I-Learning on Bandits

David Ellis Hershkowitz under the supervision of Michael L. Littman

May 14, 2016

We introduce a bandit algorithm [4] called I-Learning, detail a novel correctness condition for bandits – sort-of-\(\epsilon\)-settle-optimal correctness – and then prove that I-Learning is sort-of-\(\epsilon\)-settle-optimal with high probability.

1 Bandit Setup and Notation

We provide proofs for a 2 armed bandit where each arm pays with a fixed payoff rate according to a two-element payoff vector, \(\mathbf{R}\), where the optimal arm has reward 1 and the sub-optimal arm has reward \(1 - \epsilon\).

We understand a bandit algorithm as returning a series of probability distributions, \(D_i\) where \(D_i\) is the probability distribution used by the algorithm at the \(i\)th pull and \(D_i(a)\) is the probability that arm \(a\) is pulled on the \(i\)th pull.

2 Correctness Conditions

We could like to prove that I-Learning is with high probability nearly optimal; in particular, given a \(\delta\) and an \(\epsilon\), we would like to show that with \(1 - \delta\) probability I-Learning achieves an \(\epsilon\)-optimal solution for some sense of \(\epsilon\)-optimal. We introduce one notion of optimal, settle-optimal, and then provide a series of relaxations that make settle-optimal more amenable to proof.

2.1 \((\epsilon)\)-Settle-Optimal Correctness Condition

The notion of optimal in the bandit setting that we first consider is an algorithm that eventually pulls the optimal arm with probability 1 and at no point prior pulls a sub-optimal arm with probability 1. We term this a settle-optimal solution. More formally,

| Settle-Optimal Solution: | A bandit solution, \(D_i\), is settle-optimal if for the last round \(e = |D|\) there exists an arm \(s\) that satisfies: |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 (1) Settles: \((D_e(s) = 1)\)                                                |
| 2 (2) Optimal-Settling: \((R(s) = 1)\)                                         |
| 3 (3) No Premature Settling: \(\forall j < e \forall k (R(k) < 1) \rightarrow (D_j(k) \neq 1)\) |

We use the following \(\epsilon\)-relaxation of a settle-optimal solution. A solution is \(\epsilon\)-settle-optimal if it eventually pulls an \(\epsilon\)-optimal arm with probability \(1 - \epsilon\) and at no point prior pulls a sub-(1 - \(\epsilon\))-optimal arm. More formally,
\(\epsilon\)-Settle-Optimal Solution: A bandit solution, \(\mathbf{D}\), is \(\epsilon\)-settle-optimal if for the last round \(e = |\mathbf{D}|\) there exists an arm \(s\) that satisfies:

1. \(\epsilon\)-Settles: \((\mathbf{D}_e(s) \geq 1 - \epsilon)\)
2. \(\epsilon\)-Optimal-Settling: \((\mathbf{R}(s) \geq 1 - \epsilon)\)
3. No Premature \(\epsilon\)-Settling: \(\forall j < e |\forall k| (\mathbf{R}(k) < 1 - \epsilon) \rightarrow (\mathbf{D}_j(k) < 1 - \epsilon)\)

\[2.2\] Sort-Of-\(\epsilon\)-Settle-Optimal Correctness Condition

Because proving things about \(\epsilon\)-settle-optimal solutions is hard we instead introduce sort-of-\(\epsilon\)-settle-optimal.

Sort-Of-\(\epsilon\)-Settle-Optimal Solution: A bandit solution, \(\mathbf{D}\), is sort-of-\(\epsilon\)-settle-optimal if for the last round \(e = |\mathbf{D}|\) there exists an arm \(s\) that satisfies:

1. Sort-Of-\(\epsilon\)-Settles: \((\mathbf{D}_e(s) \geq 1 - \min(\epsilon, \frac{1}{3}))\)
2. \(\epsilon\)-Optimal-Settling: \((\mathbf{R}(s) \geq 1 - \epsilon)\)
3. No Premature \(\epsilon\)-Settling: \(\forall j < e |\forall k| (\mathbf{R}(k) < 1 - \epsilon) \rightarrow (\mathbf{D}_j(k) < 1 - \epsilon)\)

Note that a solution that is sort-of-\(\epsilon\)-settle-optimal is \(\epsilon\)-settle-optimal. However, an algorithm that is \(\epsilon\)-settle-optimal is not necessarily sort-of-\(\epsilon\)-settle-optimal. We are now situated to introduce and then prove the sort-of-\(\epsilon\)-settle-optimality of I-Learning.

3 Algorithms

Here we introduce I-Learning as well as variant of it that is more amenable to proving sort-of-\(\epsilon\)-settle-optimality.

3.1 I-Learning

Let \(\mathbf{I}\) be a vector of length \(k\) that stores the accumulated incomes associated with each arm where \(\mathbf{I}(i)\) returns the \(i\)th element of \(\mathbf{I}\). Let \(P(i)\) denote pulling the \(i\)th arm which returns some reward, 1 or \(1 - \epsilon\). We can now formalize I-Learning, which pulls arms according to their income and ends if any arm reaches probability \(1 - \epsilon\):
Algorithm 1 I-Learning

Input: $\delta$, $\epsilon$
Output: A sequence of distributions over arms, $D$

$I = \log_r(\delta(1 - \epsilon)/\epsilon)$ for $r \in (0, 1)$
pull = 1
while True do
    $D_{pull} = M$
    if $\exists j[M(j) > 1 - \epsilon]$ then
        break
    else
        $i \sim \frac{I(j)}{\sum_{j=1}^{M} I(j)}$
        $I(i) += P(i)$
    end if
    pull += 1
end while
return $D$

3.2 Sort-Of-I-Learning

Because proving properties of I-Learning is hard we introduce Sort-Of-I-Learning here. The critical difference is the stopping condition of the algorithm – we now wait until an arm has at least $1 - \min(\epsilon, \frac{1}{3})$ rather than just $1 - \epsilon$ probability to have settled. This parallels the amendment made to (1) of $\epsilon$-settle-optimality to create sort-of-$\epsilon$-settle-optimality.

Algorithm 2 I-Learning

Input: $\delta$, $\epsilon$
Output: A sequence of distributions over arms, $D$

$I = \log_r(\delta(1 - \epsilon)/\epsilon)$ for $r \in (0, 1)$
pull = 1
while True do
    $D_{pull} = M$
    if $\exists j[M(j) > 1 - \min(\epsilon, \frac{1}{3})]$ then
        break
    else
        $i \sim \frac{I(j)}{\sum_{j=1}^{M} I(j)}$
        $I(i) += P(i)$
    end if
    pull += 1
end while
return $D$

4 Analysis

In this section we prove that Sort-Of-I-Learning is sort-of-$\epsilon$-settle-optimal with probability at least $1 - \delta$. In particular, we show that Sort-Of-I-Learning when given $\delta$ and $\epsilon$ has a probability of not being sort-of-$\epsilon$-settle-optimal, $P_F$, no greater than $\delta$. 

3
4.1 Failure 2

For our analysis we will use as our failure point when the expected increase in income for the bad arm surpasses the expected increase in income of the good arm. We denote this condition as Failure 2. So long as the expected increase in income of the good arm is greater than the expected increase in income of the bad arm then we know that we are making progress towards settling on the good arm in expectation. Let \( p_g \) stand for the probability that the good arm is pulled and \( 1 - p_g \) the probability that the bad arm is pulled. Define said expectation as:

\[
E[\text{Increase in good arm income relative to bad arm}] = p_g \cdot a - (1 - p_g) \cdot b < p_g - (1 - p_g) \cdot (1 - c)
\]

Solving for the probabilities under which this remains true we get the following inequalities:

\[
\begin{align*}
p_g &> \frac{1 - \epsilon}{2 - \epsilon} \\
1 - p_g &< \frac{1}{2 - \epsilon}
\end{align*}
\]

Thus, we know that so long as \( p_g \) and \( 1 - p_g \) satisfy the above inequalities then we have not yet failed according to the above failure point.

4.2 Failure Implications

Because proving sort-of-\( \epsilon \)-Settle-Optimal solutions is hard for Sort-Of-I-Learning we note the following implications:

**Failure Implication 1:** A solution, \( D \), returned by Sort-Of-I-Learning that is not sort-of-\( \epsilon \)-settle-optimal is Failure 2.

Suppose that a solution, \( D \), returned by Sort-Of-I-Learning is not sort-of-\( \epsilon \)-settle-optimal. By construction the \( D \) returned by Sort-Of-I-Learning must satisfy both (1) and (3). Thus, \( D \) must not satisfy (2), meaning that the reward of the settled on arm, \( s \), is less than or equal to \( \min(1 - \epsilon, \frac{2}{3}) \). That is, \( s \) in this case is the sub-optimal arm. Additionally, we know that \( s \) is chosen with probability \( 1 - \min(1 - \epsilon, \frac{2}{3}) = \max(1 - \epsilon, \frac{2}{3}) \). There are two cases to consider: when \( \epsilon < \frac{1}{3} \) and when \( \epsilon \geq \frac{1}{3} \).

- **Suppose** \( \epsilon < \frac{1}{3} \). Then by assumption of being sort-of-\( \epsilon \)-settle-optimal the bad arm has probability at least \( \max(1 - \epsilon, \frac{2}{3}) = 1 - \epsilon \). Since \( \epsilon > \frac{1}{3} \) we know that \( 1 - \epsilon \geq \frac{1}{2 - \epsilon} \) and so the bad arm must have probability at least \( \frac{1}{2 - \epsilon} \), thereby satisfying Failure 2.

- **Suppose** \( \epsilon \geq \frac{1}{3} \). Then by assumption of being sort-of-\( \epsilon \)-settle-optimal the bad arm has probability at least \( \max(1 - \epsilon, \frac{2}{3}) = \frac{2}{3} \). Additionally, if \( \epsilon > \frac{1}{3} \) we know that the bad arm has probability at least \( \frac{1}{2 - \epsilon} \) since \( \frac{1}{2 - \epsilon} < \frac{2}{3} \) so Failure 2 is satisfied.
Failure Implication 2: A solution, $D$, returned by Sort-Of-I-Learning that is Failure 2 has gained at least $\frac{I_0\epsilon}{1-\epsilon}$ income on the sub-$\epsilon$-optimal arm by round $|D|$ in addition to the $I_0$ that it started with.

Suppose that a solution, $D$, returned by Sort-Of-I-Learning is Failure 2. By definition the bad arm must have probability at least $\frac{1}{2\epsilon}$. We can now solve for the minimum $x$ for which this probability will be true which is gotten if the good arm receives no income and all income goes towards the bad arm, i.e.:

\[
\frac{I_0 + x}{2I_0 + x} \geq \frac{1}{2 - \epsilon}
\]

\[
\ldots
\]

\[
x \geq \frac{I_0\epsilon}{1 - \epsilon}
\]

Since the arm started with $I_0$, it has accrued at least $x + I_0 \geq \frac{2I_0\epsilon}{1 - \epsilon}$ if it has probability $\frac{1}{2\epsilon}$.

Let $P'_F$ be the probability that we fail by allowing the sub-optimal arm to reach $\frac{2I_0\epsilon}{1 - \epsilon}$ income. By Failure implication 1 and 2, we know that $P_F$ (the probability that a solution is not not sort-of-$\epsilon$-settle-optimal) is such that $P_F \leq P'_F$. Thus, it will be sufficient for our analysis to bound $P'_F$.

### 4.3 Gambler’s Ruin

Next, some background about gambling. The Gambler’s Ruin problem is a well studied stochastic process. In it, a gambler starts with some amount of money $n$ and wishes to play until she makes $n + m$ money. At each step the gambler has a probability of $p$ of winning 1 money and a probability of $q$ of losing 1 money. The gambler plays until she has 0 money (i.e. is “ruined”) or until she makes at least $n + m$ money. Many properties of the process are well studied but for our purposes we need only note the following:

\[
P(Win \ n+m \ without \ ever \ ruining) = \left( \frac{q}{p} \right)^n - 1 \leq \left( \frac{p}{q} \right)^m
\]

Critically the upper bound is independent of $n$, the number of dollars that the gambler starts with. [1]

### 4.4 Gambler’s Ruin Conjecture

Because we will be dealing not with winning or losing 1 money but rather winning and losing real-valued amounts of money we now state a conjecture regarding a generalization of the Gambler’s Ruin where the amount of money won or lost is real-valued rather than just 1 and -1. The validity of this conjecture is left as an exercise to the reader.
Gambler’s Ruin Conjecture: Given a Gambler’s Ruin process with initial money $n$ and stopping condition of $n + m$ if the gambler wins $a$ and $-b$ in $\mathbb{R}$ with probabilities $p$ and $q$ respectively, 

$$P(\text{Win } n + m) \leq \left( \frac{a * p}{b * q} \right)^m$$

Note that this conjecture is consistent with the vanilla Gambler’s Ruin. If $a = b = 1$ as it does in the vanilla Gambler’s Ruin then the conjecture states that the probability that the gambler makes $n + m$ dollars before getting ruined is upper bounded by $\left( \frac{p}{q} \right)^m$ which is the usual gambler’s ruin bound.

4.5 Lemma 1

We begin by relating the failure probability, $P_F$, to the initial income of the arms.

**Lemma 1:** $P_F \leq r^{\frac{I_0}{\epsilon}}$ where $0 < r < 1$.

We can view Failure 2 occurring for Sort-Of-I-Learning as a gambler’s ruin process for the sub-optimal arm where the sub-optimal arm is trying to gain an additional $\frac{I_0}{\epsilon}$ income. We need not specify the amount of money we start with since the bound we are using is independent of this quantity. Since Failure 1 implies Failure 2 and we are stopping when Failure 2 is satisfied, we know that Failure 1 is never satisfied until Failure 2 is satisfied and the process completes. In particular, the good arm always has probability at least $\frac{1}{2+\epsilon}$ and the bad arm has probability no greater than $\frac{1}{2+\epsilon}$. Thus, while the probabilities of each arm are, in fact, not i.i.d. throughout the stochastic process we may fix them at their bounds since we have appropriate upper and lower bounds of the bad and good arm respectively throughout the process. Thus, we are dealing with a (generalized) Gambler’s Ruin where $p \leq \frac{1}{2-\epsilon}$, $1 - p \geq \frac{1-\epsilon}{2-\epsilon}$, $m = \frac{I_0}{\epsilon}$, $a = 1 - \epsilon$ and $b = 1$. Note that $\frac{a * p}{b * q}$ is

$$\frac{a * p}{b * q} \leq \left( \frac{(1 - \epsilon) \frac{1}{2-\epsilon}}{(1)(\frac{1-\epsilon}{2-\epsilon})} \right) = 1$$

Thus, $\frac{a * p}{b * q}$ is simply some value strictly less than 1. Let $r$ stand for this value.

Plugging these values into the Gambler’s Ruin Conjecture bound gives:

$$P_F' \leq (r)^{\frac{I_0}{\epsilon}}$$

Since as established before $P_F \leq P_F'$, we conclude that $P_F \leq (r)^{\frac{I_0}{\epsilon}}$.

4.6 Lemma 2

We now use Lemma 1 to conclude that I-Learning initializes its income to be sufficiently large to be Sort-Of-I-Learning is sort-of-$\epsilon$-settle-optimal.
Lemma 2: Sort-Of-I-Learning is sort-of-$\epsilon$-settle-optimal with probability $1 - \delta$

By Lemma 1 we know that the failure probability of I-Learning is upper bounded by a function of the initial incomes of the arms, namely $P_F \leq \frac{\log\frac{1}{\epsilon}}{r}$. We now note that since $I_0 = \frac{\log\left(\frac{1}{1-\epsilon}\right)}{r}$ in Sort-Of-I-Learning, we have that the failure probability is upper bounded by $\frac{\log\left(\frac{1}{1-\epsilon}\right)}{r} = \delta$. That is, $P_F \leq \delta$. Thus, the probability that Sort-Of-I-Learning fails to be sort-of-$\epsilon$-settle-optimal is no greater than $\delta$ meaning that Sort-Of-I-Learning is sort-of-$\epsilon$-settle-optimal with probability at least $1 - \delta$.

4.7 Future Work

We are interested in more rigorously verifying the following in the above proof:

- The Gambler’s Ruin Conjecture
- That upper and lower bounds can be substituted for $p$ and $q$ in the Gambler’s Ruin inequality as is done above.
- That no weirdness is happening with $r$ not being knowable a priori by Sort-Of-I-Learning.

We are interested in the following future work:

- Polynomially reducing sort-of-$\epsilon$-settle-optimal to more common correctness conditions such as those of [3] and [2].
- Extending to our result to $k$-armed bandits.
- Showing that the worst case is when the bad arm has exactly $1 - \epsilon$ probability.
- Extending our work to arms with stochastic payoffs (but fixed payoff rates).

References


