A note on the function approximation error bound for risk-sensitive reinforcement learning

Prasenjit Karmakar, Shalabh Bhatnagar

July 12, 2017

Abstract

In this paper we obtain several error bounds on function approximation for the policy evaluation algorithm proposed by Basu et al. when the aim is to find the risk-sensitive cost represented using exponential utility. We also give examples where all our bounds achieve the “actual error” whereas the earlier bound given by Basu et al. is much weaker in comparison. We show that this happens due to the absence of difference term in the earlier bound which is always present in all our bounds when the state space is large. Additionally, we discuss how all our bounds compare with each other.

1 Introduction

The most familiar metrics in infinite horizon sequential decision problems are additive costs such as discounted cost and long-run average cost respectively. However, there is another cost criterion namely multiplicative cost (or risk-sensitive cost as it is better known) which has important connections with dynamic games and robust control and is popular in certain applications, particularly related to finance where it offers the advantage of ‘penalizing all moments’, so to say, thus capturing the ‘risk’ in addition to mean return (hence the name). For details see [5].

Like other cost criterion, one can propose and justify iterative algorithms for solving the dynamic programming equation for risk-sensitive setting [10]. The issue we are interested in here is how to do so, even approximately, when the exact model is either unavailable or too unwieldy to afford analysis, but on the other hand simulated or real data is available easily, based on which one may hope to ‘learn’ the solution in an incremental fashion.

One important point to note here is that the usual simulation based technique of calculating average cost does not work when the objective is a risk-sensitive cost. The reason is that average cost is defined as

\[ \lim_{n \to \infty} \frac{1}{n} E \left[ \sum_{i=0}^{n-1} c(X_i) \right], \]

where \( c(i) \) is the cost of state \( i \) and \( X_n, n \geq 0 \) is an irreducible finite state Markov chain. Therefore the following iterative algorithm will almost surely converge to the average cost:

\[ \theta_{n+1} = \theta_n + a(n) [c(X_n) - \theta_n], \]

where the step sizes satisfy the Robbins-Monro conditions. This follows from the ergodic theorem for irreducible Markov chains as well the convergence analysis of stochastic approximation with Markov noise [11]. On the contrary one needs to apply multiplicative ergodic theorem [II] when the cost is risk-sensitive. However, this does not have any closed-form limit. Moreover, one cannot even write iterative algorithms like (1) in this setting because of the non-linear nature of the cost. Due to the same reason, methods of [7] also don’t work in this setting when one is solving the full control problem.

This takes us into the domain of reinforcement learning. In [9] and [8], Q-learning and actor-critic methods have been proposed respectively for such a cost-criterion. These are ‘raw’ schemes in the sense that there is no further approximation involved. Since complex control problems lead to dynamic programming equations in very large dimensions (‘curse of dimensionality’), one often looks for an approximation based scheme. One such learning algorithm with function approximation is proposed in [3].

In such approximation architectures an important problem is to obtain a good error bound for the approximation. This has been pointed out by Borkar in the future work sections of [12, 3, 13]. While [3] provides such a bound when the problem is policy evaluation, it is clearly mentioned there that the bound obtained is not good when the state space is large. In this technical note we investigate the problems with the existing bound and then improve upon the same. We show that good approximations are captured in our bounds whereas the earlier bound will infer them as bad approximation.
The paper is organized as follows: Section 2 describes the preliminaries and background of the problem considered. Section 3 discusses the shortcomings of the bound proposed by Basu et al. Section 4 shows the theoretical conditions under which there is no error. This section also describes verifiable conditions when the transition kernel is doubly stochastic. Section 5 describes the new error bounds as well as how they compare with each other and with the state of the art bound. Section 6 presents conclusions and some future research directions.

2 Preliminaries and Background

We begin by recalling the risk-sensitive framework. Consider an irreducible aperiodic Markov chain \( \{X_n\} \) on a finite state space \( S = \{1, 2, \ldots, s\} \), with transition matrix \( P = [[p(j|i)]] \)\( i, j \in S \). While our real concern is a controlled Markov chain, we aim at a policy evaluation algorithm for a fixed stationary policy. Thus we have suppressed the explicit control dependence. Let \( c : S \times S \to \mathbb{R} \) denote a prescribed ‘running cost’ function and \( C \) be the \( s \times s \) matrix whose \((i, j)\)-th entry is \( c(i, j) \). The aim is to evaluate

\[
\limsup_{n \to \infty} \frac{1}{n} \ln \left( E[e^{\sum_{m=0}^{n-1} c(X_m, X_{m+1})}] \right).
\]

That this limit exists follows from the multiplicative ergodic theorem for Markov chains (see Theorem 1.2 of Balaji and Meyn (2000) [1], the sufficient condition (4) therein is trivially verified for the finite state case here). Associated with this is the multiplicative Poisson equation (see, e.g., Balaji and Meyn (2000) [1, Theorem 1.2 (ii)]): We know from [10, 9, 8] that in the case of value iteration (with both dynamic programming and reinforcement learning) that the approximation version in [3] provides the following parameter update for \( \lambda \)

\[
V(i) = \frac{\sum_j p(j|i) e^{c(i,j)} V(j)}{\lambda}.
\]

For an explicit expression for \( V(.) \) see (5) in [1]. Let \( \tau_n = \min \{m > 0 : X_m = i_0\} \) for a prescribed \( i_0 \in S \).

Thus \( \lambda \) and \( V \) are respectively the Perron-Frobenius eigenvalue and eigenvector of the non-negative matrix \([e^{c(i,j)} p(j|i)]]_{i,j \in S} \), whose existence is guaranteed by the Perron-Frobenius theorem. Furthermore, under our irreducibility assumption, \( V \) is specified uniquely up to a positive multiplicative scalar and \( \lambda \) is uniquely specified. Also, the risk-sensitive cost defined as above is \( \ln \lambda \).

We know from [10, 9, 8] that in the case of value iteration (with both dynamic programming and reinforcement learning) that the \( i_0 \)-th component of the sequence of the iterates will converge to \( \lambda \). The linear function approximation version in [3] provides the following parameter update for \( n \geq 0 \):

\[
r_{n+1} = r_n + a(n) \left( \frac{B_n^{-1} A_n}{\max(\phi^T (i_0) r_n, \epsilon)} - I \right) r_n,
\]

where

\[
\epsilon > 0, \quad \text{is fixed}
\]

\[
V(i) \approx \sum_{k=1}^{M} r^k \phi_k(i) = \phi^T(i) r,
\]

\[
r = (r^1, \ldots, r^M)^T \quad \text{is a vector of coefficients},
\]

\[
\phi^k(\cdot), 1 \leq k \leq M, \quad \text{are the basis functions or features chosen a priori},
\]

\[
\phi(i) = (\phi^1(i), \ldots, \phi^M(i))^T,
\]

\( \Phi = a s \times M \) matrix whose \((i, k)\)-th entry is \( \phi^k(i) \) for \( 1 \leq i \leq s \) and \( 1 \leq k \leq M \), \( \phi(i) = \text{feature of state } i \),

\[
A_n = \sum_{m=0}^{n} e^{c(X_m, X_{m+1})} \phi(X_m) \phi^T(X_{m+1}),
\]

\[
B_n = \sum_{m=0}^{n} \phi(X_m) \phi^T(X_m),
\]

\( I = M \times M \) identity matrix.

We also know from [3, Theorem 5.3] that under a crucial assumption (see (†) in p 883 there) on the feature matrix, the iterates \( r_n \) satisfy the following:

\[
\phi^T (i_0) r_n \to \mu,
\]
where \( \mu > 0 \) is a Perron-Frobenius eigenvalue of the non-negative matrix \( Q = \Pi M \) with \( \Pi = \Phi (\Phi^T D \Phi)^{-1} \Phi^T D \) and \( M = C \circ P \) (unlike [3] we consider only a synchronous implementation for ease of understanding). Here \( D \) is a diagonal matrix with the \( i \)-th diagonal entry being \( \pi_i \), where \( \pi = (\pi_1, \pi_2, \ldots, \pi_n)^T \) is the stationary distribution of \( \{X_n\} \). Also, \( e^{i(j)} p(j|i) \) is the \( (i,j) \)-th entry of \( C \circ P \) where ‘\( \circ \)’ denotes the component-wise product of two matrices with identical row and column dimensions. Assume that \( \gamma_{ij} \) and \( \delta_{ij} \) are the \( (i,j) \)-th entry of the matrix \( C \circ P \) and \( \Pi M \) respectively.

Therefore \( \ln \mu \) serves as an approximation of the original risk-sensitive cost \( \ln \lambda \). Our aim is to investigate the difference between these two i.e. \( \ln \left( \frac{1}{\mu} \right) \).

**Remark 1.** Throughout the paper the results are stated in general for matrices \( A \) and \( B \) with largest eigenvalue of \( A \) and \( B \) as \( \lambda > 0 \) and \( \mu > 0 \) respectively. It should be clear from the context what are the entries of \( A \) and \( B \).

### 3 Related work and shortcomings

Let \( \|A\| \) be the operator norm of a matrix defined by \( \|A\| = \inf \{ c > 0 : \|Av\| \leq c\|v\| \ \forall v \} \) where \( \|v\| = \sum_{i=1}^n |v_i| \).

Let \( A = C \circ P \) and \( B = \Pi M \). The following bound was given in [3]:

\[
\ln \left( \frac{\lambda}{\mu} \right) \leq \ln \left( 1 + \frac{\|A\| + \|B\|}{\mu} \right),
\]

using the spectral variation bound from [3, Theorem VIII.1.1], namely that if \( A \) and \( B \) are two \( s \times s \) matrices with eigenvalues \( \alpha_1, \ldots, \alpha_s \) and \( \beta_1, \ldots, \beta_s \) respectively, then

\[
\max_{1 \leq i \leq s} |\alpha_i - \beta_i| \leq (\|A\| + \|B\|)^{-1}(\|A - B\|)^{\frac{1}{2}}.
\]

This follows from the observation that if \( \alpha_1 > 0 \) and \( \beta_1 > 0 \) are the leading eigenvalues of \( A \) and \( B \) respectively and \( \alpha_1 \leq \beta_1 \), then \( |\alpha_1 - \beta_1| < \max \min \{ |\alpha_i - \beta_j| \} \). Similar thing happens for the case \( \alpha_1 > \beta_1 \) except that the roles of \( \alpha_1 \) and \( \beta_1 \) and hence the roles of \( A \) and \( B \) get reversed thus keeping the right hand side (R.H.S) of \( \lambda \) the same.

An important point to note is that when \( \alpha_1 \leq \beta_1 \), the fact that \( \beta_1 \) is the leading eigenvalue of \( B \) is not used. Same thing happens for the other case where \( \alpha_1 \) replaces \( \beta_1 \).

Another important point above is that for large \( s \) the bound given above cannot differentiate between the cases with two pairs of matrices \( (A_1, B_1) \) and \( (A_2, B_2) \) such that \( \|A_1\| + \|B_1\| = \|A_2\| + \|B_2\| \) but \( \|A_1 - B_1\| \) and \( \|A_2 - B_2\| \) vary dramatically. This will be clear from the next toy example: Consider \( A_1 = (x_{ij})_{s \times s}, B_1 = (y_{ij})_{s \times s}, A_2 = (z_{ij})_{s \times s}, B_2 = (w_{ij})_{s \times s} \). Suppose \( x_{ij} = p, y_{ij} = q, z_{ij} = p', w_{ij} = q' \) for \( i, j \in 1, 2, \ldots, s \) with \( p + q = p' + q' = 1 \) and \( p - q > 0 \). It is easy to see that \( \|A_1\| = \|B_1\| \) but \( r(A_1) = ps, r(B_1) = qs, r(A_2) = p's, r(B_2) = q's \). Clearly, \( p - q \neq p' - q' \) unless \( pq = p'q' \). Here \( r(A) \) denotes the Perron-Frobenius eigenvalue of matrix \( A \).

In summary, when one is giving a bound between two quantities, the R.H.S should have terms involving the difference. However this does not occur while using spectral variation bound in the above example as \( p - q \) will converge to 1 as \( s \to \infty \). In Sections 5 and 6, using the above example we show that the new error bounds that we obtain contain always the difference terms irrespective of the state space size \( s \).

### 4 Conditions under which error is zero

#### 4.1 Theoretical Conditions

**4.1.1 Condition 1**

**Theorem 4.1.** Let \( \mathbf{x} \) be the left Perron eigenvector of the non-negative matrix \( C \circ P \) i.e. \( \mathbf{x}^T C \circ P = \lambda \mathbf{x}^T \). If \( \Phi \) is an \( s \times 1 \) matrix and \( \phi_i = y_i \) where \( y_i = \frac{x_i}{\pi_i} \), then \( \mu = \lambda \), i.e., there will be no error when function approximation is deployed.

**Proof.** It is easy to check that \( \delta_{ij} = \sum_{k=1}^s \phi_k^T x_{ij} \gamma_{ik} \pi_k \), where \( \gamma_{ij} = e^{i(j)} p(j|i) \).

We claim that with the choice of feature matrix as stated in the theorem, \( \lambda \) is the eigenvalue of \( B \) with eigenvector being \( y = (y_i)_{i \in \{1, 2, \ldots, s\}} \).

\[
(P_\Pi M y)_i = \sum_{k=1}^s \delta_{ik} y_k = \sum_{k=1}^s y_i \frac{x_{ik} \gamma_{ik}}{\pi_k} y_k = \sum_{k=1}^s y_i \frac{\sum_{k=1}^s x_{ik} \gamma_{ik}}{\sum_{m=1}^s \pi_m} y_k = \lambda y_i \frac{\sum_{k=1}^s y_k^2}{\sum_{m=1}^s \pi_m} \]

\( \square \)
4.1.2 Condition 2

Recall the assumption (i) on the feature matrix \( \Phi \) from [3] which says that the feature matrix \( \Phi \) has all non-negative entries and any two columns are orthogonal to each other. In this paper we strengthen the later part as follows:

(*) Every row of the feature matrix \( \Phi \) has exactly one positive entry i.e. for all \( i \) there exist \( 1 \leq k(i) \leq M \) such that \( \phi^*(i) > 0 \) if \( j = k(i) \), otherwise \( \phi^*(i) = 0 \).

From [2, Theorem 1] it is easy to see that (this theorem is applicable due to Lemma 5.1 (ii) of [3] and (*)) the error can be zero even if \( C \circ P \neq \Pi M \), namely under the following conditions:

1. there exists positive \( \lambda_0, \beta_i, i = 1, 2, \ldots, s \) such that
\[ \delta_{ij} = \frac{\lambda_0 \gamma_{ij} \beta_i}{\beta_j}, i, j = 1, 2, \ldots, s. \]
2. \( \Pi_{i,j=1}^s \delta_{ij} \gamma_{ij} x_i y_j = \Pi_{i,j=1}^s \gamma_{ij} x_i y_j. \)

**Remark 2.** Note that if the matrix \( \Phi \) has a row \( i \) with all 0s, then \( \delta_{ij} = 0 \) for all \( j = 1, 2, \ldots, s \) whereas \( \gamma_{ij} > 0 \) for at least one \( j \in \{1, 2, \ldots, s\} \) which violates the conditions for zero error stated above.

4.2 Verifiable Condition with doubly stochastic transition kernel

Note that if the transition kernel is a doubly stochastic matrix then it is very hard to find easily verifiable condition on the feature matrix such that \( C \circ P = \Pi M \). The reason is that this requires to find a feature matrix \( \Phi \) which under (*) satisfies \( \Phi (\Phi^T \Phi)^{-1} \Phi^T = I \). This will not be true under (*) as this requires \( k(i) \neq k(j) \) to hold if \( i \neq j \). This problem can be alleviated by the temporal difference learning algorithm for this setting as under:

\[ \theta_{t+1} = \theta_t + a(n) \left[ \frac{e_c(X_t, X_{t+1}) \phi^T(X_{t+1}) \theta_n}{\phi^T(x_t) \theta_n} - \phi^T(X_t) \theta_n \right] \phi(X_t). \]

The following theorem shows its convergence.

**Theorem 4.2.** If \( \Phi \Phi^T = D^{-1} \) then \( \phi^T(x_t) \theta_n \rightarrow \lambda \).

**Proof.** Note that the algorithm tracks the o.d.e
\[ \dot{\theta}(t) = \left( \frac{A'}{\phi^T(x_t) \theta(t)} - B' \right) \theta(t), \]
where \( A' = \Phi^T D \circ P \Phi \) and \( B' = \Phi^T D \Phi \).

This follows because it is easy to see that the algorithm tracks by the o.d.e
\[ \dot{\theta}(t) = h(\theta(t)), \]
where \( h(\theta) = \sum_{j} \pi(j)p(j|i) \left[ \frac{\phi^T(j) \theta}{\phi^T(x_t) \theta(t)} - \phi^T(i) \theta \right] \phi(i) \).

Now, the \( k \)-th entry of \( A' \theta \) is
\[ \left\langle \sum_{i=1}^s \phi^k(i) \sum_{j=1}^s e^{c(i,j)} p(j|i) \phi(j) \right\rangle, \theta. \]

Similarly, the \( k \)-th entry of \( B' \theta \) can be shown to be the \( k \)-th entry of \( \sum_{n} \sum_{j} \pi(n)p(j|i) \left[ \phi^T(i) \theta \right] \phi(i) \).

Now, the claim follows directly from [3, Theorem 5.3] (the synchronous implementation).

5 New error bounds

5.1 Bound based on Bapat and Lindqvist’s inequality

Motivated by the discussion in Section 3 and the fact that risk-sensitive cost is \( \ln \lambda \) rather than \( \lambda \) we need to find an upper bound for \( \ln \frac{\lambda}{\mu} \). Let \( r(A) \) denote the Perron-Frobenius eigenvalue of matrix \( A = (a_{ij})_{s \times s} \). In the following we obtain three different bounds for the same quantity under the assumptions that a) \( \lambda > \mu \) b) the matrix \( P = p(j|i) \) has positive entries and impose conditions under which one is better than the other. Suppose \( A \) admits left and right Perron eigenvectors \( x, y \) respectively, with \( \sum_i x_i y_i = 1 \) (this is satisfied, for example, if \( A \) is irreducible). The three upper bounds of \( \ln \lambda - \ln \mu \) are [7] - [9].
Note that the actual error is \(\ln(1 + \frac{\epsilon}{q})\) where \(p = q + \epsilon\) where \(\epsilon < q\). If we use (3) then the error is bounded by \(\ln(3 + \frac{\epsilon}{q})\). However, if we use (10) the error is bounded by \(\ln(1 + \frac{\epsilon}{q})\) i.e. the actual error. If we use (13) the error is bounded by \(\ln \left(1 + \left(1 + \frac{\epsilon}{q}\right) \ln \left(1 + \frac{\epsilon}{q}\right)\right)\) which reduces to \(\ln \left(1 + \frac{\epsilon}{q}\right)\) (using Binomial approximation theorem).

If \(A\) is such that all its diagonal elements are \(p\) and the off-diagonal elements are \(q\) then for large state space the actual error is zero. If we use (10) then the bound is also zero whereas the right hand side of (3) is \(\ln 3\).

If \(A\) is such that the entry in the first row and first column is \(p\) and the rest are all \(q\), then also similar thing happens except the fact that now the right hand side of (3) is \(\ln \left(1 + 2e^{-\frac{\epsilon}{q-1}}\right)\).

Note that here \(a_{ij} > b_{ij}\) for \(i, j \in \{1, 2, \ldots, s\}\) in the above example. Our bound will be much more useful in cases where there will be \(i, j\) such that \(b_{ij} > a_{ij}\). From the definition of \(\delta_{ij}, \gamma_{ij}\) it is clear that for all \(j\) there exists \(i\) such that \(\delta_{ij} > \gamma_{ij}\). In such a case, for every \(j\) there will be at least one non-positive term inside the summation over \(i\) which will make the bound small. The bound given in (3) does not capture such cases for large \(s\).

Here (8) holds under (14) which follows from the fact that the following condition is necessary and sufficient for (11) to be true:

\[
r(A) > \sum_{i=1}^{s} x_iy_i(a_{ii} - b_{ii}) + \sum_{i \neq j} a_{ij}x_iy_j \ln \left(\frac{a_{ij}}{b_{ij}}\right)
\]

and \(\min, \sum a_{ij} \leq r(A)\). Later in the proof of Lemma 5.3 we will see that, in our setting, under (*), (13) gets satisfied if the assumptions in Lemma 5.3 is true.
Therefore (7) is better than (8).

Under the condition mentioned in [6, Theorem 3 (i)],

\[ \text{Proof.} \]

Assume that for all \( i \), \( \text{Lemma 5.2.} \)

This means that (9) is better than (8).

\[ \text{Lemma 5.3.} \]

One such example where the condition of Lemma 5.2 gets satisfied is: \( \text{Lemma 5.1.} \)

\[ \text{In this section we describe some conditions. They are sufficient conditions under which (7)-(8) compare with each other. They will be referred in the next two lemmas.} \]

5.1.1 Some conditions

In this section we describe some conditions. They are sufficient conditions under which (7)-(8) compare with each other. They will be referred in the next two lemmas.

\[ \forall i, e^{c(i,j)} p(j|i) \left( \sum_{m=1}^{s} \phi_k(i)(m)^2 \pi_m - \phi_k(i)(i)^2 \pi_i \right) = \phi_k(i)(i) \sum_{l=1, l \neq i}^{s} \phi_k(i)(l) \pi_l a_{li} \] (15)

\[ \forall i \neq j, e^{c(i,j)} p(j|i) \left( \sum_{m=1}^{s} \phi_k(i)(m)^2 \pi_m - \phi_k(i)(i)^2 \pi_i \right) = \phi_k(i)(i) \sum_{l=1, l \neq i}^{s} \phi_k(i)(l) \pi_l a_{lj} \] (16)

\[ \exists i \text{ s.t } e^{c(i,i)} p(i|i) > \max_{1 \leq i \leq s, i \neq j} e^{c(i,i)} p(i|i) \text{ or,} \] (17)

\[ \exists i \text{ s.t } e^{c(i,i)} p(i|i) < \min_{1 \leq i \leq s, i \neq j} e^{c(i,i)} p(i|i) \] (18)

\[ \text{Lemma 5.2. Assume that for all } i, b_{ii} = a_{ii} \text{ [6, Theorem 3 (i)]}. \text{ Then (7) is better than (8)} \]

\[ \text{Proof. Under the condition mentioned in [6, Theorem 3 (i)],} \]

\[ r(A) \prod_{i \neq j} \left( b_{ij} \bigg/ a_{ij} \right) \geq r(A) - L. \]

Therefore (7) is better than (8). \( \Box \)

\[ \text{Remark 4. One such example where the condition of Lemma 5.2 gets satisfied is: } A = (a_{ij})_{s \times s} \text{ with } a_{ij} = q \text{ if } i = j \text{ and } a_{ij} = p \text{ otherwise, and } B = (b_{ij})_{s \times s} \text{ with } b_{ij} = q \text{ for all } 1 \leq i, j \leq s \text{ with } p = q = 0. \text{ It is easy to check that (13) gets satisfied for this example.} \]

\[ \text{Remark 5. In our setting the condition mentioned in Lemma 5.2 gets satisfied if (13) is true. If the feature matrix is a single column matrix with all entries equal then a sufficient condition for (13) is that for every } j, e^{c(i,j)} p(j|i) \text{ is same for all } i \text{ (for example, the transition probability satisfies } p(j|i) = e^{-c(i,j)} \text{ with the cost function } c(,.) \text{ being non-negative).} \]

\[ \text{Lemma 5.3. Assume that for all } i \neq j, b_{ij} = a_{ij} \text{ and there is at least one } i \text{ such that } b_{ii} \neq a_{ii} \text{ [6, Theorem 3 (ii)]. Then (8) is better than first (7).} \]
Proof. Under the condition mentioned in [6, Theorem 3 (ii)],
\[
\begin{align*}
    r(A)\Pi_{i=1}^{s} \left( \frac{b_{ii}}{a_{ii}} \right)^{\frac{a_{ii}}{\Pi_{j=1}^{s} b_{ij}}} & \leq r(A) - L.
\end{align*}
\]
Therefore (13) gets satisfied trivially if for all \( i, b_{ii} \neq 0 \) (which is true in our setting under \((\star)\) and \((b)\)). Therefore (8) is better than (7).

Remark 6. In our setting the condition mentioned in Lemma 5.3 gets satisfied if (16) is true and there exist at least one \( i \) for which either (17) or (18) is true (assuming that feature matrix is a single column matrix with all entries equal). If the feature matrix is a single column matrix with all entries equal then a necessary and sufficient condition for (17) is that for every \( j, e^{(i,j)p(j|i)} \) is same for all \( i \neq j \).

Remark 7. Similar bounds can be derived in the same way if \( \lambda < \mu \).

5.2 Another bound when \( A \) is invertible

Let, \( \alpha(A) = \max_{i}(x_{A})^{-1} \) where \( x_{A} \) is Perron-Frobenius eigenvector of \( A \) which has positive components if \( A \) is irreducible.

Theorem 5.4. Under the assumptions

(A1) \( A \) is invertible

(A2) \( A \) positive semidefinite and

(A3) \( \min_{i} \sum_{j} a_{ij} \min_{i} \sum_{j} b_{ij} \geq \Pi_{i=1}^{s} a_{ii} \),

\[
\ln \left( \frac{\lambda}{\mu} \right) \leq \ln \left( \frac{\det(A)}{\mu (\mu - \alpha(A)\|A-B\|)} \right). \tag{19}
\]

Proof. If \( \lambda \) is an eigenvalue of matrix \( A \) with eigenvector \( x_{A} \), then \( \frac{\det(A)}{\lambda} \) is an eigenvalue of the adjoint \( A^{*} \) with the same eigenvector.

\[
\begin{align*}
    \left( \frac{\det(A)}{\lambda} - \mu \right) \langle x_{A}, x_{B} \rangle &= \langle A^{*}x_{A}, x_{B} \rangle - \langle x_{A}, Bx_{B} \rangle \\
    &= \langle x_{A}, Ax_{B} \rangle - \langle x_{A}, Bx_{B} \rangle \\
    &= \langle x_{A}, (A - B)x_{B} \rangle.
\end{align*}
\]

Moreover,
\[
\langle x_{A}, x_{B} \rangle \geq \alpha(A)^{-1}\|x_{B}\| = \alpha(A)^{-1}.
\]

Then the proof follows from the observation that
\[
|\langle x_{A}, (A - B)x_{B} \rangle| \leq \|A - B\|.
\]

Here all the eigenvectors are normalized so that their norm is 1.

From the above one can easily see that
\[
\left| \frac{\det(A)}{\lambda} - \mu \right| \leq \alpha(A)\|A - B\|.
\]

The result follows from Hadamard inequality [14, p 136].

Remark 8. If \( A = C \circ P \),

1. Using Oppenheim’s Inequality [14, p 144] (A1) is satisfied if \( P \) is positive definite.

2. (A2) is satisfied if \( C \) and \( P \) are both positive semidefinite.

3. Under \((\star)\), (A3) is trivially satisfied if at least one of the diagonal elements of the transition probability matrix is zero.
Remark 9. Let us take \( A = (a_{ij})_{s \times s} \) with \( a_{ij} = p \) if \( i = j \) and \( a_{ij} = q \) otherwise and \( b_{ij} = q \) for all \( i, j \) with \( p > q \). Also, let \( \epsilon := p - q \). Clearly \((A1)-(A2)\) gets satisfied. One can easily put conditions involving \( p, q, s \) so that \((A3)\) also gets satisfied. Then \[ \det(A) = (p + (s - 1)q)(p - q)^{s-1}. \] \( \alpha(A) = s. \) The right hand side of \((17)\) becomes \[ \ln \left( \frac{(q + \epsilon)^{s-1}}{(p - q)} \right). \] If we assume \( \epsilon \ll q \) then it becomes \[ \ln \left( \frac{1}{q} \right). \] The actual error for large \( s \) becomes zero. The right hand side of \((3)\) becomes \[ \ln 3. \] Therefore if \( \frac{1}{3} \leq q < 1 \) our bound is much better than \((3)\).

Remark 10. Note that \( B \) need not be irreducible under the assumption \((\dagger)\) in \((19)\). Therefore, \( x_B \) need not have all the components positive.

6 Conclusion

In this short technical note we gave several new bounds on the function approximation error for policy evaluation algorithm in the context of risk-sensitive reinforcement learning. An important future direction will be to design and analyze suitable learning algorithms to find the optimal policy with the accompanying error bounds. It will be interesting to see whether one can use our bounds for policy evaluation problem to provide error bounds for the full control problem.

References

[1] Balaji, S., & Meyn, S. P. (2000). Multiplicative ergodicity and large deviations for an irreducible Markov chain. Stochastic Processes and Their Appl., 90, 123–144.

[2] Bapat, R. B. (1989). Comparing the Spectral Radii of Two Nonnegative Matrices. The American Mathematical Monthly, 96(2), 137–139.

[3] Basu, A., Bhattacharya, T., & Borkar, V. S. (2008). A Learning Algorithm for Risk-sensitive Cost. Mathematics of operations research, 55(2), 139–145.

[4] Bhatia, R. (1997). Matrix Analysis. Springer.

[5] Kirkwood, C. W. (1991) Notes on Attitude Toward Risk Taking and the Exponential Utility Function. Technical Report AZ 85287-4006.

[6] Lindqvist, B. H. (2002). On Comparison of the Perron-Frobenious eigenvalues of two ML-matrices. Linear Algebra and its applications, 353(2), 257–266.

[7] Marbach, P., & Tsitsiklis, J. N. (2001) Simulation-Based Optimization of Markov Reward Processes. IEEE Transactions on Automatic Control, 46(2), 191–209.

[8] Borkar, V. S. (2001). A sensitivity formula for the risk-sensitive cost and the actor-critic algorithm. Systems and Control Letters, 44, 339–346.

[9] Borkar, V. S. (2002). Q-learning for risk-sensitive control. Mathematics of operations research, 27, 294–311.

[10] Borkar, V. S. & Meyn, S. P. (2002). Risk-sensitive optimal control for Markov decision processes with monotone cost. Mathematics of operations research, 27, 192–209.

[11] Borkar, V. S. (2006). Stochastic approximation with ‘controlled Markov noise’. Systems and Control Letters, 55(2), 139–145.

[12] Borkar, V. S. (2009). Reinforcement learning – a bridge between numerical methods and Markov Chain Monte Carlo. Perspectives in Mathematical Sciences(N. S. N. Sastry, B. Rajeev, Mohan Delampady, T. S. S. R. K. Rao, eds.).

[13] Borkar, V. S. (2010). Learning Algorithms for Risk-Sensitive Control. 19th International Symposium on Mathematical Theory of Networks and Systems.

[14] Bapat, R. B. & Raghavan, T. E. S. (1997). Nonnegative Matrices and Applications. Cambridge University Press, 1997.