathbb{E}_{s, s'} \left[ E_{ss'}(w) \right]^2 \) Let \( \gamma \sim 1 \), then \( \Delta \phi \) can be considered a measure of distance in \( \phi \) space. Therefore, we can find a sequence of gradient vectors \( \nabla E_{ss'} \) satisfying \( E_{ss'}(w) = \nabla E_{ss'}(w) \cdot \Delta \phi_{ss'} \).
Algorithm 4 Total Projections for Normalized TD(0)

Input: \( \Phi \), max Iterations
Output: \( w^M \) - estimated ideal output weights
Initialize \( w_0 \) arbitrarily
for iterations = 1 to max-iterations do
    Get Trajectory: \( S_0, R_1, S_1, \ldots, S_{T-1}, R_T \)
    \( w_{k+1} \leftarrow \text{TD}(0) \text{ SUBROUTINE}(w_k, w_{k-1}, \text{Trajectory}) \)
end for

Then \( H(w) = \mathbb{E}_{s,s'} \left[ \nabla E_{s,s'}(w) \cdot \frac{\nabla \phi_{s,s'}}{||\nabla \phi_{s,s'}||} \right]^2 \). This gives our algorithm a gradient of error perspective, where we are looking to minimize the cumulative gradient of error. Note that large errors between states that are close by, would give higher error gradient. Thus our algorithm optimizes for lower error between states close by in \( \phi \) space.

4.4. When to use Normalized TD(0)

If we want the value of a given state to not be excessively affected by states that are far away in \( \phi \) space, then we should use normalized TD(0).

Note that values in \( \phi \) space often have real world meaning. Some examples are obstacle distance, or velocity (Sharifzadeh et al., 2017). These variables are continuous and ordinal (c.f. nominal variables) and \( \phi \) space distance has meaning. In such cases, our algorithm may be quite suitable.

Algorithm 5 TD(0) Subroutine

Input: \( w_k, w_{k-1}, \text{Trajectory}, \Phi \)
Output: \( w_{k+1} \)
initialize \( L, \rho \) to 0, unique states counter \( \tau \) to 0
for \( i = \text{first state to last} \) do
    if states pair \( (s_i, s_{i+1}) \) seen for first time then:
        \( \tau \leftarrow \tau + 1; \Delta \phi_{s,s'} \leftarrow \phi(s_i) - \gamma \phi(s_{i+1}) \)
        \( L_{\tau} \leftarrow \frac{\Delta \phi_{s,s'}}{||\Delta \phi_{s,s'}||}; \rho_{\tau} \leftarrow R_{s_{i+1}+1} \)
    end for
    \( \alpha \leftarrow \text{step size}; \beta \leftarrow \text{momentum multiplier} \)
    \( U_1 \leftarrow \frac{1}{\tau} \sum_{i=0}^{\tau-1} (L_i^\top w - \rho_{i}) L_i; \quad U_2 \leftarrow w_{k} - w_{k-1} \)
    \( w_{k+1} \leftarrow w_k - \alpha U_1 + \beta U_2 \)
return \( w_{k+1} \)

4.5. Algorithm for Normalized TD(0)

Let \( L_{ss'} = \frac{\Delta \phi_{s,s'}}{||\Delta \phi_{s,s'}||} \), \( \rho_{ss'} = \frac{R_{ss'}}{||\Delta \phi_{s,s'}||} \). The long matrix \( L \) can be written succinctly in the form of a tensor \( \mathcal{L} \in \mathbb{R}^{m \times n \times n} \) where the rows and columns represent \( s, s' \) respectively. Each tube-fiber at location \( (s,s') \) in the tensor represents the vector \( L_{ss'} \in \mathbb{R}^n \). Similarly, we write \( \{\rho_{ss'}\} \) in matrix form \( \mathcal{R} \). Then equation 5 can be expressed as:

\[
w_{k+1} = w_k - \frac{\alpha}{\tau} \sum_{i=1}^{\tau} (L_i^\top w_k - \rho_i) L_i + \beta(w_k - w_{k-1})
\]

where we sample \( \tau \) unique pairs: \( i = (s_i, s_{i+1}) \). Based on this update rule, we propose \textbf{Algorithm 4} where we run the trajectory for some arbitrary \( T(\geq 2) \) number of steps. Then \textbf{Algorithm 5} runs a one-step update of weight vector \( w \).

4.6. Convergence

The stochastic update in section 4.5 has a stable point \( w^N \) where \( \mathbb{E}_{s,s'} (L_i^\top w^N - \rho_i) L_i = 0 \). If The states \( (s, s') \) are sampled as per distributions \( (\pi, \mathcal{P}_s) \) respectively, then

\[
\sum_{s \in \mathcal{S}} \pi_s \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} L_{ss'} w^N = \sum_{s \in \mathcal{S}} \pi_s \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} \rho_{ss'} L_{ss'}
\]

We solve this using Tensor machinery we develop. See Section 5.3 for explanation on \( \times_1, \times_2 \) and \( \times_3 \) used below.

\textbf{Theorem 3.} \textbf{Algorithm 4} converges to

\[
w^N = [(\mathcal{L}^\top \mathcal{P} \mathcal{P}^{\top} \mathcal{L}) \times_1 D]^{-1} (\mathcal{L}^\top \mathcal{P} \mathcal{P}^{\top} \mathcal{R})^T D
\]

where \( \mathcal{L}_{ss'} = \frac{\Delta \phi_{s,s'}}{||\Delta \phi_{s,s'}||} \), \( \mathcal{R}_{ss'} = \frac{R_{ss'}}{||\Delta \phi_{s,s'}||} \), \( \mathcal{P} \) is a tensor corresponding to transition matrix \( \mathcal{P} \) such that for a given state \( s \), slices \( \mathcal{P}_s \) are diagonal matrices associated with vector \( \mathcal{P}_s \) (see Figure 2). \( D \) is a diagonal matrix corresponding to \( \pi \).

\textbf{Proposition 4.1.} Let \( h_N(\cdot) \) be the function which the normalized TD(0) update equation tracks asymptotically. Then the unique globally asymptotically stable equilibrium point for the limiting o.d.e \( \dot{w}(t) = h_N(w(t)) \) is given by \( w^N = [(\mathcal{L}^\top \mathcal{P} \mathcal{P}^{\top} \mathcal{L}) \times_1 D]^{-1} (\mathcal{L}^\top \mathcal{P} \mathcal{P}^{\top} \mathcal{R})^T D \)

\textbf{Proof Sketch.} Using appendix proposition H.4.2 and H.4.3:

\[
\mathbb{E}_{s,s'} L_{ss'} w^N = [(\mathcal{L}^\top \mathcal{P} \mathcal{P}^{\top} \mathcal{L}) \times_1 D] w^N
\]

\[
\mathbb{E}_{(s,s')\rho_{ss'}} L_{ss'} = (\mathcal{L}^\top \mathcal{P} \mathcal{P}^{\top} \mathcal{R})^T D
\]

Substituting these in equation 6, we get

\[
w^N = [(\mathcal{L}^\top \mathcal{P} \mathcal{P}^{\top} \mathcal{L}) \times_1 D]^{-1} (\mathcal{L}^\top \mathcal{P} \mathcal{P}^{\top} \mathcal{R})^T D
\]

Details are provided in appendix H.4 and appendix I.

\textbf{Proof of Thm 3.} Since the basic algorithm is the same, the assumptions A1-A4 for stochastic approximation algorithm
We show the full proof in proposition J.3.6 in section J.3 (Borkar, 2008). To achieve this, our step size sequence TD(0) solution, then

5. Step Size, Momentum and Tensor Products

5.1. Adaptive step sizes for Total Projections Algorithm

Choice of step size is extremely important for ML practitioners. We propose a novel variation for a step size sequence. First we note that to achieve convergence for a stochastic approximation algorithm, we need the step size sequences \( \{ \alpha_k \}_{k=1}^{\infty} \) to be such that \( \sum_{i=0}^{\infty} \alpha_k = \infty \) and \( \sum_{i=0}^{\infty} \alpha_k^2 < \infty \) (Borkar, 2008). To achieve this, our step size sequence \( \alpha_k = \eta_k \cdot \theta_k / ||TP_k(\cdot)|| \), where \( \eta_k = \frac{1}{k^p} \) \( p \in (0.5, 1) \). The second term \( \theta_k \) is the term of interest currently, and the third term makes the existing update term \( TP_k(\cdot) \) unit norm.

5.1.1. IDEA FOR CURVATURE STEP

To motivate the discussion, we first observe that a stochastic gradient descent (with some subset of hyperplanes) induces a gradient field on the error we are descending.

The local behavior of the field-curves can be studied by their Taylor expansion (Kuhnel, 2015). The Taylor expansion up to 2D gives us a tangential and a normal direction. Such curves have a curvature at any given point, \( \kappa \), which can be calculated from the Frenet Serret equations. When we assume the curvature is constant, we get an osculating circle.

Our claim is that for our gradient curves, we can take in the direction of the tangent (our update), a step equal to the radius of the osculating circle.

5.1.2. ESTIMATING RADIUS OF OSCULATING CIRCLE

Let \( w_k(t) \) be some stochastic gradient curve we are descending, with some subset of hyperplanes fixed. Then the curvature is given by \( \kappa = ||w''(\cdot)|| \), where \( w \) is parameterized to some unit vector in the space, and radius \( R = 1/\kappa \).

Note that our updates, \( TP_k(\cdot) \) are tangents to \( w(t) \). Since our estimates are not unit parameterized, we need an appropriate change of scale (re-parametrization). In other words, we divide our estimate for tangent by \( ||TP_k(w_k)|| \), to get the unit tangent. Similar re-scaling of our estimate for curvature yields \( ||TP_k(w_k)||^2 \) in the denominator (Chaplers, 2017).

Let \( \Delta TP_k(w_k) = TP_k(w_k - TP_k(w_k)) - TP_k(w_k) \). Then, our guess for the second derivative is \( ||\Delta TP_k(w_k)|| \), which after re-parametrization gives \( ||\Delta TP_k(w_k)||/||TP_k(w_k)||^2 \). Then we have \( R = 1/\kappa = ||TP_k(w_k)||^2/||\Delta TP_k(w_k)|| \).

Thus:

\[
\theta_k = \frac{||TP_k(w_k)||^2}{||\Delta TP_k(w_k)||}
\]

Thus our update equation (without momentum) becomes:

\[
w_{k+1} = w_k - \eta_k \theta_k \frac{TP_k(w_k)}{||TP_k(w_k)||} \]

We call the step size sequence \( \alpha_k \) as curvature-step sequence. We now provide a visual illustration and rationale for the curvature-step, for consideration alongside Figure 1.
5.1.3. **ILLUSTRATION**

Figure 3 illustrates the working of our curvature step on the step size based on the radius of the osculating circle. $w_k$ is the iterate, and $C$ is the center of the circle formed by the osculating circle. We calculate the radius based on intermediate points $w'_k$ and $w''_k$, to finally get to point $w_{k+1}$.

5.1.4. **RATIONALE**

**Claim.** Minimizer $w^*$ of the error on $G(\cdot)$ is outside the $(n-1)$ sphere in $\mathbb{R}^n$ with center at the iterate $w_k$ and radius equal to radius of osculating circle.

**Proof Sketch.** We first note that the convergence point is an attractor node, and the convergence is Lyapunov stable in expectation. The convergence rate here is slower along the lower eigenvalue-vectors of the system (Strogatz, 2000). Thus in figure 3, the iterate has more to move along other directions than along the first direction (i.e. along $TP(w_k)$). Thus the convergence point is outside the $n-1$ sphere centered at $w_k$ with radius $R$.

5.2. **Choice of Momentum**

**Momentum Method Used:** Of the various momentum optimization methods used in gradient descent algorithms (Ruder, 2017), our comparisons (figure 4) showed Heavy Ball momentum with $\beta = 0.5$ works best (reasons in appendix F). We use this for our step size sequence.

5.3. **Tensor Product**

We propose a family of the tensor product operations: $\times$, with the following classifications:

- **Mode vs Slice:** We call mode transformations as $\hat{\times}$ and slice transformations as $\check{\times}$ where modes and slices are 1-D and 2-D cuts of tensor respectively.

- **Contraction Products:** We use $\bowtie$ to describe contractive products that yield smaller tensors vs the regular $\times$.

- **Multiplication Modes:** A regular product is in the last mode whereas $\times_p$ indicates multiplication in the p’th mode of a tensor. This notation is as per (Kolda & Bader, 2009).

**Transpositions:** A regular transposition denoted $L^\top$ denotes switching of last two indices (dimensions) of tensor $L$. We also allow $L^A$ where $A$ is any permutation of indices.

Appendix I provides more detail for the family of tensor products. We use these in section H for the TD(0) problem.

6. **Simulations**

In Figure 5, we compare convergence using (1) normal Total Projections (2) Curvature Step and (3) Momentum.

7. **Conclusions and Possible Extensions**

We find there to be several scopes for extending our work:

1. One would like to test whether our proposed step size method works in case of non-linear systems (say in deep learning contexts).

2. Our tensor products should be extensible to other contexts. One may look at longer state sequences (Wu & Chu, 2017).
for Markov chains with memory.

3. One may look to undertake finite sample analysis for Normalized TD(0) (Dalal et al., 2018).

4. One may look to theoretically quantify advantages of our adaptive step size on convergence rates

5. One may look to extend our momentum convergence proofs to methods like ADAM and RMSProp
References

Avrachenkov, K., Patil, K., and Thoppe, G. Online algorithms for estimating change rates of web pages, 2020.

Axler, S. J. *Linear Algebra Done Right*. Undergraduate Texts in Mathematics. Springer, New York, 1997. ISBN 0387982582. URL http://linear.axler.net/.

Ba, L. J., Kiros, J. R., and Hinton, G. E. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Bader, B. W. and Kolda, T. G. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Trans. Math. Softw.*, 32(4):635–653, December 2006. ISSN 0098-3500. doi: 10.1145/1186785.1186794. URL https://doi.org/10.1145/1186785.1186794.

Ben-Gal, I. *Outlier Detection*, pp. 131–146. Springer US, Boston, MA, 2005. ISBN 978-0-387-25465-4. doi: 10.1007/0-387-25465-X_7. URL https://doi.org/10.1007/0-387-25465-X_7.

Bishop, C. M. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., USA, 1995. ISBN 0198538642.

Borkar, V. S. *Stochastic Approximations, A Dynamical Systems Viewpoint*. Cambridge University Press, 2008. ISBN 0521515920, 9780521515924.

Botchkarev, A. A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14:045–076, 2019. ISSN 1555-1237. doi: 10.28945/4184. URL http://dx.doi.org/10.28945/4184.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, USA, 2004. ISBN 0521833787.

Chappers. Curvature derivation for arbitrary parameterization. Mathematics Stack Exchange, 2017. URL https://math.stackexchange.com/q/2153902. Author: Chappers, https://math.stackexchange.com/users/221811/chappers, URL:https://math.stackexchange.com/q/2153902 (version: 2017-02-21).

Dalal, G., Szörényi, B., Thoppe, G., and Mannor, S. Finite sample analyses for td(0) with function approximation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/12079.

Han, J., Kamber, M., and Pei, J. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. ISBN 0123814790.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009. doi: 10.1137/07070111X.

Kuhnel, W. *Differential Geometry*. Student Mathematical Library. American Mathematical Society, 2015. ISBN 9781470423209. URL https://books.google.co.in/books?id=qNBYCwAAQBAJ.

Kushner, H. J. and Yin, G. G. *Stochastic Approximation Algorithms and Applications*. Springer New York, New York, NY, 1997. ISBN 978-1-4899-2696-8. doi: 10.1007/978-1-4899-2696-8_3. URL https://doi.org/10.1007/978-1-4899-2696-8_3.

Lakshminarayanan, C. and Bhatnagar, S. A stability criterion for two timescale stochastic approximation schemes. *Automatica*, 79:108 – 114, 2017. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2016.12.014. URL http://www.sciencedirect.com/science/article/pii/S0005109816305222.

Lathauwer, L. D., Moor, B. D., and Vandewalle, J. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000. ISSN 0895-4798. doi: http://dx.doi.org/10.1137/S0895479899305696. URL http://portal.acm.org/citation.cfm?id=354398.

Li, D., Zhang, B., and Li, C. A feature-scaling-based k-nearest neighbor algorithm for indoor positioning systems. *IEEE Internet of Things Journal*, 3(4):590–597, 2016. doi: 10.1109/JIOT.2015.2495229.
Mathematics Stack Exchange, 2021. URL https://math.stackexchange.com/q/4005591.
URL:https://math.stackexchange.com/q/4005591 (version: 2021-01-30).

Ruder, S. An overview of gradient descent optimization algorithms, 2017.

Sharifzadeh, S., Chiotellis, I., Triebel, R., and Cremers, D. Learning to drive using inverse reinforcement learning and deep q-networks, 2017.

Sigman, K. Stopping times. 2009. URL http://www.columbia.edu/~ks20/stochastic-I/stochastic-I-ST.pdf.

Singh, D. and Singh, B. Investigating the impact of data normalization on classification performance. Applied Soft Computing, 97:105524, 2020. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2019.105524. URL http://www.sciencedirect.com/science/article/pii/S1568494619302947.

Sola, J. and Sevilla, J. Importance of input data normalization for the application of neural networks to complex industrial problems. IEEE Transactions on Nuclear Science, 44(3):1464–1468, 1997. doi: 10.1109/23.589532.

Stolcke, A., Kajarekar, S., and Ferrer, L. Nonparametric feature normalization for svm-based speaker verification. In 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1577–1580, 2008. doi: 10.1109/ICASSP.2008.4517925.

Strogatz, S. H. Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering. Westview Press, 2000.

Strohmer, T. and Vershynin, R. Comments on the randomized kaczmarz method. Journal of Fourier Analysis and Applications, 15(4):437–440, aug 2009b. ISBN 1531-5851. doi: 10.1007/s00041-009-9082-0. URL https://doi.org/10.1007/s00041-009-9082-0.

Sutton, R. S. and Barto, A. G. Introduction to Reinforcement Learning. MIT Press, Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981.

Sutton, R. S., Szepesvári, C., and Maei, H. R. A convergent $\bar{i}_t\bar{O}_t/i_t(i_t\bar{O}_t/i_t)$ algorithm for off-policy temporal-difference learning with linear function approximation. In Proceedings of the 21st International Conference on Neural Information Processing Systems, NIPS’08, pp. 1609–1616, Red Hook, NY, USA, 2008. Curran Associates Inc. ISBN 9781605609492.

Szepesvari, C. Algorithms for Reinforcement Learning. Morgan and Claypool Publishers, 2010. ISBN 1608454924.

Thoppe, G., Borkar, V., and Manjunath, D. A stochastic kaczmarz algorithm for network tomography. Automatica, 50(3):910 – 914, 2014. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2013.12.016. URL http://www.sciencedirect.com/science/article/pii/S000510981300575X.

Tsitsiklis, J. N. and Van Roy, B. An analysis of temporal-difference learning with function approximation. IEEE Transactions on Automatic Control, 42(5):674–690, 1997. doi: 10.1109/9.580874.

Wu, S.-J. and Chu, M. T. Markov chains with memory, tensor formulation, and the dynamics of power iteration. Applied Mathematics and Computation, 303:226 – 239, 2017. ISSN 096-3003. doi: https://doi.org/10.1016/j.amc.2017.01.030. URL http://www.sciencedirect.com/science/article/pii/S0096300317300383.
Appendices

A. Toy Examples to show Least Squares is affected by large feature norm values

We have mentioned in the main paper how Least squares is affected by large feature norms. Here we propose concrete examples where outlier features with large norms make the outcome deviate from expected values. It will be easy to see why the least squares outcome is not the best possible in this case.

A.1. Example - 20 state chain with outlier feature

Let us consider an $m$ state Markov chain, with the first $m - 1$ features selected from some normal random distribution with mean $\mu$ and standard deviation $\sigma$. Let the $m$th state be an outlier with feature $p\mu$. Now let the transition probability matrix be as follows: From any state we can go to any other state with equal probability. Let $\gamma$ be the discount factor and reward on each transition be $r$. Then the value of each state is $r + \gamma r + \gamma^2 r + \cdots = \frac{r}{1 - \gamma}$. We note that the least squares is excessively disturbed by the outlier feature, which causes it to have a higher error that our algorithm at almost all states (except the outlier). For example, in case $m = 20$, $\mu = 1$, $\sigma = 0.05$, $p = 5$, $r = 1$, $\gamma = 0.5$ we have the following table A.1.

| State | Our Error | Least Squares Error |
|-------|-----------|---------------------|
| 1     | 0.08      | 0.91                |
| 2     | 0.08      | 0.91                |
| 3     | 0.08      | 0.91                |
| ...   | ...       | ...                 |
| 19    | 0.07      | 0.91                |
| 20    | -7.63     | -3.45               |

A.2. Example - 50 state chain with outlier feature

The more the number of states, our errors on the non-outliers get asymptotically closer to 0, thus we aren’t affected by high feature value states. For example, for a 50 state chain, with same feature distributions as in example

| State | Our Error | Least Squares Error |
|-------|-----------|---------------------|
| 1     | 0.02      | 0.54                |
| 2     | 0.03      | 0.55                |
| 3     | 0.03      | 0.54                |
| ...   | ...       | ...                 |
| 49    | 0.02      | 0.54                |
| 50    | -7.87     | -5.29               |

A.3. Example - 2 state system where least squares solution is not equal to the mean

Consider a toy example of an overdetermined $2 \times 1$ system: $2w = 1$ and $w = 2$. One may expect the answer to be the mean of $\frac{1}{2}$ and 2, i.e. $w^* = \frac{5}{2}$, but the least squares solution for this system is $w^* = \frac{1}{5}$. The least squares solution is dominated by the first feature vector, viz 2, thus gives a solution closer to the first linear equation.
B. Convergence Point of the Monte Carlo - Least Squares solution \((w^L, V^L)\)

We now calculate the convergence point of the Monte Carlo algorithm. The first visit Monte Carlo is an unbiased estimator for the value corresponding to states. Further, the updates under the Monte Carlo algorithm with linear function approximation correspond to a stochastic gradient descent on the least squares error function (Szepesvari, 2010; Sutton & Barto, 1998). We will show here that the convergence point of the algorithm is given by \(w^L = (\Phi^\top \Phi)^{-1}(\Phi^\top V)\)

**Proposition B.1.** \(w^L = (\Phi^\top \Phi)^{-1}(\Phi^\top V)\) and \(V^L = \Phi(\Phi^\top \Phi)^{-1}(\Phi^\top V)\)

**Proof.**

\[
   w^L = \arg \min_{w \in \mathbb{R}^n} \sum_{s \in S} (\phi_s^\top w - V_s)^2
\]

Taking the derivative and setting it to 0 for the arg-min:

\[
   \frac{d}{d\mathbf{w}} \left( \sum_{s \in S} (\phi_s^\top \mathbf{w}^L - V_s)^2 \right) = 0
\]

taking the derivative:

\[
   \sum_{s \in S} 2\phi_s (\phi_s^\top \mathbf{w}^L - V_s) = 0
\]

\[
   \left( \sum_{s \in S} \phi_s \phi_s^\top \right) \mathbf{w}^L = \sum_{s \in S} V_s \phi_s
\]

\[
   \left( \sum_{s \in S} \phi_s \phi_s^\top \right) = \Phi \cdot \Phi^\top \quad \text{and} \quad \left( \sum_{s \in S} \phi_s V_s \right) = \Phi^\top \cdot V
\]

Thus:

\[
   \Phi^\top \Phi \mathbf{w}^L = \Phi^\top V
\]

Then we have:

\[
   \mathbf{w}^L = (\Phi^\top \Phi)^{-1}(\Phi^\top V)
\]

\[
   V^L = \Phi \mathbf{w}^L = \Phi(\Phi^\top \Phi)^{-1}(\Phi^\top V)
\]

Thus in the case of Least Squares we have the solution given by \(V^L = \Phi \mathbf{w}^L = \Phi(\Phi^\top \Phi)^{-1}(\Phi^\top V)\)

In figure 6, we illustrate the \(\mathbb{R}^m\) perspective of the least squares solution. The least squares solution is a projection onto the column space of \(\Phi\). In other words, the solution is the point on the column space of \(\Phi\), which is at least distance from \(V\). Our claim is that such a solution may be unduly affected by rows which have large feature-norm.

For comparison, this solution can be compared with figure 7, where we illustrate in \(\mathbb{R}^n\) why distances to hyperplanes might be a scale invariant solution, which is unaffected by the feature norms.
**C. Differences in Total Projections from the traditional Kaczmarz Algorithm**

Our algorithm is a variation on the Randomized Kaczmarz algorithm described in (Strohmer & Vershynin, 2009a). We note the major differences below

1. The Randomized Kaczmarz algorithm samples the hyperplanes with a probability proportional to the square of the feature-norm, viz $||\phi_s||^2$ (Strohmer & Vershynin, 2009a). This approach has been criticized in literature (Strohmer & Vershynin, 2009b). (In our own simulations, this sampling did not provide any benefits). To sample proportional to the feature-norm square of the states, one needs to know the features-norms of all states, which may not be possible.

2. In the RL context, obtaining all possible features ab-initio is difficult, and so is sampling as per feature-norm square. Natural sampling would be as per the stationary distribution of the ergodic Markov Chain and we allow for this.

3. The original Randomized Kaczmarz method was meant for a fully determined $Ax = b$ system. Therefore, in the original setup, the iterates lie on hyperplanes onto which one projects. On the other hand, our iterates don’t lie on any hyperplane. This makes it easier to identify the sequence of iterates with a gradient field (of our error function).

4. We obtain major speedups (up to a few orders of magnitude) over the regular Kaczmarz method due to our usage of momentum and step size based on radius of osculatory-circle.

5. Our formulation makes the algorithm directly a gradient descent on the error function $\sum_{s \in S} d_s \left[ \frac{\phi_s^\top w - V_s}{||\phi_s||} \right]^2$ where $d_s$ are some positive weights corresponding the hyperplanes $H_s \equiv \phi_s^\top w - V_s$. For example, $d_s = \frac{1}{|S|}$ $\forall s \in S$ may correspond to a uniform sampling. Another example is where $d_s = \pi_s$ where $\pi$ is the stationary distribution corresponding to the Transition Matrix of a Markov Chain.
D. Properties of the Total Projections Operation

In this section, we will consider properties of the Total Projections operation $TP(w) = \frac{1}{|S|} \sum_{s \in S} d_s \left[ \frac{\phi_s^T w - V_s}{2||\phi_s||_2^2} \right] \phi_s$ and the error function $G(w) = \frac{1}{|S|} \sum_{s \in S} d_s \left[ \frac{\phi_s^T w - V_s}{2||\phi_s||_2^2} \right]^2$ such that $\sum_{s \in S} d_s \leq |S|$ where $d_s$ are some positive weights attached to hyperplanes $H_s \equiv \phi_s^T w - V_s$.

The properties shown below hold in general for positive weights $\{d_s\}_{s \in S}$ as long as $\sum_{s \in S} d_s \leq |S|$. But it may be worthwhile to consider what these positive weights may be. One example set of weight is $d_s = 1 \forall s \in S$, which may be considered as uniform weights. Another weight set is $d_s = \pi_s |S|$ where $\pi_s$ is the probability of occurrence of state $s$ in the stationary distribution, which will be of interest to us in our algorithms.

We will now show the following properties in the section numbers given:

D.1. $TP(\cdot) = \nabla_w G(\cdot)$
D.2. $G(\cdot)$ is convex
D.3. $G(\cdot)$ is strongly convex
D.4. $\nabla G(\cdot)$ is a Lipschitz function
D.5. $Hess(G(\cdot))$ is bounded above
D.6. The batch version of the Total Projections algorithm converges
D.7. Conditions on the step size of the total projection algorithm
D.8. Convergence Rate of the Total Projections Algorithm

D.1. Total Projection is a gradient descent on the error function

Proposition D.1.1. Let

$$TP(w) = \frac{1}{|S|} \sum_{s \in S} d_s \left[ \frac{\phi_s^T w - V_s}{2||\phi_s||_2^2} \right] \phi_s$$

Then $TP(w) = \nabla_w G(w)$

Proof. We obtain this by just differentiating $G(\cdot)$ with respect to $w$.

Figure 7 is an illustration of the convergence point of the Total Projections Algorithm. We have three hyperplanes in $\mathbb{R}^2$ and we attempting to find a $w$ such that $w$ is the point that minimizes the total sum of squares of distances to these hyperplanes. Note that hyperplanes are scale invariant in the sense, $\phi_s^T w = V_s$ is the same hyperplane as $c \cdot \phi_s^T w = c \cdot V_s$ for any arbitrary $c \in \mathbb{R}$. Thus our solution remains invariant under a multiplication of any row by a constant $c \in \mathbb{R}$

D.2. $G(\cdot)$ is convex

In this subsection, we will show:

(a) $G(w) = \frac{1}{|S|} \sum_{s \in S} d_s \left[ \frac{\phi_s^T w - V_s}{2||\phi_s||_2^2} \right]^2$ is convex

using:

(b) $\phi \phi^T$ is a positive semi definite matrix for all $\phi \in \mathbb{R}^n$

Proposition D.2.1. $G(\cdot)$ is convex in $\mathbb{R}^n$
Figure 7. Illustration of the Point that minimizes the sum of squares of distances to hyperplanes

Proof. We have already seen in Section D.1.1 that $\nabla_w G(w) = \frac{1}{|S|} \sum_{s \in S} d_s \left[ \frac{\phi_s^T w - V_s}{2||\phi_s||^2_2} \right] \phi_s$. Now we have

1. $\mathbb{R}^n$ is a convex set
2. $G(\cdot)$ is twice differentiable

Thus it is sufficient to show that $Hess(G(\cdot))$ is positive semi-definite.

$\nabla_w G(w) = \frac{1}{|S|} \sum_{s \in S} d_s \left[ \frac{\phi_s^T w - V_s}{2||\phi_s||^2_2} \right] \phi_s$

Then,

$Hess(G(\cdot)) = \nabla_w (\nabla_w G(w))$

$= \nabla_w \left( \frac{1}{|S|} \sum_{s \in S} d_s \left[ \frac{\phi_s^T w - V_s}{2||\phi_s||^2_2} \right] \phi_s \right)$

$= \frac{1}{|S|} \sum_{s \in S} d_s \nabla_w \left( \frac{\phi_s^T w - V_s}{||\phi_s||^2_2} \phi_s \right)$

$= \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\nabla_w (\phi_s^T w - V_s) \phi_s}{||\phi_s||^2_2} \right)$

$= \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \nabla_w (\phi_s^T w)}{||\phi_s||^2_2} \right)$

$= \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{||\phi_s||^2_2} \right)$
By proposition D.2.2, \( \text{Hess}(G(\cdot)) \) is the sum of \( |S| \) positive definite matrices, weighted by some positive coefficients \( \frac{1}{|S|} \cdot \|\phi_s\|_2^2 \). Thus \( \text{Hess}(G(\cdot)) \) is positive semi definite. Thus \( G(\cdot) \) is a convex function.

An alternate method to show \( G(\cdot) \) is convex, would be to show that \( w^T \{\text{Hess}(G(w))\} w \geq 0 \). We showed earlier that \( \text{Hess}(G(\cdot)) = \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|_2^2} \right) \). Let \( w \in \mathbb{R}^n \). Then we have

\[
w^T \{\text{Hess}(G(w))\} w = w^T \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|_2^2} \right) w
\]

where \( \phi_s \neq 0 \quad \forall i \)

\[
= \frac{1}{|S|} \sum_{s \in S} d_s \left( w^T \phi_s \phi_s^T w \right)
\]

\[
= \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s^T w}{\|\phi_s\|_2} \right) \left( \frac{\phi_s^T w}{\|\phi_s\|_2} \right)
\]

\[
= \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{y_i^T y_i}{\|\phi_s\|_2} \right)
\]

where \( y_i = \phi_s^T w \)

\[
= \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{||y||_2^2}{\|\phi_s\|_2} \right)
\]

\[
\geq 0 \quad \forall w \in \mathbb{R}^n
\]

This shows that \( G(\cdot) \) is convex.

**Proposition D.2.2.** \( \phi \phi^T \) is a positive semi definite matrix for all \( \phi \in \mathbb{R}^n \)

**Proof.** Let \( \Phi = \phi \phi^T \) (for this proposition). Then to prove \( \Phi \) is psd, it is sufficient to show \( w^T \Phi w \geq 0 \quad \forall w \in \mathbb{R}^n \)

\[
w^T \Phi w = w^T \phi \phi^T w
\]

\[
= (\phi^T w)^T (\phi^T w)
\]

\[
= y^T y \quad \text{where} \quad y = \phi^T w
\]

\[
= ||y||_2^2 \quad \text{2-norm squared of} \quad y
\]

\[
\geq 0
\]

Thus \( \Phi = \phi \phi^T \) is positive semi-definite for all \( a \in \mathbb{R}^n \)

**D.3.** \( G(\cdot) \) is strongly convex and thereby strictly convex

In this subsection, we will show:

(a) \( G(\cdot) \) is strongly convex when \( \text{rank}(\Phi) = n \)

(b) \( G(\cdot) \) is strictly convex when \( \text{rank}(\Phi) = n \)

**Proposition D.3.1.** \( G(\cdot) \) is strongly convex if \( \Phi \in \mathbb{R}^m \times n \) has rank \( n \)

**Proof.** If \( \Phi \) has rank \( n \), then the vectors \( \{\phi_s\}_{s \in S} \) span \( \mathbb{R}^n \). Then we have to show that if \( \lambda_{\text{min}} \) is the least eigenvalue of \( \text{Hess}(G(\cdot)) = \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|_2^2} \right) \), then \( \lambda_{\text{min}} > 0 \). We show this as follows:
\( \phi_s^T \phi_s \) is a rank 1 symmetric matrix. Symmetric matrices have real eigenvalues. Further,  
\[
(\phi_s^T w)^2 \geq 0 \quad \forall \phi_s, w \in \mathbb{R}^n
\]
\[
(\phi_s^T w)(\phi_s^T w) \geq 0
\]
\[
(w^T \phi_s)(\phi_s^T w) \geq 0
\]
\[
w^T (\phi_s \phi_s^T) w \geq 0
\]
\[
w^T \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) w \geq 0
\]
\[
\frac{1}{|S|} \sum_{s \in S} d_s w^T \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) w \geq 0
\]
\[
w^T \left( \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) \right) w \geq 0
\]

This means the eigenvalues of \( \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) \) are non-negative. It remains to be shown that the no eigenvalue is equal to 0. This is true as if some eigenvalue is equal to 0, then for the corresponding eigenvector, say \( w \),
\[
w^T \left( \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) \right) w = 0
\]
\[
\frac{1}{|S|} \sum_{s \in S} d_s w^T \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) w = 0
\]
\[
w^T \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) w = 0 \quad \forall \phi_s \in A
\]
\[
w^T \phi_s \phi_s^T = 0 \quad \forall \phi_s \in A
\]
\[
(\phi_s^T w)^2 = 0 \quad \forall \phi_s \in A
\]
\[
\phi_s^T w = 0 \quad \forall \phi_s \in A
\]

But this is a contradiction as \( \{\phi_s\}_{i=1}^n \) is a spanning set for \( \mathbb{R}^n \). Thus minimum eigenvalue of \( \text{Hess}(G(\cdot)) = \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) \) is greater than 0. Thus by definition of strong convexity we have that \( G(\cdot) \) is strongly convex when \( \Phi \in \mathbb{R}^{m \times n} \) has rank \( n \).

\[ \Box \]

**Proposition D.3.2.** Let \( \lambda_{\text{min}} \) be the least eigen value of \( \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) \). Then, \( G(\cdot) \) is \( \mu \)--strongly convex where \( \mu = \lambda_{\text{min}} \).

**Proof.** Note that we can show \( \mu \)--strongly convex when we show the following. Consider
\[
w^T (\text{Hess}(G(\cdot)) - \mu I) w = w^T \left( \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) - \mu I \right) w
\]
\[
= w^T \left( \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s \phi_s^T}{\|\phi_s\|^2} \right) \right) w - \mu w^T w
\]

\[ \text{where } \phi_s \neq 0 \quad \forall i \]
If $\lambda_{\text{min}}$ is the least eigenvalue of $\left( \frac{1}{|S|} \sum_{s \in S} d_s \frac{\phi_s \phi_s^T}{||\phi_s||_2^2} \right)$, $\lambda_{\text{min}} > 0$ as we have already shown

$$w^T (\text{Hess}(G(\cdot)) - \mu I) w \geq \lambda_{\text{min}} w^T w - \mu w^T w$$

$$\geq (\lambda_{\text{min}} - \mu) ||w||_2^2$$

Thus $\exists \mu \in [0, \lambda_{\text{min}})$ such that

$$w^T (\text{Hess}(G(\cdot)) - \mu I) w > 0$$

Thus we see that $G(\cdot)$ is $\mu$ strongly convex where $\mu = \lambda_{\text{min}}$

**Proposition D.3.3.** $G(\cdot)$ is strictly convex

**Proof.** Strict convexity is a subset of strong convexity. Thus $G$ is strictly convex.

**D.4. $\nabla G(\cdot)$ is a Lipschitz function**

In this subsection, we will show:

(a) $\nabla_w G(\cdot)$ is Lipschitz continuous with Lipschitz constant equal to 1

(b) $\frac{\phi_s \phi_s^T}{||\phi_s||_2^2}$ has one eigenvalue 1 and rest $n - 1$ eigenvalues 0.

**Proposition D.4.1.** Let $\phi_s \neq 0$. Then, $\mathcal{V}_s = \frac{\phi_s \phi_s^T}{||\phi_s||_2^2}$ has one eigenvalue 1 and rest $n - 1$ eigenvalues 0.

**Proof.**

**Claim 1: eigenvalues are real and $\mathcal{V}_s$ is p.s.d:** Let $\mathcal{V}_s = \frac{\phi_s \phi_s^T}{||\phi_s||_2^2}$. Then, $\mathcal{V}_s$ is symmetric thus has real eigen values ((Axler, 1997)). The second part follows from **proposition D.2.2**

**Claim 2: $\mathcal{V}_s$ is a rank 1 matrix:** We note that the rank of $vv^T = 1$ for any $v \neq 0$, $v \in \mathbb{R}^n$. This is because the rank is the dimension of the column space of the matrix. Since the columns of $vv^T$ are all scalar multiples of $v$, rank is 1

**Claim 3: $\phi_s$ is an eigenvalue of $\frac{\phi_s \phi_s^T}{||\phi_s||_2^2}$ with eigenvalue 1:** Let $\nu_s = \frac{\phi_s}{||\phi_s||_2}$. Then let $\nu_s \nu_s^T = \mathcal{V}_s$. Then we have to show $\nu_s$ is an eigenvector of $\mathcal{V}_s = \nu_s \nu_s^T$. But this is easy to see. $\mathcal{V}_s \nu_s = \nu_s \nu_s^T \nu_s = \nu_s \frac{\phi_s \phi_s^T}{||\phi_s||_2^2} = \nu_s$. Thus $\nu_s$ is an eigenvector of $\mathcal{V}_s$ with eigen value 1

**Claim 4: The other eigenvectors are orthogonal to eigenvector with eigenvalue 1** First we note that $\nu_s$ is the only eigenvector of $\mathcal{V}_s$ with eigen value 1. Then we show in general that in a real symmetric matrix, eigenvectors with distinct eigenvalues are orthogonal.

Let $\nu_1$ and $\nu_2$ be two eigenvectors of $\mathcal{V}_s$ with distinct eigenvalues $\mu_1$ and $\mu_2$. Then $\mathcal{V}_s \nu_1 = \mu_1 \nu_1$ and $\mathcal{V}_s \nu_2 = \mu_2 \nu_2$.

Consider $\mu_1 \nu_2^T \nu_1$. This is equal to $\nu_2^T (\mu_1 \nu_1) = \nu_2^T \mathcal{V}_s \nu_1 = \nu_2^T \nu_1 = (\mathcal{V}_s \nu_2)^T \nu_1 = (\mu_2 \nu_2)^T \nu_1 = \mu_2 \nu_2^T \nu_1$. Thus $\mu_1 \nu_2^T \nu_1 = \mu_2 \nu_2^T \nu_1$ for distinct $\mu_1, \mu_2$, implying that $\nu_2^T \nu_1 = 0$, or in other words $\nu_1$ and $\nu_2$ are orthogonal

**Claim 5: eigenvalue 0 has a multiplicity of $n - 1$:** It can be shown (Axler, 1997) that a rank 1 matrix has at most 1 non-zero eigenvalue and eigenvalue 0 with multiplicity $n - 1$ as follows.
First we note that there are n eigenvectors for $V_s = \nu_s \nu_s^T$ in $\mathbb{R}^n$. We have found one eigenvector $\nu_s$ with eigenvalue 1. We have also shown that all other eigenvectors are orthogonal to $\nu_s$. Consider any eigenvector $\nu$ orthogonal to $\nu_s$. Then $\nu_s^T \nu = 0$. Now consider $V_s \nu = (\nu_s \nu_s^T) \nu = \nu_s (\nu_s^T \nu) = \nu_s \cdot 0 = 0$. Thus for all $n - 1$ eigenvectors orthogonal to $\nu_s$, eigenvalue is 0.

Thus $\phi_s \phi_s^T / ||\phi_s||_2^2$ has eigenvalue 1 with multiplicity 1, and eigenvalue 0 with multiplicity $n - 1$.

Now we are ready to show the Lipschitz property of $\nabla G(\cdot)$.

**Proposition D.4.2.** $\nabla G(\cdot)$ is Lipschitz continuous.

**Proof.** We already showed that:

$$\nabla G(w) = \frac{1}{|S|} \sum_{s \in S} d_s \left( \frac{\phi_s^T w - V_s \phi_s}{||\phi_s||_2^2} \right)$$

Then we have

$$\nabla G(w) - \nabla G(y) = \left( \frac{1}{|S|} \sum_{s \in S} d_s \frac{(\phi_s^T w - V_s \phi_s)}{||\phi_s||_2^2} \right) - \left( \frac{1}{|S|} \sum_{s \in S} d_s \frac{(\phi_s^T y - V_s \phi_s)}{||\phi_s||_2^2} \right)$$

$$= \left( \frac{1}{|S|} \sum_{s \in S} d_s \frac{\phi_s^T (w - y) \phi_s}{||\phi_s||_2^2} \right)$$

$$= \left( \frac{1}{|S|} \sum_{s \in S} d_s \phi_s \phi_s^T (w - y) \right)$$

$$= \left( \frac{1}{|S|} \sum_{s \in S} d_s \phi_s \phi_s^T \right) (w - y)$$

Since max eigen value of $\phi_s \phi_s^T / ||\phi_s||_2^2$ is 1

$$||\nabla G(w) - \nabla G(y)|| \leq \left( \frac{1}{|S|} \sum_{s \in S} d_s \right) ||w - y||$$

$$||\nabla G(w) - \nabla G(y)|| \leq \frac{|S|}{|S|} ||w - y|| = ||w - y||$$

Thus the function $G(\cdot)$ is Lipschitz where the Lipschitz constant, $L \leq 1$.

**D.5. The Hessian of $G(\cdot)$ is bounded above**

In this subsection, we will show:

(a) The Hessian of $G(\cdot)$ is bounded above, or

$$HessG(\cdot) \preceq I$$

where $I$ is the identity matrix.

**Proposition D.5.1.** $HessG(\cdot) \preceq I$ where $I$ is the identity matrix.
Proof. The proposition is equivalent to showing \( w^\top \text{Hess}(G(w)) w \leq \|w\|^2 \)

Using \( \text{Hess}(G(\cdot)) = \frac{1}{|S|} \sum_{s \in S} d_s \left[ \frac{\phi_s \phi_s^\top}{\| \phi_s \|^2} \right] \), for some \( w \in \mathbb{R}^n \), we have

\[
w^\top \text{Hess}(G(w)) w = w^\top \frac{1}{|S|} \sum_{s \in S} d_s \left[ \frac{\phi_s \phi_s^\top}{\| \phi_s \|^2} \right] w \]

where \( \phi_s \neq \mathbf{0} \quad \forall i \)

Since the maximum eigenvalue of \( \frac{\phi_s \phi_s^\top}{\| \phi_s \|^2} = 1 \) by proposition D.4.1,

\[
w^\top \text{Hess}(G(w)) w = \frac{1}{|S|} \sum_{s \in S} d_s w^\top \left[ \frac{\phi_s \phi_s^\top}{\| \phi_s \|^2} \right] w \leq \frac{1}{|S|} \sum_{s \in S} d_s \|w\|^2 \]

\[
\leq \|w\|^2
\]

D.6. The batch version of the Total Projections algorithm converges

Now we proceed to prove convergence of the batch version (non stochastic version) of the TP algorithm. We have already shown \( G(\cdot) \) is convex. Thus, we know that it has a unique optimum point. Thus if our algorithm converges to some optimum, it is guaranteed that we will converge to the unique optimum.

**Proposition D.6.1.** Let \( w^* \) be the minimizer of \( G(w) = \frac{1}{|S|} \sum_{s \in S} d_s \left[ \frac{|\phi_s^\top w - V_s|^2}{2\| \phi_s \|^2} \right] \). Then if the sequence \( \{w_1, w_2, w_3, \ldots \} \) is obtained by successive total projection operations, starting from some arbitrary point \( w_0 \in \mathbb{R}^n \), then \( w_\infty = w^* \)

**Proof.** Consider the algorithm \( w_{k+1} = w_k - \alpha_k (TP(w_k)) \) where \( TP(w_k) = \nabla_w G(w_k) = \frac{1}{|S|} \sum_{s \in S} d_s \left[ (\phi_s^\top w_k - V_s) \phi_s \right] \) and \( \alpha_k \) is some step size sequence. This is a gradient descent algorithm on \( G(\cdot) \). It has been proved in literature (Boyd & Vandenberghe, 2004) that a (batch) gradient descent algorithm converges to the local minimizer. Since we have shown that \( G(\cdot) \) is a convex function over a convex set, it has a single local minimizer, which is also the global optimum.

We start with the second order Taylor series expansion of \( G(\cdot) \) at some point \( y \in \mathbb{R}^n \) in the neighborhood of \( w \in \mathbb{R}^n \), and some \( z \) between \( w \) and \( y \), we have

\[
G(y) = G(w) + \nabla G(w)^\top (y - w) + \frac{1}{2} (y - w)^\top \text{Hess}(G(z))(y - w)
\]

By proposition D.5.1, \( H(G(\cdot)) \) is bounded above by 1

\[
G(y) \leq G(w) + \nabla G(w)^\top (y - w) + \frac{1}{2} \|y - w\|^2
\]
In gradient descent, we proceed in the opposite direction of the gradient. \( : y - w = -\alpha \nabla G(w) \)

\[
G(y) \leq G(w) - \alpha \nabla G(w) \top \nabla G(w) + \frac{1}{2} ||y - w||^2
\]

Then, \( ||y - w||_2 = \alpha ||\nabla G(w)||_2 \)

\[
G(y) \leq G(w) - \alpha ||\nabla G(w)||^2_2 + \frac{1}{2} (\alpha ||\nabla G(w)||_2)^2
\]

\[
G(y) \leq G(w) - (\alpha - \frac{\alpha^2}{2}) ||\nabla G(w)||^2_2
\]

If \( \alpha \in (0, 2) \), then \( \alpha - \frac{\alpha^2}{2} = \alpha \left( 1 - \frac{\alpha}{2} \right) > 0 \). We then have:

\[
G(y) \leq G(w) - c ||\nabla G(w)||^2_2 \quad \text{for some constant } c > 0
\]

\[
G(y) < G(w)
\]

Since \( TP(w) = \nabla G(w) \), we have \( y = w - \alpha TP(w) \) where \( \alpha \in (0, 2) \)

\[
G(w - \alpha TP(w)) < G(w)
\]

If we label the successive iterates as \( w_k \) and \( w_{k+1} \), and the step size for the k'th step as \( \alpha_k \):

\[
G(w_{k+1}) < G(w_k) \quad \text{for } \alpha_k \in (0, 2)]
\]

Let \( w^* = \arg \min_{w \in \mathbb{R}^n} G(w) \). Then:

\[
G(w_{k+1}) - G(w^*) < G(w_k) - G(w^*)
\]

Then for some constant \( \gamma_k < 1, \gamma_k \in \mathbb{R} \):

\[
G(w_{k+1}) - G(w^*) = \gamma_k (G(w_k) - G(w^*))
\]

Similarly, for some constant \( \gamma_k - 1 < 1, \gamma_k - 1 \in \mathbb{R} \):

\[
G(w_{k+1}) - G(w^*) = \gamma_k - 1 \gamma_k (G(w_{k-1}) - G(w^*))
\]

\[
= \ldots
\]

\[
G(w_{k+1}) - G(w^*) = \left( \prod_{i=0}^k \gamma_i \right) (G(w_0) - G(w^*)) \quad \text{where } \gamma_i < 1 \ \forall i \in \{0, \ldots, k\}
\]

Now we take the limit as \( k \to \infty \)

\[
\lim_{k \to \infty} (G(w_{k+1}) - G(w^*)) = \lim_{k \to \infty} \left( \prod_{i=0}^k \gamma_i \right) (G(w_0) - G(w^*)) \quad \text{where } \gamma_i < 1 \ \forall i \in \lim_{k \to \infty} \{0, \ldots, k\}
\]

\[
G(w_{\infty}) - G(w^*) = \left( \lim_{k \to \infty} \prod_{i=1}^k \gamma_i \right) (G(w_1) - G(w^*)) \quad \text{where } \gamma_i < 1 \ \forall i \in \lim_{k \to \infty} \{1, \ldots, k\}
\]
Since the product of infinite numbers less than 1 is 0, we have:

$$G(w_\infty) - G(w^*) = 0$$

$$G(w_\infty) = G(w^*)$$

Since $G(\cdot)$ is convex over $\mathbb{R}^n$, there $w^*$ is the unique minimizer

$$w_\infty = w^*$$

Thus we show convergence. To get rate of convergence, we need to make some assumptions about $\alpha$.

### D.7. Conditions on the step size of the total projection algorithm

We showed in proposition D.6.1 that the batch version of Total Projections converges to the global optimum for $\alpha_k \in (0, 2)$. Now we will study what is the ideal step size to take in this above range as part of the TP algorithm.

**Proposition D.7.1.** The optimal step-size $\alpha_{OPT} = 1$

**Proof.** We have already seen in proposition D.6.1 that for some $w_{k+1}$ in the neighborhood of $w_k$, we have

$$G(w_{k+1}) \leq G(w_k) - (\alpha - \frac{\alpha^2}{2}) \|\nabla G(w_k)\|^2$$

which is quadratic in $\alpha$. If we want to minimize the LHS, with respect to $\alpha$, we set the derivative of the RHS with respect to $\alpha$ to 0. Thus for an optimal alpha, viz. $\alpha_{OPT}$ we have:

$$\nabla_\alpha G(w_{k+1}) = 0$$

$$\nabla_\alpha \left( G(w_k) - \left[ \alpha_{OPT} - \frac{\alpha_{OPT}^2}{2} \right] \|\nabla G(w_k)\|^2 \right) = 0$$

$$\nabla_\alpha G(w_k) - \nabla_\alpha \left[ \alpha_{OPT} - \frac{\alpha_{OPT}^2}{2} \right] \|\nabla G(w_k)\|^2 = 0$$

Since $\nabla_\alpha G(w_k) = 0$ and $\|\nabla G(w_k)\|^2$ is independent of $\alpha$

$$\nabla_\alpha \left[ \alpha_{OPT} - \frac{\alpha_{OPT}^2}{2} \right] = 0$$

which leads to:

$$1 - \alpha_{OPT} = 0$$

Thus

$$\alpha_{OPT} = 1$$

In light of this, the stochastic update equation for the batch version of the TP algorithm is

$$w_{k+1} = w_k - \left( \frac{1}{|S|} \sum_{s \in S} d_s \left[ \phi_s^T w - V_s \phi_s \right] \phi_s \right)$$
D.8. Convergence Rate of the Total Projections Algorithm

Now we are ready to show the exponential convergence rate for the Total Projections algorithm. We will now show:

(a) Rate of convergence of the TP algorithm is exponential when \( \Phi \) has full column rank

(b) \( G(w_{k+1}) - G(w^*) \leq - \frac{1}{2} \| \nabla G(w_k) \|^2 \)

(c) \( \| \nabla G(w_k) \|^2 \geq \frac{2(G(w_k) - G(w^*))}{2 - \lambda_{min}} \) (Note: \( \lambda_{min} \leq \lambda_{max} \leq |S| \))

**Proposition D.8.1.** Rate of convergence of the TP algorithm is exponential when \( \Phi \) has full column rank

**Proof.** Firstly, from proposition D.8.2, we have:

\[
G(w_{k+1}) - G(w^*) \leq G(w_k) - G(w^*) - \frac{1}{2} \| \nabla G(w_k) \|^2
\]

Then from proposition D.8.3 we have:

\[
\| \nabla G(w_k) \|^2 \geq \frac{2(G(w_k) - G(w^*))}{2 - \lambda_{min}}
\]

Combining, we get:

\[
G(w_{k+1}) - G(w^*) \leq G(w_k) - G(w^*) - \frac{1}{2} \frac{2(G(w_k) - G(w^*))}{2 - \lambda_{min}}
\]

\[
= [G(w_k) - G(w^*)] \left[ 1 - \frac{1}{2 - \lambda_{min}} \right]
\]

\[
= [G(w_k) - G(w^*)] \left[ \frac{1}{2 - \lambda_{min}} \right]^n
\]

We now can create a telescoping product. For successive iterates \( \{w_1, \ldots, w_k\} \):

\[
G(w_{k+1}) - G(w^*) \leq [G(w_k) - G(w^*)] \left[ \frac{1}{2 - \lambda_{min}} \right]^n
\]

\[
\leq [G(w_{k-1}) - G(w^*)] \left[ \frac{1}{2 - \lambda_{min}} \right]^{n-1}
\]

\[
\leq \ldots
\]

\[
\leq [G(w_0) - G(w^*)] \left[ \frac{1}{2 - \lambda_{min}} \right]^{n+1}
\]

Thus we have a Q-linear rate of convergence, also known as exponential rate of convergence

\[
G(w_{k+1}) - G(w^*) \leq G(w_k) - G(w^*) - \frac{1}{2} \| \nabla G(w_k) \|^2
\]

**Proposition D.8.2.**

**Proof.** From equation 14 in proposition D.7.1, we can see \( G(w_{k+1}) \leq G(w_k) - (\alpha - \alpha^2 \| \nabla G(w_k) \|^2 \). Substituting \( \alpha = 1 \) from proposition D.7.1, we get

\[
G(w_{k+1}) \leq G(w_k) - \frac{1}{2} \| \nabla G(w_k) \|^2
\]
Then subtracting $G(w^*)$ from both sides:

$$G(w_{k+1}) - G(w^*) \leq G(w_k) - G(w^*) - \frac{1}{2} \|\nabla G(w_k)\|^2$$

□

**Proposition D.8.3.** $\|\nabla G(w_k)\|^2 \geq \frac{2}{2 - \lambda_{\text{min}}} (G(w_k) - G(w^*))$

**Proof.** From proposition D.3.2 we note that when $\Phi$ has full column rank, then $G(\cdot)$ is $\mu-$ strongly convex, with $\mu = \lambda_{\text{min}}$, where $\lambda_{\text{min}}$ is the least eigenvalue of $\text{Hess}(G(\cdot))$

Let $w_{k+1} \in \mathbb{R}^n$ be some point in the neighborhood of $w_k \in \mathbb{R}^n$, and $z$ be a point in the interval $[w_k, w_{k+1}]$. Then by second order Taylor series expansion,

$$G(w_{k+1}) = G(w_k) + \nabla_w G(w_k)^T (w_{k+1} - w) + \frac{1}{2} (w_{k+1} - w_k)^T \text{Hess}(G(z))(w_{k+1} - w_k)$$

Since the Hessian is bounded below:

$$G(w_{k+1}) \geq G(w_k) + \nabla_w G(w_k)^T (w_{k+1} - w_k) + \frac{1}{2} (w_{k+1} - w_k)^T \lambda_{\text{min}} (w_{k+1} - w_k)$$

$$= G(w_k) + \nabla_w G(w_k)^T (w_{k+1} - w_k) + \frac{\lambda_{\text{min}}}{2} \|w_{k+1} - w_k\|^2$$

But if $w_{k+1} = w_k - \nabla G(w_k)$ or $w_{k+1} - w_k = -\nabla G(w_k)$. Thus,

$$G(w_{k+1}) \geq G(w_k) + \nabla_w G(w_k)^T (-\nabla G(w_k)) + \frac{\lambda_{\text{min}}}{2} \|\nabla G(w_k)\|^2$$

$$= G(w_k) - \|\nabla_w G(w_k)\|^2 + \frac{\lambda_{\text{min}}}{2} \|\nabla G(w_k)\|^2$$

$$= G(w_k) - \left[\frac{2 - \lambda_{\text{min}}}{2}\right] \|\nabla G(w_k)\|^2$$

$$\left[\frac{2 - \lambda_{\text{min}}}{2}\right] \|\nabla G(w_k)\|^2 \geq G(w_k) - G(w_{k+1})$$

But $G(w_k) - G(w_{k+1}) > G(w_k) - G(w^*)$. Thus

$$\left[\frac{2 - \lambda_{\text{min}}}{2}\right] \|\nabla G(w_k)\|^2 \geq G(w_k) - G(w^*)$$

$$\|\nabla G(w_k)\|^2 \geq \left[\frac{2}{2 - \lambda_{\text{min}}}\right] (G(w_k) - G(w^*))$$

(Note: $\lambda_{\text{min}} \leq \lambda_{\text{max}} \leq 1$)

□
E. Convergence of the Normalized Monte Carlo without momentum

In this section, we will show the convergence point of the Normalized Monte Carlo.

E.1. Notation and Problem Setup

Firstly note that under the linear function approximation regime, we are solving the overdetermined system \( \Phi w = \tilde{V} \) with \(|S| = m\) hyperplanes of the form \( H_s = \phi_s^T w - \tilde{V}_s = 0 \). We know that the first visit Monte Carlo, is an unbiased estimator of the value function \( V \) for each state. Thus \( \tilde{V} \) is an unbiased estimator of \( V \)

The sampling of the hyperplanes is as per the stationary distribution of the transition matrix \( P \). The stationary distribution is given by \( \pi \) with the probability of \( H_s \) given by \( \pi_s \). We denote the diagonal matrix associated with \( \pi \) as \( D \). Finally, we define a normalization matrix \( N \) where \( N \) is a diagonal matrix with \( N_{(s,i)} = \frac{1}{||\phi(i)||^2} \)

Let us define \( TP_k(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n \) to be the function that takes a point \( w_k \) and gives us the shift in \( w_k \) for the \( k \)'th iteration. Thus \( TP_k(w_k) = \frac{1}{\tau} \sum_{i=1}^{\tau} \phi_i^T w_k - \tilde{V}_i \phi_i \). Further, let us define \( TP(\cdot) \) (ref. section D) as \( TP(w_k) = \sum_s \pi_s \phi_s^T w_k - \tilde{V}_s \phi_s \)

\( TP_k(w_k) \) then depends on the trajectory for the Monte Carlo. In other words, it depends on the set of hyperplanes sampled (which is random), where the number of hyperplanes sampled is also random. We will assume that the stopping time \( \tau \) is obtained by some independent random process (Sigman, 2009). In other words, \( \tau \) is an independent Random Variable.

We make this assumption as if \( \tau \) is dependent explicitly on landing at certain states in the Markov Chain, then we lose the stationarity of the distribution as all states will eventually reach the absorbing states.

The limiting ODE that the stochastic update equation \( w_{k+1} = w_k - \alpha_k TP_k(w_k) \) tracks is given by \( \dot{w}(t) = h_{k+1}(w(t)) \) where \( h_{k+1}(w) = \mathbb{E} \left[ TP_k(w_k) | \mathcal{F}_k \right] \) for the filtration \( \mathcal{F}_k = \{w_0, \ldots, w_k\} \). Note that \( \dot{w}(t) = h_{k+1}(w(t)) \) is a well studied o.d.e which converges to the point where \( \dot{w}(t) = 0 \) (Borkar, 2008). Let us denote this point as \( w^M \). Then the problem in this section is to find the point of convergence, \( w^M \).

E.2. Putting the update equation in standard form:

Consider the update equation \( w_{k+1} = w_k - \alpha_k TP_k(w_k) \). Given the filtration \( \mathcal{F}_k = \{w_0, \ldots, w_k\} \), we wish to find \( h_{k+1}(w) \). Let \( \{1 \ldots \tau\} \) be the set of unique hyperplanes sampled on the \( k \)'th run of trajectory. Then:

\[
h_{k+1}(w) = \mathbb{E} \left[ TP_k(w_k) | \mathcal{F}_k \right] = \mathbb{E} \left[ \frac{1}{\tau} \sum_{i=1}^{\tau} \phi_i^T w_k - \tilde{V}_i \phi_i | \mathcal{F}_k \right]
\]

We note that \( \tau \), the set of hyperplanes \( \{1 \ldots \tau\} \) sampled, as well as \( \tilde{V} \) are all random variables. To simplify from the three random variables, first we write the above expression as an expectation over the conditional expectation given \( \tau \). Then:

\[
h_{k+1}(w) = \mathbb{E}_\tau \left[ \mathbb{E} \left[ \frac{1}{\tau} \sum_{i=1}^{\tau} \phi_i^T w_k - \tilde{V}_i \phi_i | \mathcal{F}_k, \tau \right] \right]
\]

By linearity of expectation, we can take the expectation inside the brackets:

\[
= \mathbb{E}_\tau \left[ \frac{1}{\tau} \sum_{i=1}^{\tau} \mathbb{E} \left[ \phi_i^T w_k - \tilde{V}_i \phi_i | \mathcal{F}_k, \tau \right] \right]
\]
But any hyperplane $\mathcal{H}_i \equiv \phi_i^Tw - V_i$ is chosen with probability equal to $\pi_i$ where $\pi$ is the stationary distribution. Thus the weights $d_s$ that we used in Section D now take the form $d_s = \pi_s|S|$. Thus $\forall i \in \{1, \ldots, \tau\}$, $E_{(i,\bar{V})} \left[ \frac{\phi_i^Tw - \bar{V}_i}{||\phi_i||^2} \phi_i | F_k, \tau \right] = \frac{1}{|S|} \sum_{s \in S} \pi_s|S| \cdot E_{\bar{V}} \left[ \frac{\phi_s^Tw - \bar{V}_s}{||\phi_s||^2} \phi_s \right] = \sum_{s \in S} \pi_s \left[ \frac{\phi_s^Tw - \bar{V}_s}{||\phi_s||^2} \phi_s \right]$

Substituting this back in (17), we get:

$$h_{k+1}(w) = E_{\tau} \left[ \frac{1}{\tau} \sum_{i=1}^{\tau} \sum_{s \in S} \pi_s \frac{\phi_s^Tw - \bar{V}_s}{||\phi_s||^2} \phi_s \right]$$

Since each of the terms in the sum is the same:

$$= E_{\tau} \left[ \sum_{s \in S} \pi_s \frac{\phi_s^Tw - \bar{V}_s}{||\phi_s||^2} \phi_s \right]$$

Since each term inside is independent of $\tau$, the expectation stays the same. Thus:

$$h_{k+1}(w) = \sum_{s \in S} \pi_s \frac{\phi_s^Tw - \bar{V}_s}{||\phi_s||^2} \phi_s$$

But the RHS is simply $TP(w_k)$ for the Monte Carlo. Thus:

$$h_{k+1}(w) = TP(w_k)$$

(18)

Since the function $h_k(\cdot)$ is constant for all $k$, i.e. $h_k(\cdot) = TP(\cdot)$, we can simply refer to this as $h(\cdot) = TP(\cdot)$.

Now we are in a position to put our update equation in standard form. Let $M_{k+1} = TP_k(w_k) - E[TP_k(w_k)|F_k]$, then we can write the update rule as:

$$w_{k+1} = w_k - \alpha_k(h(w_k) + M_{k+1})$$

(19)

where $\alpha_k$ is as defined in section 5.1. Further, $h(w_k) = TP(w_k)$ and $M_{k+1} = TP_k(w_k) - TP(w_k)$.

In the next section we will show that the four conditions required for convergence ((Borkar, 2008)) are satisfied. In the section after that we will show the point it converges to.

**E.3. Showing satisfaction of assumptions A1-A4 required for convergence**

We need to show the following assumptions are satisfied:

1. The map $h(\cdot)$ is Lipschitz
2. Step sizes $\{\alpha_k\}$ are positive scalars satisfying $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$
3. $\{M_k\}$ is a martingale difference sequence with respect to the filtrations $F_k$.
   Further $\{M_k\}$ are square integrable with $E \left[ ||M_{k+1}||^2 | F_k \right] \leq K(1 + ||w_k||^2)$ a.s. for some positive constant $K$
4. The iterates $\{w_k\}$ remain bounded almost surely

We will show these in order.
E.3.1. The map $h(\cdot)$ is Lipschitz

In Appendix section D.4, we showed that the TP update is Lipschitz for general weights $d_s$ as long as $\sum_{s \in \mathcal{S}} d_s \leq |\mathcal{S}|$. Now we are considering the specific case where $d_s = \pi_s |\mathcal{S}|$. Since $\pi$ is a probability distribution (and therefore sums to 1), we satisfy $\sum_{s \in \mathcal{S}} d_s \leq |\mathcal{S}|$. Thus $h(\cdot)$ is Lipschitz.

E.3.2. The sequence $\{\alpha_k\}$ is square summable but not summable

**Proposition E.3.1.** The step size sequence $\{\alpha_k\}_{k=1}^{\infty}$ satisfies $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$

**Proof.** We provide the full proof for proposition 3.2 as follows.

$$\sum_{k=0}^{\infty} \alpha_k = \sum_{k=0}^{\infty} \frac{\theta_k \eta_k}{||TP_k(w_k)||} = \sum_{k=0}^{\infty} \frac{\theta_k}{k^p \cdot ||TP_k(w_k)||}$$

Expanding $\theta_k = \frac{||TP_k(w_k)||^2}{||\Delta TP_k(w_k)||}$, we get:

$$\sum_{k=0}^{\infty} \frac{||TP_k(w_k)||}{||\Delta TP_k(w_k)||}$$

Let $\vartheta_k = \frac{||TP_k(w_k)||}{||\Delta TP_k(w_k)||}$. Then:

$$\sum_{k=0}^{\infty} \frac{\vartheta_k}{k^p}$$

We first show the almost sure lower bounds on $TP_k(w_k)$ and $\Delta TP_k(w_k)$. Note that $TP_k(w_k) = \frac{1}{\tau} \sum_{i=1}^{\tau} \left[ \frac{\phi_i^T w_k - \bar{V}_i}{||\phi_i||^2} \right] \phi_i = \frac{1}{\tau} \sum_{i=1}^{\tau} \left[ \frac{\phi_i^T}{||\phi_i||^2} \right] w_k - \frac{1}{\tau} \sum_{i=1}^{\tau} \left[ \frac{\bar{V}_i}{||\phi_i||^2} \right] \phi_i$ is almost surely not equal to 0 for random $w_k \in \mathbb{R}^n$. For $\Delta TP_k(w_k)$, we write:

$$\Delta TP_k(w_k) = TP_k(w_k - TP_k(w_k)) - TP_k(w_k)$$

$$= \frac{1}{\tau} \sum_{i=1}^{\tau} \left[ \frac{\phi_i^T}{||\phi_i||^2} \right] (w_k - TP_k(w_k)) - \bar{V}_i \phi_i$$

$$= \frac{1}{\tau} \sum_{i=1}^{\tau} \left[ \frac{\phi_i^T}{||\phi_i||^2} \right] TP_k(w_k)$$

We firstly note that $TP_k(w_k) \neq 0$ a.s. Further, for any given vector $TP_k(w_k)$, the chance of $TP_k(w_k)$ being perpendicular to all the vectors $\{\phi_i\}_{i=1}^{\tau}$ is almost surely 0. Thus $\Delta TP_k(w_k) \neq 0$ a.s.

For the upper bounds, we first note that the iterates $\{w_k\}$ are bounded a.s. as per proposition 3.4 and Appendix G.7. Then we further have that the estimates $\bar{V}$ are bounded by $R_{max}/1 - \gamma$ where $R_{max}$ is the maximum reward on transitions and $\gamma$ is the discounting factor. Since, the iterates $\{w_k\}$ are bounded, $TP_k(w_k)$ and $\Delta TP_k(w_k)$ are upper bounded.

Now by these statements, $\vartheta_k = \frac{||TP_k(w_k)||}{||\Delta TP_k(w_k)||}$ is upper and lower bounded almost surely. 2

---

2Note: In our simulations, such points where $||\Delta TP_k(w_k)|| \sim 0$ were never reached and iterates were stable even very close to the solution (see Figure 1). But to ensure algorithmic stability (given limited floating point precision), we can physically set the updates to not occur when $\Delta TP_k(w_k)$ is below a certain $\varepsilon$ (say $10^{-6}$) threshold.
Then, let $\overline{\vartheta} = \sup k \theta_k$ and $\underline{\vartheta} = \inf k \vartheta_k$. Then

$$\sum_{k=0}^{\infty} \alpha_k = \sum_{i=0}^{\infty} \vartheta_k k^p$$

$$\geq \sum_{k=0}^{\infty} \vartheta \frac{1}{k^p}$$

$$= \vartheta \sum_{k=0}^{\infty} \frac{1}{k^p}$$

$$= \vartheta \propto \infty$$

$$= \infty$$

Similarly,

$$\sum_{k=0}^{\infty} \alpha_k^2 = \sum_{k=0}^{\infty} \vartheta_k^2 \eta_k^2$$

$$\leq \vartheta \sum_{k=0}^{\infty} \eta_k^2$$

$$= \vartheta \sum_{k=0}^{\infty} \frac{1}{k^{2p}}$$

Now since $\sum_{k=0}^{\infty} \frac{1}{k^{2p}}$ is finite, and $\vartheta$ is finite. Thus:

$$\sum_{k=0}^{\infty} \alpha_k^2 < \infty$$

\[\square\]

**E.3.3.** $\{M_k\}$ IS A MARTINGALE DIFFERENT SEQUENCE THAT IS SQUARE INTEGRABLE:

We need to show that $E[M_{k+1} | F_k] = 0$ a.s. and $E[|M_{k+1}|^2 | F_k] \leq K(1 + \|w_k\|^2)$ where $M_{k+1} = TP_k(w_k) - TP(w_k)$

For the first part, we have that

$$E[M_{k+1} | F_k] = E[TP_k(w_k) - TP(w_k) | F_k]$$

$$= E[TP_k(w_k) | F_k] - E[TP(w_k) | F_k]$$

Then since $E[TP(w_k) | F_k] = TP(w_k)$, we get:

$$= E[TP_k(w_k) | F_k] - TP(w_k)$$

But we already computed in appendix E.2 that $E[TP_k(w_k) | F_k] = TP(w_k)$. Thus:

$$= TP(w_k) - TP(w_k)$$

$$= 0$$
For the second part, we write
\[ M_{k+1} = TP_k(w_k) - TP(w_k) \]
\[ = \frac{1}{\tau} \sum_{i=1}^{\tau} \left[ \phi_i^T w_k - \tilde{V}_i \right] \phi_i - \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T w_k - V_s}{||\phi_s||^2} \right] \phi_s \]
\[ = \frac{1}{\tau} \sum_{i=1}^{\tau} \left[ \phi_i \phi_i^T - \sum_{s \in S} \pi_s \phi_s \phi_s^T \right] w_k - \frac{1}{\tau} \sum_{i=1}^{\tau} \tilde{V}_i \phi_i - \sum_{s \in S} \pi_s \phi_s V_s \phi_s \]
We can call \( \frac{1}{\tau} \sum_{i=1}^{\tau} \phi_i \phi_i^T - \sum_{s \in S} \pi_s \phi_s \phi_s^T \) as \( A \) and \( \frac{1}{\tau} \sum_{i=1}^{\tau} \tilde{V}_i \phi_i - \sum_{s \in S} \pi_s \phi_s V_s \phi_s \) as \( b \). Then:
\[ M_{k+1} = Aw_k - b \]
Note that the eigenvalues of \( A \) are bounded as each \( \frac{\phi_i \phi_i^T}{||\phi_i||^2} \) has a maximum eigenvalue of 1 as per proposition D.4.1.
Similarly \( b \) is bounded as \( \tilde{V} \) is bounded by \( \frac{R_{\text{max}}}{1 - \gamma} \) where \( R_{\text{max}} \) is the maximum reward on transitions between states and \( \gamma \) is the discounting factor.
Now we see that \( M_{k+1} = Aw_k - b \) is linear in \( w_k \) with bounded coefficients. Thus \( \mathbb{E} \left[ ||Aw_k - b||^2 | F_k \right] \) is quadratic in \( w_k \).
Now it’s straightforward to see that there exists some constant \( K \) such that \( \mathbb{E} \left[ ||M_{k+1}||^2 | F_k \right] \leq K(1 + ||w_k||^2) \).

E.3.4. The iterates remain bounded almost surely
We have already shown this in proposition 3.4. We also provide a proof based on stability criterion from (Lakshminarayanan & Bhatnagar, 2017) in appendix section G.7
Now that we satisfy conditions A1-A4 for iterate convergence (Borkar, 2008) in sections E.3.1 to E.3.4, we know that the iterates will converge. It remains to be seen where it converges to, which we will cover in the next section.

E.4. Convergence point of the Normalized Monte Carlo
In this section we will show that:

(a) If \( w^M \) is the point of convergence of the normalized Monte Carlo Algorithm, then
\[ w^M = \left[ (\Phi^T NDN\Phi)^{-1} \Phi^T NDN \right] V \]
using
\[ \left( \sum_{s \in S} \pi_s \frac{\phi_s \phi_s^T}{||\phi_s||^2} \right) w^M = \sum_{s \in S} \pi_s \left[ \frac{\phi_s V_s}{||\phi_s||^2} \right] \]
\[ \sum_{s \in S} \pi_s \frac{\phi_s V_s}{||\phi_s||^2} = \Phi^T NDN\Phi \]
\[ \sum_{s \in S} \pi_s \frac{\phi_s V_s}{||\phi_s||^2} = \Phi^T NDNV \]

(b) \( \sum_{s \in S} \sum_{s \in S} \pi_s \frac{\phi_s \phi_s^T}{||\phi_s||^2} \) using
\[ \sum_{s \in S} \pi_s \frac{\phi_s \phi_s^T}{||\phi_s||^2} = \Phi^T NDN\Phi \]
\[ \sum_{s \in S} \pi_s \frac{\phi_s V_s}{||\phi_s||^2} = \Phi^T NDNV \]

Proposition E.4.1. The convergence point \( w^M \) of our algorithm, which is the stable point of the o.d.e that our stochastic update equation tracks, satisfies the condition
\[ w^M = \left[ (\Phi^T NDN\Phi)^{-1} \Phi^T NDN \right] V \]
Scale Invariant Solutions for Overdetermined Linear Systems with Applications to Reinforcement Learning

Proof. Since we are looking for the point $w^M$ where $\mathbb{E} \; TP(w^M) = 0$, from proposition E.4.2 we have:

$$\left( \sum_{s \in S} \pi_s \left[ \frac{\phi_s\phi_s^\top}{||\phi_s||^2_2} \right] \right) w^M = \sum_{s \in S} \pi_s \left[ \frac{\phi_s V_s}{||\phi_s||^2_2} \right]$$

From proposition E.4.4, we have that

$$\sum_{s \in S} \pi_s \left[ \frac{\phi_s\phi_s^\top}{||\phi_s||^2_2} \right] = \Phi^\top NDN\Phi.$$ Thus:

$$\left( \sum_{s \in S} \pi_s \left[ \frac{\phi_s\phi_s^\top}{||\phi_s||^2_2} \right] \right) w^M = \Phi^\top NDN\Phi$$

From proposition E.4.3, we have that

$$\sum_{s \in S} \pi_s \left[ \frac{\phi_s\phi_s^\top}{||\phi_s||^2_2} \right] = \Phi^\top NDN\Phi.$$ Thus:

$$\Phi^\top NDN\Phi w^M = \Phi^\top NDN\Phi$$

Multiplying by $[\Phi^\top NDN\Phi]^{-1}$ on both sides:

$$(\Phi^\top NDN\Phi)^{-1} \Phi^\top NDN\Phi w^M = (\Phi^\top NDN\Phi)^{-1} \Phi^\top NDN\Phi$$

To finally get:

$$w^M = \left[ (\Phi^\top NDN\Phi)^{-1} \Phi^\top NDN \right] \Phi$$

(20)

Proposition E.4.2. The convergence point $w^M$ of our algorithm, which is the stable point of the o.d.e that our stochastic update equation tracks, satisfies the condition

$$\left( \sum_{s \in S} \pi_s \left[ \frac{\phi_s\phi_s^\top}{||\phi_s||^2_2} \right] \right) w^M = \sum_{s \in S} \pi_s \left[ \frac{\phi_s V_s}{||\phi_s||^2_2} \right]$$

Proof. We are looking for the point where $h(w(t)) = 0$. In other words, we are looking for a point $w^M$ where $\mathbb{E} \; TP(w^M) = 0$. Then we have:

$$\sum_{s \in S} \pi_s \frac{\phi_s^\top w^M - V_s}{||\phi_s||^2} \phi_s = 0$$

Which we can directly rewrite to:

$$\left( \sum_{s \in S} \pi_s \left[ \frac{\phi_s\phi_s^\top}{||\phi_s||^2_2} \right] \right) w^M = \sum_{s \in S} \pi_s \left[ \frac{\phi_s V_s}{||\phi_s||^2_2} \right]$$

Proposition E.4.3. $\sum_{s \in S} \pi_s \left[ \frac{\phi_s\phi_s^\top}{||\phi_s||^2_2} \right] = \Phi^\top NDN\Phi$
Proof. Note that the LHS and RHS are both matrices of size $n \times n$. We will show the equality explicitly for each $(i,j)$'th entry of this matrix.

For the LHS, the entry at position $(i,j)$ is given by
$$\sum_{s \in S} \pi_s \left( \frac{\phi_s(i)\phi_s(j)}{||\phi_s||_2^2} \right)$$

For the RHS, first note that $NDN$ is a diagonal matrix of size $|S| \times |S|$. The diagonal entries are given by $[NDN]_{(s,s)} = \frac{\pi_s}{||\phi_s||_2^2}$. Then $NDN\Phi$ has $|S|$ rows of the form $\frac{\pi_s}{||\phi_s||_2^2}\phi_s$. Finally, the entry at the $(i,j)$'th location of $\Phi^\top NDN\Phi$, which is a $n \times n$ matrix is given by
$$\sum_{s \in S} \pi_s \left( \frac{\phi_s(i)\phi_s(j)}{||\phi_s||_2^2} \right),$$

which is the same as the LHS.

Proposition E.4.4. $\sum_{s \in S} \pi_s \left[ \frac{\phi_s V_s}{||\phi_s||_2^2} \right] = \Phi^\top NDNV$

Proof. In this case we are dealing with a vector in $\mathbb{R}^n$ for both the LHS and the RHS. We will show equality by showing the $i$'th entry of this vector on both LHS and RHS are the same.

For the LHS, we have a sum of $|S|$ vectors of the form $\left( \frac{\pi_s V_s}{||\phi_s||_2^2} \right)\phi_s$. Then the entry at $i$'th location is given by
$$\sum_{s \in S} \left( \frac{\pi_s V_s}{||\phi_s||_2^2} \right)\phi_s(i)$$

For the RHS, note that $[NDN]_{(s,s)} = \frac{\pi_s}{||\phi_s||_2^2}$ as in proposition E.4.3. Then $NDNV$ is a vector of size $|S|$ where the entry for state $s$ is given as $[NDNV]_s = \frac{\pi_s V_s}{||\phi_s||_2^2}$. Finally, we have that the entry at the $i$'th row $(i \in \{1, \ldots, n\})$ in $\Phi^\top NDNV$ is given by $\frac{\pi_s V_s}{||\phi_s||_2^2}\phi_s(i)$, which is the same as the LHS.

F. Choice of constant momentum multiplier $\beta$

We plot the mean error with iterations for different $\beta$ values to do a comparison between the various constant values. This will enable us to see reasons for our choice of $\beta = 0.5$.

Note that when we increase $\beta$ beyond 0.5, we see non-smoothness in convergence of the stochastic case. Thus we do not go for $\beta > 0.5$ even though it sometimes leads to faster convergence.

![Comparison of error for non-stochastic updates for Constant Momentum using different constants](image1)

![Comparison of error for stochastic updates for Constant Momentum using different constants](image2)

Figure 8. Comparison of different Momentum

We note that in the non-stochastic case, all values of $\beta \in [0, 1)$ lead to convergence. Given enough iterations, we expect the same in the stochastic case as well.
Scale Invariant Solutions for Overdetermined Linear Systems with Applications to Reinforcement Learning

G. Showing Convergence with Momentum

G.1. Problem Setup

Our original stochastic approximation equation with momentum can be written as

\[ w_{k+1} = w_k - \alpha_k \sum_{i=1}^{\tau} \left[ \frac{\phi_i^T w - \bar{V}_i}{||\phi_i||^2} \phi_i \right] + \beta (w_k - w_{k-1}) \]

where the notations have the usual meaning explained in section 3.2 and further, \( \beta \in [0, 1) \). We want to show that this converges, where we have already shown that the update \( w_{k+1} = w_k - \alpha_k \sum_{i=1}^{\tau} \left[ \frac{\phi_i^T w - \bar{V}_i}{||\phi_i||^2} \phi_i \right] \) converges.

Approach used

Traditional algorithms may attempt such a momentum under the two timescale approximation scheme. These have been considered in (Borkar, 2008; Lakshminarayan & Bhatnagar, 2017). Two time scale approximation are also considered in (Avrachenkov et al., 2020) in the context of web page change rate estimation. We take a different approach. First we convert the given stochastic approximation equation with momentum into a two timescale regime, with two iterates getting updated. Then we collapse the second iterate into a perturbation on the first iterate \( w_k \), and thus show convergence. We detail this in the following sections.

G.2. Breaking the stochastic approximation equation with momentum into a two timescale structure:

**Proposition G.2.1.** The update equation \( w_{k+1} = w_k - \alpha_k \sum_{i=1}^{\tau} \left[ \frac{\phi_i^T w - \bar{V}_i}{||\phi_i||^2} \phi_i \right] + \beta (w_k - w_{k-1}) \) can also be written as the set of equations

\[
\begin{align*}
  w_{k+1} - w_k &= \alpha_k z_k \\
  z_0 &= TP_k(w_k) \\
  z_i &= z_{i-1} + \zeta_i TP_{k-i}(w_{k-i}) \quad \forall i \in [1, k]
\end{align*}
\]

where \( \zeta_i = \beta \frac{\alpha_{k-i}}{\alpha_k} \)

**Proof.** Consider:

\[
\begin{align*}
  w_{k+1} &= w_k - \alpha_k \sum_{i=1}^{\tau} \left[ \frac{\phi_i^T w - \bar{V}_i}{||\phi_i||^2} \phi_i \right] + \beta (w_k - w_{k-1}) \\
  \text{(21)}
\end{align*}
\]

Rewriting as a difference:

\[
\begin{align*}
  w_{k+1} - w_k &= \alpha_k \sum_{i=1}^{\tau} \left[ \frac{\phi_i^T w - \bar{V}_i}{||\phi_i||^2} \phi_i \right] + \beta (w_k - w_{k-1}) \\
  \text{(21)}
\end{align*}
\]

We will call the term \( \sum_{i=1}^{\tau} \left[ \frac{\phi_i^T w - \bar{V}_i}{||\phi_i||^2} \phi_i \right] \) as \( TP_k(w_k) \)

\[
\begin{align*}
  w_{k+1} - w_k &= \alpha_k TP_k(w_k) + \beta (w_k - w_{k-1})
\end{align*}
\]
Expanding the momentum term

\[ w_{k+1} - w_k = \alpha_k TP_k(w_k) + \beta(\alpha_{k-1} TP_{k-1}(w_{k-1}) + \beta(w_{k-1} - w_{k-2})) \]
\[ = \alpha_k TP_k(w_k) + \beta \alpha_{k-1} TP_{k-1}(w_{k-1}) + \beta^2 (w_{k-1} - w_{k-2}) \]
\[ = \alpha_k TP_k(w_k) + \beta \alpha_{k-1} TP_{k-1}(w_{k-1}) + \beta^2 (\alpha_{k-2} TP_{k-2}(w_{k-2}) + \beta(w_{k-2} - w_{k-3})) \]
\[ = \ldots \]

Thus we can write the whole thing as:

\[ = \alpha_k TP_k(w_k) + \beta \alpha_{k-1} TP_{k-1}(w_{k-1}) + \beta^2 \alpha_{k-2} TP_{k-2}(w_{k-2}) + \cdots + \beta^k \alpha_0 TP_0(w_0) \] (22)

We note that this is in the form of a discounted sum of vectors, which we have to bring into a form that is the sum of two iterates (Kushner & Yin, 1997).

We reverse the order of the second iterate set. We build \( z_k \) bottom up as follows. Let:

\[ z_0 = TP_k(w_k) \]
\[ z_1 = z_0 + \beta \frac{\alpha_{k-1}}{\alpha_k} TP_{k-1}(w_{k-1}) \]
\[ \ldots = \ldots \]
\[ z_k = z_{k-1} + \beta^k \frac{\alpha_0}{\alpha_k} TP_0(w_0) \]

Further, to simplify this set of equations, we let \( \zeta_{(i,k)} \) be the step size corresponding to \( z_i \) such that \( \zeta_{(i,k)} = \beta^i \frac{\alpha_{k-i}}{\alpha_k} \). Then we have the set of equations as:

\[ w_{k+1} - w_k = \alpha_k z_k \]
\[ z_0 = TP_k(w_k) \]
\[ z_1 = x_0 + \zeta_{(1,k)} TP_{k-1}(w_{k-1}) \]
\[ \ldots = \ldots \]
\[ z_k = z_{k-1} + \zeta_{(k,k)} TP_0(w_0) \]

Or more generally if \( \zeta_{(i,k)} = \beta^i \frac{\alpha_{k-i}}{\alpha_k} \):

\[ w_{k+1} - w_k = \alpha_k z_k \] (23)
\[ z_0 = TP_k(w_k) \]
\[ z_i = z_{i-1} + \zeta_{(i,k)} TP_{k-i}(w_{k-i}) \quad \forall i \in [1, k] \]

G.3. Collapsing the two iterate stochastic approximation equations into a single iterate form:

Now wish to express the above equation in terms of an expected update and a Martingale noise term (with respect to the filtration). For \( z_0 \), such an expression is straightforward: We add and subtract the expectation to change the equation from \( z_0 = TP_k(w_k) \) to

\[ z_0 = \mathbb{E} \left[ TP_k(w_k) \mid \mathcal{F}_k \right] + (TP_k(w_k) - \mathbb{E} \left[ TP_k(w_k) \mid \mathcal{F}_k \right]) \] (24)
where the first term is the expected update term \( H_{(0,k)} = \mathbb{E} \left[ TP_k(w_k) | \mathcal{F}_k \right] \) second term is a martingale difference noise term, \( M_{(0,k)} = (TP_k(w_k) - \mathbb{E} \left[ TP_k(w_k) | \mathcal{F}_k \right] \).

Now let us focus on \( z_i \) for \( i \in [1, k] \)

\[
z_i = z_{i-1} + \zeta_{(i,k)} TP_{k-i}(w_{k-i})
\]

can be rewritten as:

\[
z_i = z_{i-1} + \zeta_{(i,k)} [TP_{k-i}(w_k) + (TP_{k-i}(w_{k-i}) - TP_{k-i}(w_k))]
\]

Which can be further broken down as:

\[
z_i = z_{i-1} + \zeta_{(i,k)} \left[ \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] + (TP_{k-i}(w_k) - \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k]) + (TP_{k-i}(w_{k-i}) - TP_{k-i}(w_k)) \right]
\]

Now we take an expectation of the first term over all possible \( \mathcal{F}_k \). Thus the first term breaks into:

\[
= z_{i-1} + \zeta_{(i,k)} \left[ \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] + (\mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] - \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k]) + (TP_{k-i}(w_{k-i}) - TP_{k-i}(w_k)) \right]
\]

Since \( \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] \) remains unaffected by the filtration, we can write \( \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] = \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] \). For ease of notation, we simply write \( \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] \) as \( \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] \). Then we have:

\[
= z_{i-1} + \zeta_{(i,k)} \left[ \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] + (\mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] - \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k]) + (TP_{k-i}(w_{k-i}) - TP_{k-i}(w_k)) \right]
\]

Notice that the third term above \( TP_{k-i}(w_k) - \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] \) is actually 0 as the filtration provides the exact hyperplanes as well as \( w_k \). Thus the expression is deterministic. Therefore, \( \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] = TP_{k-i}(w_k) \). Thus we finally have

\[
z_i = z_{i-1} + \zeta_{(i,k)} \left[ \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] + (\mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] - \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k]) + (TP_{k-i}(w_{k-i}) - TP_{k-i}(w_k)) \right] = z_{i-1} + \zeta_{(i,k)} (TP_{k-i}(w_k) + \epsilon_{(i,k) - (i,k)}) \]

(25)

Let

\[
H_{(i,k)} = \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k]
\]

\[
M_{(i,k)} = \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] - \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k]
\]

\[
\epsilon_{(i,k)} = TP_{k-i}(w_{k-i}) - TP_{k-i}(w_k)
\]

If \( \zeta_{(i,k)} = \beta^{2 \frac{\alpha_{i-k}}{\alpha_i}} \). Further, let \( h_{(i,k)}(\cdot) \) be some limiting o.d.e that asymptotically tracks \( H_{(i,k)}(\cdot) \). Thus we have the set of equations:

\[
w_{k+1} - w_k = \alpha_k z_k
\]

\[
z_0 = h_{(0,k)} + M_{(0,k)}
\]

\[
z_i = z_{i-1} + \zeta_{(i,k)} (h_{(i,k)} + M_{(i,k)} + \epsilon_{(i,k)}) \quad \forall i \in [1, k]
\]

(26)
We collapse these now into a single equation. Since \( h_{(i,k)}(w_k) = TP(w_k) \forall i, k \) based on proposition G.7.1 We will label this simply as \( h(w_k) \)

Let \( h(w_k) = h(w_k) \left( 1 + \sum_{i=1}^{k} \zeta_{(i,k)} \right) \cdot \hat{M}_k = M_{(0,k)} + \sum_{i=1}^{k} \zeta_{(i,k)} M_{(i,k)} \) and \( \hat{e}_k = \sum_{i=1}^{k} \zeta_{(i,k)} \epsilon_{(i,k)} \). Then:

\[
    w_{k+1} - w_k = \alpha_k [h(w_k) + \hat{e}_k + \hat{M}_k]
\]

(27)

Now we have to show that this single equation follows the requirements for convergence. We will show each of the assumptions in order.

G.4. Showing basic properties of required for convergence:

**Proposition G.4.1.** The step size sequence \( \{\alpha_i\}_{i=1}^{\infty} \) satisfies \( \sum_{i=0}^{\infty} \alpha_i = \infty \) and \( \sum_{i=0}^{\infty} \alpha_i^2 < \infty \)

*Proof.* The step size sequence remains the same as in proposition E.3.1. Thus the proof remains the same. □

**Proposition G.4.2.** Let \( \check{\zeta}_k = 1 + \sum_{i=1}^{k} \zeta_{(i,k)} \). Then \( \check{\zeta}_k \) is bounded.

*Proof.* Consider

\[
    \check{\zeta}_k = \frac{1}{\alpha_k} \left[ \alpha_k + \beta \alpha_{k-1} + \beta^2 \alpha_{k-2} + \ldots \right]
\]

Recall that \( \alpha_k = \eta_k \theta_k \) where \( \eta_k = \frac{1}{k^p} \), \( p \in (0.5, 1] \) and \( \theta_k = \frac{||TP_k(w_k)||}{||\Delta TP_k(w_k)||} \). Further, recall that \( \sup \theta_k = \overline{\theta} \). Then:

\[
    \check{\zeta}_k \leq \frac{\overline{\theta}}{\theta_k} \left[ 1 + \beta \cdot \frac{k}{k-1} + \beta^2 \cdot \frac{k}{k-2} + \ldots \right]
\]

\[
    = \frac{\overline{\theta}}{\theta_k} \left[ 1 + \beta + \frac{\beta}{k-1} + \beta^2 \cdot \frac{2}{k-2} + \beta^3 + \frac{3\beta^3}{k-3} + \ldots \right]
\]

\[
    = \frac{\overline{\theta}}{\theta_k} \left[ 1 + \beta + \beta^2 + \ldots \right] + \left( \frac{\beta}{k-1} + \frac{2\beta^2}{k-2} + \frac{3\beta^3}{k-3} + \ldots \right)
\]

As \( k \to \infty \), the first half of the above expression is \( \frac{\overline{\theta}}{\theta_k(1-\beta)} \). As \( k \to \infty \), the second half converges to 0 (Murthy, 2021). Thus the entire expression remains bounded. □

**Proposition G.4.3.** The expected update for \( w, \hat{h} : \mathbb{R}^n \to \mathbb{R}^n \) is Lipschitz

*Proof.* We have already shown that \( h(\cdot) = TP(w) \) is Lipschitz (as can be seen from the fact that \( \sum_{s \in S} \pi_s \phi_s \phi_{s}^T w + C \) where \( C = \sum_{s \in S} \pi_s \phi_s V_s \) is linear in \( w \)). Now we will show that \( \hat{h}(\cdot) = h(\cdot)(1 + \sum_{i=1}^{k} \zeta_{i}) \) is also Lipschitz. But we have shown that \( \check{\zeta}_k = 1 + \sum_{i=1}^{k} \zeta_{(i,k)} \) is bounded in proposition G.4.2.

Thus we have that if \( h(\cdot) \) is Lipschitz, then \( \check{\zeta}_k h(\cdot) = \check{\hat{h}}(\cdot) \) is also Lipschitz for some constant \( \check{\zeta}_k \). □
G.5. Showing that the noise term is a martingale difference sequence:

**Proposition G.5.1.** We specifically consider $M_{(0,k)}$ first. $\mathbb{E}[M_{(0,k)} | \mathcal{F}_k] = 0$ and $\mathbb{E} [||M_{(0,k)}||^2 | \mathcal{F}_k] \leq K[1 + ||w_k||^2]

**Proof.** We note that $M_{(0,k)} = TP_k(w_k) - \mathbb{E} [TP_k(w_k) | \mathcal{F}_k] = TP_k(w_k) - TP(w_k)$ as per appendix section E.2.

Now we have already shown in appendix section E.3.3 that $\mathbb{E}[TP_k(w_k) - TP(w_k)] \leq K[1 + ||w_k||^2]$ for some $K \in \mathbb{R}$.

For the second part, note that the filtration gives us the hyperplanes, say $E$.

Further we also showed $\exists A_{(0,k)}, C_{(0,k)}$ such that $M_{(0,k)} = A_{(0,k)} w_k - C_{(0,k)}$ whence $\mathbb{E} [||TP_k(w_k) - TP(w_k)||^2 | \mathcal{F}_k] \leq K[1 + ||w_k||^2]$ for some $K \in \mathbb{R}$. 

$\blacksquare$

**Proposition G.5.2.** $\mathbb{E}[M_{(i,k)} | \mathcal{F}_k] = 0$ and $\mathbb{E} [||M_{(0,k)}||^2 | \mathcal{F}_k] \leq K[1 + ||w_k||^2]

**Proof.** First we note that

$$M_{(i,k)} = \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] - \mathbb{E}_{F_k}[\mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k]]$$

Note that the second expectation remains unchanged given the filtration, thus we can rewrite this as:

$$= \mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] - \mathbb{E}_{F_k}[\mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] | \mathcal{F}_k]$$

Given such a definition, $\mathbb{E}[M_{(i,k)} | \mathcal{F}_k] = \mathbb{E}_{F_k}[\mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] | \mathcal{F}_k] - \mathbb{E}_{F_k}[\mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] | \mathcal{F}_k] = 0$

For the second part, note that the filtration gives us the hyperplanes, say $\{1, \ldots \}$ that have been sampled. Then:

$$\mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] = \left[\frac{1}{\tau} \sum_{i=1}^{\tau} \phi_i^T w_k - \bar{V}_i \phi_i \right]$$

We obtain from appendix proposition G.7.1 that $\mathbb{E}[TP_{k-i}(w_k) | \mathcal{F}_k] = TP(w_k) = \sum_{s \in \mathcal{S}} \pi_s \left[ \phi_s^T w - V_s \phi_s \right]$. Therefore:

$$M_{(i,k)} = \left[\frac{1}{\tau} \sum_{i=1}^{\tau} \frac{\phi_i^T w_k - \bar{V}_i}{||\phi_i||^2} \phi_i \right] - \sum_{s \in \mathcal{S}} \pi_s \left[ \frac{\phi_s^T w - V_s}{||\phi_s||^2} \phi_s \right]$$

$$= \left[\frac{1}{\tau} \sum_{i=1}^{\tau} \frac{\phi_i^T w_k - \bar{V}_i}{||\phi_i||^2} \phi_i \right] - \sum_{s \in \mathcal{S}} \pi_s \left[ \frac{\phi_s^T w_k - \bar{V}_i \phi_i}{||\phi_s||^2} \phi_s \right]$$

$$= A_{(i,k)} w_k - C_{(i,k)}$$

which is linear in $w_k$ with bounded coefficients. Further note that $A_{(i,k)}$ is bounded above as $\frac{\phi_i^T \phi_i}{||\phi_i||^2}$ has maximum eigenvalue 1. Further, $C_{(i,k)}$ is bounded as $\bar{V}$ is bounded above by $\frac{R_{max}}{1 - \gamma}$ where $R_{max}$ is the maximum reward and $\gamma$ is the discounting factor.

Thus $||M_{(i,k)}||^2$ is quadratic in $w_k$. Now it is easy to see that there would exist some $K$ such that $\mathbb{E} [||M_{(i,k)}||^2 | \mathcal{F}_k] \leq K(1 + ||w_k||^2)$

$\blacksquare$

**Proposition G.5.3.** Consider the filtration $\mathcal{F}_k = \{w_0, \ldots, w_k\}$. Then the sequence $\{\mathcal{M}_k\}$ is a zero-mean martingale difference noise sequence. Specifically, we have that:
1. \( \mathbb{E}[\hat{M}_k | F_k] = 0 \)

2. \( \mathbb{E}[\|\hat{M}_k\|^2 | F_k] \leq K_i (1 + \|w_k\|^2) \)

**Proof.** For the first part, we need to show \( \mathbb{E}[\hat{M}_k | F_k] = 0 \) where \( \hat{M}_k = M_{(0,k)} + \sum_{i=1}^{k} \zeta_{(i,k)} M_{(i,k)} \). We have:

\[
\mathbb{E}[\hat{M}_k | F_k] = \mathbb{E}[M_{(0,k)} + \sum_{i=1}^{k} \zeta_{(i,k)} M_{(i,k)} | F_k]
\]

By linearity of expectation:

\[
= \mathbb{E}[M_{(0,k)} | F_k] + \sum_{i=1}^{k} \zeta_{(i,k)} \mathbb{E}[M_{(i,k)} | F_k]
\]

But we have from proposition G.5.2 that \( \mathbb{E}[M_{(i,k)} | F_k] = 0 \) and from proposition G.5.1 that \( \mathbb{E}[M_{(0,k)} | F_k] = 0 \). Therefore

\[
\mathbb{E}[\hat{M}_k | F_k] = 0 + \sum_{i=1}^{k} \zeta_{(i,k)} 0
\]

\[
= 0
\]

For the second part, we see this by linearity.

\( \hat{M}_k = M_{(0,k)} + \sum_{i=1}^{k} \zeta_{(i,k)} M_{(i,k)} \)

From propositions G.5.1 and G.5.2, we can write the above as:

\[
= (A_{(0,k)} + \sum_{i=1}^{k} \zeta_{(i,k)} A_{(i,k)}) w_k - (C_{(0,k)} + \sum_{i=1}^{k} \zeta_{(i,k)} C_{(i,k)})
\]

Since \( A_{(i,k)}, C_{(i,k)} \) are bounded \( \forall i \in \{0,\ldots,k\} \), we can write the above as:

\( \hat{M}_k = \hat{A} w_k - \hat{C} \)

where \( \hat{A} = A_{(0,k)} + \sum_{i=1}^{k} \zeta_{(i,k)} A_{(i,k)} \) is bounded and \( \hat{C} = C_{(0,k)} + \sum_{i=1}^{k} \zeta_{(i,k)} C_{(i,k)} \) is bounded. Now we see that \( \hat{M}_k \) is linear in \( w_k \) with bounded coefficients.

Thus \( \|\hat{M}_k\|^2 \) is quadratic in \( w_k \), whence \( \exists K \in \mathbb{R} \) such that \( \mathbb{E} \left[ \|\hat{M}_k\|^2 | F_k \right] \leq K (1 + \|w_k\|^2) \)

\( \Box \)

**G.6. Showing that the momentum terms sum to a perturbation:**

**Proposition G.6.1** (Helper proposition for G.6.2). \( \sum_{i=1}^{k} \zeta_i \cdot ||w_k - w_{k-i}|| \to 0 \) as \( k \to \infty \)

**Proof.** \( \sum_{i=1}^{\infty} \zeta_i \cdot ||w_k - w_{k-i}|| = \sum_{i=1}^{m} \zeta_i \cdot ||w_k - w_{k-i}|| + \sum_{i=m+1}^{\infty} \zeta_i \cdot ||w_k - w_{k-i}|| \).
Now given any $\epsilon$, there exists $m$ such that $\sum_{i=m+1}^{\infty} \zeta_i \cdot ||w_k - w_{k-i}|| < \epsilon$. This is because $\zeta_i \sim \beta_i$ go to 0 and $||w_k - w_{k-i}||$ are bounded (shown separately when we show stability of iterates).

Then given any finite $m$, at the asymptote as $k \to \infty$, we have $||w_k - w_{k-i}|| \to 0$ as $\alpha_k \to 0$. Thus $\sum_{i=1}^{k} \zeta_i \cdot ||w_k - w_{k-i}|| < \epsilon$ as $k \to \infty$ for any arbitrary $\epsilon$.

Thus $\sum_{i=1}^{k} \zeta_i \cdot ||w_k - w_{k-i}|| \downarrow 0$ as $k \to \infty$.

**Proposition G.6.2.** $\hat{\epsilon}_k$ are perturbation terms that satisfy $||\hat{\epsilon}_k|| \leq d_k (1 + ||w_k||)$ where $d_k$ are a sequence of positive scalars such that $\lim_{k \to \infty} d_k = 0$.

**Proof.** First note that $\hat{\epsilon}_k = \sum_{i=1}^{k} \zeta_i \epsilon_{(i,k)}$ and $\epsilon_{(i,k)} = TP_{k-i}(w_{k-i}) - TP_{k-i}(w_k)$. Thus

$$\hat{\epsilon}_k = \sum_{i=1}^{k} \zeta_i \epsilon_{i}$$

Expanding $\epsilon_i$ using the inequality in (28)

$$||\hat{\epsilon}_k|| \leq d_k (1 + ||w_k||)$$

Now we note that asymptotically as $k \to \infty$, we have for finite $i$, $||w_k - w_{k-i}|| \to 0$ and for large $i$, $\zeta_i \to 0$. Thus by proposition G.6.1 the above is bounded above by some arbitrary $\epsilon$.

Thus asymptotically we see that this perturbation term is $o(1)$.

**G.7. Stability Criterion: Iterates remain bounded**

In (Borkar, 2008), we have to prove that the iterates of in the update equation remain bounded. (Lakshminarayanan & Bhatnagar, 2017) have provided a stability criterion to ensure that the iterates remain bounded. While we have already shown that the iterates on after the expected update remain bounded in proposition 3.4, here we will explicitly show the stability criterion is satisfied.

But first a basic proposition:
Proposition G.7.1. \( \mathbb{E} \left[ TP_{k-1}(w_k) | \mathcal{F}_k \right] = TP(w_k) \)

Proof. Note that we are considering the expectation over all possible filtrations. Note that the random variables under consideration are \( \tau \) - the number of hyperplanes sampled, \( i \in \{1, \ldots, \tau\} \) - the set of hyperplanes sampled, and \( \tilde{V} \) - the value function. The filtration gives us \( w_k \) and the set of hyperplanes chosen in a particular trajectory. Let’s label the unique hyperplanes in the trajectory by \( \{1, \ldots, \tau\} \). Then:

\[
\mathbb{E}_{\mathcal{F}_k} \left[ \mathbb{E} \left[ TP_{k-1}(w_k) | \mathcal{F}_k \right] \right] = \mathbb{E}_{\mathcal{F}_k = (\tau, i, \tilde{V})} \left( \frac{1}{\tau} \sum_{i=1}^{\tau} \left[ \frac{\phi_i^\top w_k - \tilde{V}_i}{||\phi_i||^2} \phi_i \right] \right)
\]

By linearity we rewrite this as:

\[
= \mathbb{E}_\tau \left( \frac{1}{\tau} \sum_{i=1}^{\tau} \mathbb{E}_{(i, \tilde{V})} \left[ \frac{\phi_i^\top w_k - \tilde{V}_i}{||\phi_i||^2} \phi_i \right] \right)
\]

Over all possible filtrations, we can write the expectation of the inner term as:

\[
\mathbb{E}_{(i, \tilde{V})} \left[ \frac{\phi_i^\top w_k - \tilde{V}_i}{||\phi_i||^2} \phi_i \right] = \sum_{s \in \mathcal{S}} \pi_s \frac{\phi_s^\top w_k - V_s}{||\phi_s||^2} \phi_s
\]

Substituting this in the previous expression, we get:

\[
\mathbb{E}_{\mathcal{F}_k} \left[ \mathbb{E} \left[ TP_{k-1}(w_k) | \mathcal{F}_k \right] \right] = \mathbb{E}_\tau \left( \frac{1}{\tau} \sum_{i=1}^{\tau} \sum_{s \in \mathcal{S}} \pi_s \frac{\phi_s^\top w_k - V_s}{||\phi_s||^2} \phi_s \right)
\]

But the inner expression is now independent of \( \tau \). Thus:

\[
\mathbb{E}_{\mathcal{F}_k} \left[ \mathbb{E} \left[ TP_{k-1}(w_k) | \mathcal{F}_k \right] \right] = \left[ \sum_{s \in \mathcal{S}} \pi_s \phi_s^\top w_k - V_s \right] \frac{\phi_s}{||\phi_s||^2}
\]

We note that the RHS is \( TP(w_k) \)

\( \blacksquare \)

Proposition G.7.2. Let us define the sequence of functions \( \hat{h}_c(w) : \mathbb{R}^n \mapsto \mathbb{R}^n \) such that \( \hat{h}_c(w) = \frac{\hat{h}(cw)}{c} \); \( c \geq 1 \). Then

1. \( \hat{h}_c(\cdot) \to \hat{h}_\infty(\cdot) \) as \( c \to \infty \) uniformly on compact sets Further,

2. The limiting o.d.e, \( \dot{w}(t) = \hat{h}_\infty(w(t)) \) has a unique globally asymptotically stable equilibrium at the origin.

Proof. First note that

\[
\hat{h}(w) = h(w) + \sum_{i=1}^{k} \zeta_i
\]

But \( \dot{\zeta} = (1 + \sum_{i=1}^{k} \zeta_i) \). Then:

\[
= \dot{\zeta} h(w)
\]
Expanding $h(w)$:

$$
= \hat{\zeta} \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T w - V_s}{||\phi_s||^2} \right] \phi_s
$$

Now we write $\hat{h}_c(w)$ from its definition:

$$
\hat{h}_c(w) = \hat{\zeta} \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T c w - V_s}{c||\phi_s||^2} \right] \phi_s
$$

But the constants $c$ can be cancelled for the $w$ term:

$$
= \hat{\zeta} \left( \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T \phi_s}{||\phi_s||^2} \right] w - \frac{1}{c} \sum_{s \in S} \pi_s \left[ \frac{V_s \phi_s}{||\phi_s||^2} \right] \right)
$$

We observe the uniform convergence of this set of functions $\hat{h}_c(w)$ to $\hat{h}_\infty(w)$ in the limit $c \to \infty$ as the term $c$ is only involved with a constant coefficient given by $\sum_{s \in S} \pi_s \left[ \frac{V_s \phi_s}{||\phi_s||^2} \right]$. Thus the first part is proved.

For the second part, we note the following:

$$
\hat{h}_\infty(w) = \lim_{c \to \infty} \hat{\zeta} \left( \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T \phi_s}{||\phi_s||^2} \right] w - \frac{1}{c} \sum_{s \in S} \pi_s \left[ \frac{V_s \phi_s}{||\phi_s||^2} \right] \right)
$$

Now we apply the limit only on the second term:

$$
= \hat{\zeta} \left( \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T \phi_s}{||\phi_s||^2} \right] w - \lim_{c \to \infty} \frac{\hat{\zeta}}{c} \sum_{s \in S} \pi_s \left[ \frac{V_s \phi_s}{||\phi_s||^2} \right] \right)
$$

Evaluating the limit, we get 0 for the second term:

$$
= \hat{\zeta} \left( \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T \phi_s}{||\phi_s||^2} \right] w \right)
$$

Thus:

$$
\hat{h}_\infty(w) = \hat{\zeta} \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T \phi_s}{||\phi_s||^2} \right] w
$$

Now consider the system $\dot{w}(t) = \hat{\hat{h}}_\infty(w(t)) = \hat{\zeta} \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T \phi_s}{||\phi_s||^2} \right] w$. At the equilibrium point,

$$
\dot{w}(t) = 0
$$

$$
\hat{\zeta} \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T \phi_s}{||\phi_s||^2} \right] w = 0
$$

But $\Phi$ has full column rank (by assumption). Thus no eigen value of $\sum_{s \in S} \pi_s \left[ \frac{\phi_s^T \phi_s}{||\phi_s||^2} \right]$ is 0. Thus:

$$
w = 0
$$

Thus $\dot{w}(t) = \hat{h}_\infty(w(t))$ has a unique globally asymptotically stable equilibrium at the origin.
G.8. The stochastic update equation with momentum converges:

In section G.4, we showed the assumptions A1 and A2 required for convergence. In section G.5 we showed that the noise term is a martingale difference sequence - assumption A3 (per (Borkar, 2008)). In section G.7, we showed assumption A4, which was the stability criterion required to show that the iterates remain bounded (Lakshminarayanan & Bhatnagar, 2017). Finally, in section G.6, we showed that the momentum terms added a perturbation term to the o.d.e that we are asymptotically tracking.

All that is left to see is where we converge to.

**Proposition G.8.1.** The globally asymptotically stable equilibrium for the limiting o.d.e \( \dot{w}(t) = \hat{h}(w(t)) \) that our stochastic approximation equation tracks is given by

\[
w^M = \left( (\Phi^T NDN\Phi)^{-1} \Phi^T NDN \right) V
\]

**Proof.** The update equation, \( \dot{w}(t) = \hat{h}(w(t)) \) can be written as \( \dot{w}(t) = \hat{\zeta}h(w(t)) = \hat{\zeta} \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T w - V_s}{||\phi_s||^2} \right] \phi_s \). Considering that the states are sampled from the stationary distribution \( \pi \), we have:

\[
\dot{w}(t) = \hat{\zeta} \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T w - V_s}{||\phi_s||^2} \right] \phi_s = 0
\]

Then the equilibrium point is given by the point where:

\[
\dot{w}(t) = 0
\]

\[
\hat{\zeta} \sum_{s \in S} \pi_s \left[ \frac{\phi_s^T w - V_s}{||\phi_s||^2} \right] \phi_s = 0
\]

But \( \hat{\zeta} \) is just a constant. Therefore:

\[
\sum_{s \in S} \pi_s \left[ \frac{\phi_s^T w - V_s}{||\phi_s||^2} \right] \phi_s = 0 \tag{29}
\]

But this is an equation that we have already solved in section E.4. The solution is given by

\[
w^M = \left( (\Phi^T NDN\Phi)^{-1} \Phi^T NDN \right) V
\]

We’ve now satisfied all the criteria and also shown the point to which we converge. Thus we show the convergence for the full algorithm with momentum.

**H. Normalized TD(0) Problem Setup:**

**H.1. Notations and Problem Setup:**

We will continue with the same notations that we have been using in the rest of the paper. From equation 4.5 in section 4.5, we have that \( w_{k+1} = w_k - \frac{1}{\hat{\zeta}} \left[ \sum_{i=1}^{T} (L_i^T w_k - \rho_i) L_i \right] + \beta (w_k - w_{k-1}) \) where each term \( i \) samples a pair \( (s, s') \).

\[
L_{ss'} = \frac{\Delta \phi_{ss'}}{||\Delta \phi_{ss'}||_2}, \quad \rho_{ss'} = \frac{R_{ss'}}{||\Delta \phi_{ss'}||_2} \quad \text{and} \quad \Delta \phi_{ss'} = (\phi_s - \gamma \phi_{s'}).
\]

We sample state \( s \) from the stationary distribution \( \pi \), with corresponding diagonal matrix \( D \), and the state \( s' \) given \( s \) from transition matrix \( P_s \).
H.2. Stable point of the Normalized TD(0):

**Proposition H.2.1.** The stochastic update equation \( w_{k+1} = w_k - \frac{\beta}{2} \left[ \sum_{i=1}^{T} (L_i^T w_k - \rho_i) L_i \right] \) converges to \( w^N \) where \( \left[ \sum_{s \in S} D_{\mu}(s) \left( \sum_{s' \in S} P_{ss'} L_{ss'} L_{ss'}^T \right) \right] w^N = \sum_{s \in S} D_{\mu}(s) \left( \sum_{s' \in S} P_{ss'} \rho_{ss'} L_{ss'} \right) \)

**Proof.** The stochastic update equation given, can be considered without momentum, as we have already shown that the update equation with momentum converges to the same point as the update equation without momentum in appendix G. Therefore consider the stochastic equation \( w_{k+1} = w_k - \frac{\beta}{2} \left[ \sum_{i=1}^{T} (L_i^T w_k - \rho_i) L_i \right] \).

The expected update equation \( \dot{w}(t) = h(w(t)) \) which the above equation tracks can be written as \( h(w) = \mathbb{E}_{(s, s') \in S} \left[ (L_{ss'} w - \rho_{ss'}) L_{ss'} \right] \). Since we want the stable point \( w^N \), we can set this expression equal to 0 at \( w^N \). Thus we have:

\[
\dot{w}(t) = 0 \quad (30)
\]

\[
\mathbb{E}_{(s, s') \in S} \left[ (L_{ss'} w^N - \rho_{ss'}) L_{ss'} \right] = 0 \quad (31)
\]

We now convert the expectation into the probability weighted sums:

\[
\sum_{s, s' \in S} \left[ (L_{ss'} w^N - \rho_{ss'}) L_{ss'} P(S = s, S' = s') \right] = 0 \quad (32)
\]

From Bayes, the above converts to:

\[
\sum_{s, s' \in S} \left[ (L_{ss'} w^N - \rho_{ss'}) L_{ss'} P(S = s) P(S' = s'|S = s) \right] = 0 \quad (33)
\]

But we have \( P(S = s) = D_{\mu}(s) \) as we sample \( s \) from the stationary distribution. Further, \( P(S' = s'|S = s) \) is obtained from the row of the transition matrix \( P \) corresponding to state \( s \). We then have:

\[
\sum_{s, s' \in S} \left[ (L_{ss'} w^N - \rho_{ss'}) L_{ss'} D_{\mu}(s) P_{ss'} \right] = 0 \quad (34)
\]

We can rearrange the terms:

\[
\sum_{s, s' \in S} D_{\mu}(s) P_{ss'} \left[ L_{ss'} (L_{ss'} w^N - \rho_{ss'}) \right] = 0 \quad (35)
\]

\[
\sum_{s, s' \in S} D_{\mu}(s) P_{ss'} \left[ L_{ss'} L_{ss'}^T w^N - \rho_{ss'} L_{ss'} \right] = 0 \quad (36)
\]

moving some terms to the right, and taking \( w^N \) outside the summation:

\[
\left[ \sum_{s, s' \in S} D_{\mu}(s) P_{ss'} L_{ss'} L_{ss'}^T \right] w^N = \sum_{s, s' \in S} D_{\mu}(s) P_{ss'} \rho_{ss'} L_{ss'} \quad (37)
\]

We can express this as a double sum, to get a simpler form inside the summation

\[
\left[ \sum_{s \in S} D_{\mu}(s) \left( \sum_{s' \in S} P_{ss'} L_{ss'} L_{ss'}^T \right) \right] w^N = \sum_{s \in S} D_{\mu}(s) \left( \sum_{s' \in S} P_{ss'} \rho_{ss'} L_{ss'} \right) \quad (38)
\]
H.3. Description of Tensors used for our expressions

In this section we will use the mathematical framework laid down in appendix I.

If we look at the components of equation (38), we note that while the others can be thought of as vectors or matrices, \( \mathcal{L} \) stands out naturally as a tensor, as it contains a vector for each state, next state pair, i.e. there’s some vector \( L_{ss'} \) for each \((s, s') \in \mathcal{S}\). Now, we can also think of this above tensor as slices along each state, i.e., we can think of one matrix per state \( s \in \mathcal{S} \) consisting of vectors for each \( s' \in \mathcal{S} \). These matrices can be thought of as slices of the tensor. We visualize these in Figure 9.

We can view the reward tensor (matrix), broken down along each state in Figure 10a.

The probability Tensor is not as straightforward. We can’t simply use the matrix as in the reward tensor. Consider the inner sum in the RHS of 38, \( \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} \rho_{ss'} L_{ss'} \). This is an inner sum of three values, where two are scalars (\( \mathcal{P}_{ss'} \), \( \rho_{ss'} \)) and one is a vector, viz. \( L_{ss'} \). In other words, if we fix \( s \), we find that \( \mathcal{P}_s \) and \( \mathcal{R}_s \) are vectors and \( \mathcal{L}_s \) is a matrix. To multiply these through matrix multiplication operations, we need to transform \( \mathcal{P}_s \) into a diagonal matrix, such that \( \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} \rho_{ss'} L_{ss'} = \mathcal{L}^{\top} \mathcal{P}_s \mathcal{R}_s \).

Since \( \mathcal{P}_s \) is a diagonal matrix, we need \( \mathcal{P} \) to be an order-3 tensor, where each stack is a diagonal matrix. We visualize this in Figure 10b.

Lastly, we make an adjustment in the order of indices for \( \mathcal{L} \in \mathbb{R}^{m \times m \times n} \). From the discussion on transpositions, we have that \( \mathcal{L}^{\top} \in \mathbb{R}^{m \times n \times m} \).

H.4. Expressing \( w^M \) in terms of Tensors

Proposition H.4.1. \( w^M = \left[ (\mathcal{L}^{\top} \times \mathcal{P} \times \mathcal{L}) \times_1 D \right]^{-1} (\mathcal{L}^{\top} \times \mathcal{P} \times \mathcal{R})^{\top} D \)

Proof. First we will consider the RHS of equation 38. We want to express \( \sum_{s \in \mathcal{S}} D_{\mu}(s) \left( \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} \rho_{ss'} L_{ss'} \right) \) succinctly to perform mathematical operations on it.

With all the tensor components in place from section H.3, we’re ready to define the summation in the RHS of 38 in any of the following ways, to yield column vectors on the right.
\[
\sum_{s \in S} D_\mu(s) \left( \sum_{s' \in S} P_{ss'} \rho_{ss'} L_{ss'}^\top \right) = (L^\top \times P \times R)^\top D
\]
\[
= ( (L^\top \times P \times R) \times_1 D )^\top
\]
\[
= (L^\top \times P \times R)^\top \times_2 D
\]

Consider now the LHS of 38,
\[
\left[ \sum_{s \in S} D_\mu(s) \left( \sum_{s' \in S} P_{ss'} L_{ss'}^\top \right) \right] w^M
\]

With the machinery we’ve developed, it’s possible to express this as follows
\[
\left[ \sum_{s \in S} D_\mu(s) \left( \sum_{s' \in S} P_{ss'} L_{ss'}^\top \right) \right] w^M = \left[ (L^\top \times P \times L) \times_1 D \right] w^M
\]

But now we can solve for \( w^M \) since \( \left[ (L^\top \times P \times L) \times_1 D \right] \) is just a matrix and is (almost surely) invertible. Thus,
\[
\begin{align*}
    w^M &= (L^\top \times P \times L)^{-1} (L^\top \times P \times R)^\top D
\end{align*}
\]

Proposition H.4.2.
\[
\sum_{s \in S} D_\mu(s) \left( \sum_{s' \in S} P_{ss'} \rho_{ss'} L_{ss'}^\top \right) = (L^\top \times P \times R)^\top D
\]

Proof. Let’s fix \( s \in S \). Then consider \( \left( \sum_{s' \in S} P_{ss'} \rho_{ss'} L_{ss'}^\top \right) \). As inputs to construct this expression, we have \( L_s - \) a matrix, \( P_s - \) a diagonal matrix and \( R_s - \) a vector.

The product \( L_s \times P_s \times R_s \) gives us the vector we need in a straightforward manner, i.e. \( \sum_{s' \in S} P_{ss'} \rho_{ss'} L_{ss'}^\top = L_s \times P_s \times R_s \)

We need such products for each state, and this is where our tensor product definitions are handy. \( L \times P \times R \) gives us a tensor of size \( m \times m \), where the first dimension corresponds to \( s \), and the second dimension corresponds to the vector \( \sum_{s' \in S} P_{ss'} \rho_{ss'} L_{ss'}^\top \) with \( s \) fixed.

Now all we need to do is to multiply each of these vectors with the corresponding weights from \( D \).

Proposition H.4.3.
\[
\left[ \sum_{s \in S} D_\mu(s) \left( \sum_{s' \in S} P_{ss'} L_{ss'}^\top \right) \right] w^M = \left[ (L^\top \times P \times L) \times_1 D \right] w^M
\]

I. Defining New Tensor Products

In this section, we will define a couple of new tensor products which ease operations on Tensors.

I.1. Background on Tensors

For the purposes of this section, we understand tensors as a multidimensional array. The number of dimensions of this array is called the Order of a tensor. Each dimension is called a mode. (Kolda & Bader, 2009). Consider a tensor \( A \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N} \). The Cardinality of \( A \) is \( I_1 \) along mode-1, \( I_2 \) along mode-2, \ldots \( I_N \) along mode-N. We visualize these in Figure 11a.

A tensor may be thought of as made up of vectors. Each vector in any of these dimensions is termed a fiber. Fibers are the higher-order analogue of matrix rows and columns (Kolda & Bader, 2009). A mode-\( n \) fiber is a fiber oriented along mode-\( n \).
in a tensor. We can also view tensors as made up of matrices. These can be thought of as slices of the tensor. These are visualized in Figure 11b and 11c

These above notations will form the basis of our exploration of new tensor products. We first consider two existing type of products in tensor literature - the mode-n Vector multiplication (Kolda & Bader, 2009) and the mode-n matrix multiplication (Bader & Kolda, 2006)(Lathauwer et al., 2000).

In the mode-n vector multiplication, we multiply each dot product each fiber in the Tensor with one particular vector (of the same cardinality as the mode of the tensor we are multiplying with). Note that we are multiplying (by dot product) each fiber of the whole Tensor with a single vector.

Another product defined in literature is the mode-n matrix multiplication. Here, we multiply each fiber in the Tensor with one particular matrix (of appropriate size). Note that we are multiplying each fiber of the whole Tensor with a single matrix.

I.2. Mode Transforming Tensor Product

This immediately leads to an extension, which we call the mode-transforming tensor product. What if we didn’t have a single matrix to multiply the whole tensor with, but instead had a different matrix for each fiber? This leads to a tensor-tensor multiplication, which we formally define below.

**Definition 1.** The *mode transforming product* of a tensor $A \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $T \in \mathbb{R}^{I_1 \times \cdots \times I_N \times J_{N+1}}$ is a tensor $B \in \mathbb{R}^{I_1 \times \cdots \times I_{N-1} \times J_{N+1}}$, denoted as

$$B = A \hat{\times} T$$

where each vector of $B$ is defined as the product of corresponding entries in $A$ and $T$:

$$B(i_1, i_2, \ldots, i_{N-1}) = A(i_1, i_2, \ldots, i_{N-1}) \times T(i_1, i_2, \ldots, i_{N-1})$$

where $\times$ denotes regular vector-matrix multiplication and:

$$A(i_1, i_2, \ldots, i_{N-1}) \in \mathbb{R}^{I_N}$$

$$T(i_1, i_2, \ldots, i_{N-1}) \in \mathbb{R}^{I_N \times J_{N+1}}$$
I.2.1. Special Case Where $J_{N+1} = 1$

We note that in regular multiplication using operator $\times$, the order of the output tensor would be the same as the input tensor. But we could have a contraction of order in the case when $J_{N+1} = 1$. In this case the output $B(i_1, i_2, \ldots, i_{N-1})$ is just a scalar equal to the dot product $A(i_1, i_2, \ldots, i_{N-1}) \cdot T(i_1, i_2, \ldots, i_{N-1})$. This is visualized in Figure 12. Note that the two tensors need to be of the same size.

![Figure 12](image)

**Figure 12.** Mode Transforming Tensor Product for special case where $J_{N+1} = 1$

Now we can move to the full general case of the slice transforming tensor product.

I.3. Slice Transforming Tensor Product

In similar vein as in the previous section, consider the case where we break the tensor up into slices. Now it is possible to transform each slice with the same matrix. Such a matrix-tensor product would transform each slice into a new slice. Note that this leads to an extension as well.

What if we allowed each slice of a tensor, to be multiplied with a different tensor. This directly leads to what we call a slice-transforming tensor product and is a tensor-tensor multiplication. Formally, we write

**Definition 2.** The slice transforming product of a tensor $A \in \mathbb{R}^{I_1 \times \cdots \times I_{N-2} \times I_{N-1} \times I_N}$ and a transforming tensor $T \in \mathbb{R}^{I_1 \times \cdots \times I_{N-2} \times I_N \times J_N}$ is a tensor $B \in \mathbb{R}^{I_1 \times \cdots \times I_{N-1} \times J_N}$, denoted as

$$B = A \hat{\times} T$$

(44)

where each slice of $B$ is defined as the product of corresponding slices in $A$ and $T$:

$$B(i_1, i_2, \ldots, i_{N-2}) = A(i_1, i_2, \ldots, i_{N-2}) \times T(i_1, i_2, \ldots, i_{N-2})$$

(45)

where $\times$ denotes regular matrix-matrix multiplication and:

$$A(i_1, i_2, \ldots, i_{N-2}) \in \mathbb{R}^{I_{N-1} \times I_N}$$

$$T(i_1, i_2, \ldots, i_{N-2}) \in \mathbb{R}^{I_N \times J_N}$$

I.4. Contraction Products:

In the previous definition, think of the special case where $J_N = 1$. In such a case, the dimension of the output tensor $B$ is one lower than the input tensor $A$. This is because $A(i_1, i_2, \ldots, i_{N-2}) \times T(i_1, i_2, \ldots, i_{N-2})$ is effectively a matrix-vector product.

In such cases, we specifically denote the output as a dimensional contraction. We denote dimensional contractions by using the symbol $\hat{\times}$ instead of $\times$. Thus when $J_N = 1$, we might say $B = A \hat{\times} T$ to denote a contractive product.
I.5. Multiplication Modes

In line with existing tensor literature (Kolda & Bader, 2009), we also provide various modes of multiplication. Take the case of a tensor-vector multiplication, where we want to multiply the vector in the $p$-th mode of the tensor. We specify this as $(A \times_p v)$ for tensor $A$ and vector $v$. Similarly, we can do a mode-transforming tensor product in any given mode. Here for the $p$-th mode of multiplication between tensors $A$ and $T$, we write $A \tilde{\times}_p T$

This is not very convenient for slice-transforming products, thus we provide an additional tool as below.

I.6. Transpositions:

To get the indices of the input tensor in appropriate order for multiplication, we may need to permute them. Note that a permutation of a set is formally an ordering of the elements of the set.

But defining the full permutation each time is cumbersome. An easier way to represent permutations is by use of transpositions. Note that a transposition is a permutation which exchanges two indices and keeps all others fixed.

It can be shown that any permutation is a product of transpositions and the default permutation is the identity permutation.

Therefore, to denote the rearrangement of the indices of an array, we define a rearrangement $R$, where $R$ is of the form $\{(i_1, i_2), \ldots, (i_{k-1}, i_k)\}$ where $\{(i_1, i_2), \ldots, (i_{k-1}, i_k)\}$ are transpositions of indices $(i_1$ and $i_2), \ldots, (i_{k-1}$ and $i_k)$. Then for a tensor $A$, $A^R$ is used to denote the rearranged tensor.

In the special case of interchanging the last two indices of a tensor $A$ of order $N$, we use the notation $A^\top$ to denote $A^{(N-1,N)}$

J. Bounds on TP for TD(0) algorithm

J.1. Notations and Setup

Notation: We will be denote the element-wise product (a.k.a Hadamard product) of two matrices $A$ and $B$ as $A \odot B$. We will denote the matrix of ones of size $m \times n$ as $1_{m \times n}$

Recall our notations from the other sections. We will denote our matrix representing $R_{ss'}$ for all pairs $(s, s') \in S$ as $R$. We will denote our transition matrix of size $|S| \times |S|$ as $P$. Let $m = |S|$. The value vector $V \in \mathbb{R}^n$ denotes the actual value of the states. $V^L$ is the least squares solution obtained by the standard Monte Carlo method. $V^N$ is the value vector output by the Normalized TD(0) method. Finally, let $V^T$ be the value vector obtained by the regular TD(0) algorithm.

Let $N$ be the normalizer matrix. In other words $N_{ss'} = \frac{1}{||\phi_s - \gamma \phi_s'||_2}$

In the rest of this section, we will first outline an alternative proof for the convergence guarantees of the regular TD(0) algorithm from (Tsitsiklis & Van Roy, 1997), and use the methods developed there to show some convergence guarantees for the Normalized TD(0)

J.2. Convergence guarantees for regular TD(0)

Proposition J.2.1. $P$ is non-expansive, or in other words $||PV||_D \leq ||V||_D \quad \forall V \in \mathbb{R}^n$. Further $P$ has only positive terms

Proof Sketch. First part shown in (Tsitsiklis & Van Roy, 1997) and both are general properties of transition matrices for Markov chains
Proposition J.2.2. $||V - V^L||_D \geq ||V^T - \gamma PV^T - R||$

Proof. Expanding on $V = R + \gamma PR + \gamma^2 P^2 R + \ldots$

$$||V^L - V||_D = ||V^L - (R + \gamma PR + \gamma^2 P^2 R + \ldots)||_D$$

Expanding $V^L = V^L + (\gamma PV^L - \gamma PV^L) + (\gamma^2 P^2 V^L - \gamma^2 P^2 V^L) + (\gamma^3 P^3 V^L - \gamma^3 P^3 V^L) + \ldots$, we get

$$||V^L - V||_D = ||(V^L + (\gamma PV^L - \gamma PV^L) + (\gamma^2 P^2 V^L - \gamma^2 P^2 V^L) + \ldots) - (R + \gamma PR + \gamma^2 P^2 R + \ldots)||_D$$

We rearrange the terms now. Note that we are not in a conditionally convergent series, thus the Riemann rearrangement theorem doesn't apply for $\gamma < 1$

$$= ||(V^L - \gamma PV^L - R) + \gamma P(V^L - \gamma PV^L - R) + \gamma^2 P^2(V^L - \gamma PV^L - R) + \ldots||_D$$

From Cauchy Schwarz and using proposition J.2.1

$$\geq ||V^L - \gamma PV^L - R||_D$$

But since $V^T$ is the minimizer of the above equation:

$$\geq ||V^T - \gamma PV^T - R||_D \quad (46)$$

$\square$

Proposition J.2.3. $||V^T - V||_D \leq \frac{1}{1 - \gamma} ||V^L - V||_D$

Proof. We start with the LHS. Recall that $V = R + \gamma PR + \gamma^2 P^2 R + \ldots$

$$||V^T - V||_D = ||V^T - (R + \gamma PR + \gamma^2 P^2 R + \ldots)||_D$$

We now add and subtract terms of the form $\gamma^i P^i V^T \quad \forall i \in \{1 \ldots\}$ to get:

$$= ||(V^T + (\gamma^1 P^1 V^T - \gamma^1 P^1 V^T) + (\gamma^2 P^2 V^T - \gamma^2 P^2 V^T) + (\gamma^3 P^3 V^T - \gamma^3 P^3 V^T) + \ldots) - (R + \gamma PR + \gamma^2 P^2 R + \ldots)||_D$$

Note that we can rearrange the terms as the expressions are not conditionally convergent, and the Riemann rearrangement theorem doesn’t apply for $\gamma < 1$

$$= ||(V^T - \gamma PV^T) + (\gamma PV^T - \gamma^2 P^2 V^T) + (\gamma^2 P^2 V^T - \gamma^3 P^3 V^T) + \ldots - (R + \gamma PR + \ldots)||_D$$

which we can rewrite as:

$$= ||(V^T - \gamma PV^T) + \gamma P(V^T - \gamma PV^T) + \gamma^2 P^2(V^T - \gamma PV^T) + \ldots - (R + \gamma PR + \ldots)||_D$$
Taking R terms inside the brackets

\[ = \|(V^T - \gamma PV^T - R) + \gamma P(V^T - \gamma PV^T - R) + \gamma^2 P^2(V^T - \gamma PV^T - R) + \ldots \|_D \]

Then from Cauchy Schwarz:

\[ \leq \|(V^T - \gamma PV^T - R)\|_D + \|(\gamma P(V^T - \gamma PV^T - R))\|_D + \|\gamma^2 P^2(V^T - \gamma PV^T - R)\|_D + \ldots \]

Which we can rewrite based on proposition J.2.1 that \(P\) is non-expansive

\[ \leq \|(V^T - \gamma PV^T - R)\|_D + \|\gamma (V^T - \gamma PV^T - R)\|_D + \|\gamma^2 (V^T - \gamma PV^T - R)\|_D + \ldots \]

Taking \(\gamma\) terms outside:

\[ = \|(V^T - \gamma PV^T - R)\|_D + \gamma \|(V^T - \gamma PV^T - R)\|_D + \gamma^2 \|(V^T - \gamma PV^T - R)\|_D + \ldots \]

We now use proposition J.2.2 to bound the above:

\[ \leq \|V^L - V\|_D + \gamma \|V^L - V\|_D + \gamma^2 \|V^L - V\|_D + \ldots \]

Taking terms common:

\[ = \|V^L - V\|_D (1 + \gamma + \gamma^2 + \ldots) \]

To finally get:

\[ = \frac{1}{1 - \gamma} \|V^L - V\|_D \]  \hspace{1cm} (47)

Now we follow similar methods for our Normalized TD(0).

### J.3. Convergence guarantees for Normalized TD(0)

**Proposition J.3.1** (Helper proposition). Let \(\overline{R}\) be the one step expected reward vector. Then \(\overline{R} = [R \circ \mathcal{P}] \cdot 1_{m \times 1}\)

**Proof.** For the LHS, we have for a given state, the expected reward \(\overline{R}_s = R \cdot \mathcal{P}_s = \sum_{s' \in S} \mathcal{P}_{ss'} R_{ss'}\)

For the RHS, \([P \circ R]_{ss'} = \mathcal{P}_{ss'} R_{ss'}\) and therefore \(([P \circ R] \cdot 1_{m \times 1})_s = \sum_{s' \in S} \mathcal{P}_{ss'} R_{ss'}\)

**Definition 3.** Let us construct a Value matrix \(V\) from the value vector \(V\) as follows. \(V = V \cdot 1_{m \times 1}^T\). In other words the value matrix \(V \in \mathbb{R}^{|S| \times |S|}\) has each element in row \(V_s\) filled with the same value \(V_s\).

Then we define \(V\) corresponding to \(V\), \(V^L\) corresponding to \(V^L\), \(V^N\) corresponding to \(V^N\)

**Proposition J.3.2** (Helper proposition). We have the following facts with respect to the value matrix \(V\) that will help us in the main proposition:

1. \([P \circ V] \cdot 1_{m \times 1} = V\)
2. 

\[ [\mathcal{P} \circ \mathcal{V}^T] \cdot 1_{m \times 1} = \mathcal{P} \mathcal{V} \]

**Proof.** For the first part notice that the rows of the probability matrix sum to 1.

For the second part, consider the LHS. Then \((\mathcal{P} \circ \mathcal{V}^T)_{s s'} = \mathcal{P}_{s s'} \mathcal{V}_{s'}\). Further, the entry for the \(s'\)th row of the LHS is given by \(\sum_{s' \in \mathcal{S}} \mathcal{P}_{s s'} \mathcal{V}_{s'}\).

Now consider the RHS. The entry for the \(s'\)th row of the RHS is given by \([\mathcal{P} \mathcal{V}]_{s} = \sum_{s' \in \mathcal{S}} \mathcal{P}_{s s'} \mathcal{V}_{s'}\).

\[ \square \]

**Proposition J.3.3.** We have the following:

1. \(V = [R \circ \mathcal{P}(1 + \gamma \mathcal{P} + \gamma^2 \mathcal{P}^2 + \ldots)] \cdot 1_{m \times 1}\)

2. The above can be written as \(V = \mathcal{R} + \gamma \mathcal{P} \mathcal{R} + \gamma^2 \mathcal{P}^2 \mathcal{R} + \ldots\)

**Proof.** The value is the sum of the expected rewards at step 1, 2, 3 and so on. At the first step, we earn a reward equaling \([R \circ \mathcal{P}] \cdot 1_{m \times 1}\). At the next step, the expected reward can be given as transitions to state \(s\) with probability \(\mathcal{P}_s\), and the one step reward from there equaling \(\mathcal{P} \cdot [R \circ \mathcal{P}] \cdot 1_{m \times 1}\). Since we discount future rewards by \(\gamma\), the expected discounted reward for the second step is \(\gamma \mathcal{P} \cdot [R \circ \mathcal{P}] \cdot 1_{m \times 1}\). In this manner we build up the value vector as a discounted sum of rewards. Then \(V = (1 + \gamma \mathcal{P} + \gamma^2 \mathcal{P}^2 + \ldots) [R \circ \mathcal{P}] \cdot 1_{m \times 1}\), which can be rewritten as \(V = [R \circ \mathcal{P}(1 + \gamma \mathcal{P} + \gamma^2 \mathcal{P}^2 + \ldots)] \cdot 1_{m \times 1}\) since \((1 + \gamma \mathcal{P} + \gamma^2 \mathcal{P}^2 + \ldots)\) is a scalar.

The second part follows from proposition J.3.1.

\[ \square \]

**Proposition J.3.4.** \(w^N\) is the minimizer for \(\left\| \mathcal{N} (\mathcal{V} - \gamma \mathcal{V}^T - R) \circ \mathcal{P} \right\|_D\).\]

**Proof.** The matrix corresponding to the individual error terms \(V_s - \gamma V_{s'} = R_{s s'}\) is \(\mathcal{V} - \gamma \mathcal{V}^T = R\). Then the matrix corresponding to \(V_s - \gamma V_{s'}\) is \(\mathcal{N} [\mathcal{V} - \gamma \mathcal{V}^T]\) which is \(\mathcal{N} R\). Or in other words, we are attempting to set \(E_{s' \in \mathcal{S}} \mathcal{N} [\mathcal{V} - \gamma \mathcal{V}^T] = 0\).

Since we are sampling \(s' \in \mathcal{S}\) as per \(\mathcal{P}_s\), we get the expectation above in the matrix form as \(\mathcal{N} [\mathcal{V} - \gamma \mathcal{V}^T - R] \circ \mathcal{P} \cdot 1_{m \times 1}\). Finally, since \(w^N\) is the minimizer for the above equation, where states \(s \in \mathcal{S}\) are sampled as per \(\pi\) which has diagonal matrix \(D\), we get:

\[ w^N = \arg \min_{w \in \mathbb{R}^n} \left\| \mathcal{N} (\mathcal{V} - \gamma \mathcal{V}^T - R) \circ \mathcal{P} \right\|_D \]

where the D-norm is due to sampling from \(\pi\) which corresponds to diagonal matrix \(D\).

\[ \square \]

Based on proposition J.3.4, we have the following corollary:

**Corollary J.3.1.** Converting the matrices in proposition J.3.4 to a vector form, we have:

\(w^N\) is the minimizer for \(\|\mathcal{N}(\mathcal{V} - \gamma \mathcal{P} \mathcal{V} - \mathcal{R})\|_D\).
We rearrange the terms now. Note that this is an absolutely convergent series. i.e. we are not in a conditionally convergent series. Then if $w^N$ is the minimizer for $\|N(V - \gamma V^T - R) \circ P \cdot 1_{m \times 1}\|_D = \|N[V \circ P \cdot 1_{m \times 1} - \gamma V^T \circ P \cdot 1_{m \times 1} - R \circ P \cdot 1_{m \times 1}]\|_D$ Then this is equivalent to saying, $w^N$ is the minimizer for $\|N[V - \gamma PV - \overline{R}]\|_D$

Now we can get into our main proposition that bound $V^N$. First we see

**Proposition J.3.5.** $\left\| N[V^L - V] \right\|_D \geq \left\| N[V^N - \gamma PV^N - \overline{R}] \right\|_D$

**Proof.** Expanding on $V = \overline{R} + \gamma P\overline{R} + \gamma^2 P^2 \overline{R} + \ldots$

$$\left\| N[V^L - V] \right\|_D = \left\| N[V^L - (\overline{R} + \gamma P\overline{R} + \gamma^2 P^2 \overline{R} + \ldots)] \right\|_D$$

Expanding $V^L = V^L + (\gamma PV^L - \gamma PV^L) + (\gamma^2 P^2 V^L - \gamma^2 P^2 V^L) + (\gamma^3 P^3 V^L - \gamma^3 P^3 V^L) + \ldots$, we get

$$= \left\| N[(V^L + (\gamma PV^L - \gamma PV^L) + (\gamma^2 P^2 V^L - \gamma^2 P^2 V^L) + \ldots) - (\overline{R} + \gamma P\overline{R} + \gamma^2 P^2 \overline{R} + \ldots)] \right\|_D$$

We rearrange the terms now. Note that this is an absolutely convergent series. i.e. we are not in a conditionally convergent series, thus the Riemann rearrangement theorem does not apply, thus, we are free to rearrange terms as long as $\gamma < 1$

$$= \left\| N[((V^L - \gamma PV^L) + (\gamma PV^L - \gamma^2 P^2 V^L) + \ldots) - (\overline{R} + \gamma P\overline{R} + \gamma^2 P^2 \overline{R} + \ldots)] \right\|_D$$

Note that we can bring the $\overline{R}$ terms inside the brackets, and also take $(\gamma P)^i \quad \forall i \in \{1, \ldots \}$ common

$$= \left\| N[(V^L - \gamma PV^L - \overline{R}) + \gamma P(V^L - \gamma PV^L - \overline{R}) + (\gamma P)^2(V^L - \gamma PV^L - \overline{R}) + \ldots] \right\|_D$$

$$= \left\| N(V^L - \gamma PV^L - \overline{R}) + \gamma P N(V^L - \gamma PV^L - \overline{R}) + (\gamma P)^2 N(V^L - \gamma PV^L - \overline{R}) + \ldots \right\|_D$$

From Cauchy Schwarz and using proposition J.2.1

$$\geq \left\| N(V^L - \gamma PV^L - \overline{R}) \right\|_D$$

But since $V^N$ is the minimizer of the above equation from corollary J.3.1:

$$\geq \left\| N(V^N - \gamma PV^N - \overline{R}) \right\|_D \quad \text{(49)}$$

By proposition J.3.1, we have $[\mathbf{P} \circ V] \cdot 1_{m \times 1} = V$ and $[\mathbf{P} \circ V^T] \cdot 1_{m \times 1} = \mathbf{P}V$. By proposition J.3.1, we have

$$\overline{R} = [\mathbf{R} \circ \mathbf{P}] \cdot 1_{m \times 1}.$$

**Proposition J.3.6.** $\left\| N[V^N - V] \right\|_D \leq \frac{1}{1 - \gamma} \left\| N[V^L - V] \right\|_D$

**Proof.** We start with the LHS. Recall that $V = \overline{R} + \gamma P\overline{R} + \gamma^2 P^2 \overline{R} + \ldots$

$$\left\| N[V^N - V] \right\|_D = \left\| N[V^N - (\overline{R} + \gamma P\overline{R} + \gamma^2 P^2 \overline{R} + \ldots)] \right\|_D$$
We now add and subtract terms of the form \( \gamma^i \mathcal{P}^i V^N \) \( \forall i \in \{1 \ldots \} \) to get:

\[
= \left\| \mathcal{N} \left[ (V^N + (\gamma \mathcal{P} V^N - \gamma \mathcal{P} V^N) + (\gamma^2 \mathcal{P}^2 V^N - \gamma^2 \mathcal{P}^2 V^N) + \ldots ) - (\mathcal{R} + \gamma \mathcal{P} \mathcal{R} + \ldots) \right] \right\|_D
\]

The above expression is absolutely convergent. Note that we can rearrange the terms as the expressions are not conditionally convergent, and the Riemann rearrangement theorem doesn’t apply for \( \gamma < 1 \)

\[
= \left\| \mathcal{N} \left[ (V^N - \gamma \mathcal{P} V^N + \mathcal{R}) + \gamma \mathcal{P} (V^N - \gamma \mathcal{P} V^N + \mathcal{R}) + \gamma^2 \mathcal{P}^2 (V^N - \gamma \mathcal{P} V^N + \mathcal{R}) + \ldots \right] \right\|_D
\]

Which we rewrite as:

\[
= \left\| \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) + \gamma \mathcal{P} \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) + \gamma^2 \mathcal{P}^2 \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) + \ldots \right\|_D
\]

Then from Cauchy Schwarz:

\[
\leq \left\| \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) \right\|_D + \left\| \gamma \mathcal{P} \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) \right\|_D + \ldots
\]

Since \( \mathcal{P} \) is non expansive based on proposition J.2.1, we get:

\[
\leq \left\| \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) \right\|_D + \left\| \gamma \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) \right\|_D + \left\| \gamma^2 \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) \right\|_D + \ldots
\]

Since \( \gamma \) is scalar:

\[
= \left\| \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) \right\|_D + \gamma \left\| \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) \right\|_D + \gamma^2 \left\| \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) \right\|_D + \ldots
\]

\[
= \left\| \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) \right\|_D (1 + \gamma + \gamma^2 + \ldots)
\]

\[
= \frac{1}{1 - \gamma} \left\| \mathcal{N}(V^N - \gamma \mathcal{P} V^N + \mathcal{R}) \right\|_D
\]

We now use proposition J.3.5 to bound the above:

\[
\leq \frac{1}{1 - \gamma} \left\| \mathcal{N} [V^L - V] \right\|_D
\] (50)