CSI 4900: Honours Project Report

Wei Hu
whu061@uottawa.ca

University of Ottawa — June 17, 2019

1 Introduction

An Erdös-Selfridge-Spencer game consists of an attacker A, a defender D, and a game state represented by an array S, where $S[i]$ is the number of pieces on the board that is i levels away from the top of the board. The board is zero-index so that a piece which is at level 0 will attain tenure and provide a point for the attacker, if not destroyed by the defender this turn.

Definition 1.1. Let v be a value function, which maps a level i to a real number representing the value of a piece at level i. A value function can be represented by an array $(w_0, w_1, w_2, ..., w_k)$ where:

$$v(i) = w_i.$$

Definition 1.2. Let S be a state on a board with k levels. The value function applied to S is defined as:

$$v(S) = \sum_{i=0}^{k} S[i]v(i).$$

A defender would apply its value function to the two sets partitioned by the attackers, and it would destroy the set it deems to be more valuable.

Theorem 1.1. *The optimal value function v_* for the defender is defined as:*

$$v_*(i) = \frac{1}{2^{i+1}}. \tag{1}$$

2 Biased Defenders

We observe that the optimal value function satisfies the property that $v(i) = 2v(i+1)$. By deviating from this equality, we can create sub-optimal defenders. These defenders either overvalue pieces that are close to the top of the board, or those that are the furthest away. This bias is dependent on the direction of the inequality.

Definition 2.1. A farsighted defender is a defender whose values function satisfies:

$$v(i) < 2v(i+1). \tag{2}$$

for all $0 <= i < k$, where k is the length of the board.

A farsighted defender disproportionally value pieces that are close to the bottom of the board.

Definition 2.2. A nearsighted (myopic) defender is a defender whose value function satisfies:

$$v(i) > 2v(i+1). \tag{3}$$

for all $0 <= i < k$, where k is the length of the board.

A nearsighted defender will overvalue pieces that are near the top of the board.

3 Project Summary

We intend use various reinforcement learning (RL) techniques to train attackers to exploit the sub-optimality of biased defenders, both nearsighted and farsighted. We observe that playing optimally against a biased defender is more complicated than playing optimally against an optimal defender:

Theorem 3.1. *When playing against an optimal defender, the optimal approach for the attacker is to try to make the value according to v_* of the two partitioned sets as close as possible.*

This approach maximizes, at each turn, the value according to v_* of the surviving pieces. A linear algorithm exists for finding the partition described in Theorem 3.1. The theorem can be proved by:

1. Showing that when playing against an optimal defender, the best score that can be achieved for a starting state S is $\lfloor v_*(S) \rfloor$.
2. Showing that playing according to Theorem 3.1 guarantees that we will score at least $\lfloor v_*(S) \rfloor$.

However, when playing a biased defender, it is possible for the real value of the surviving subset to be larger than that of the one destroyed. Due to its bias, a set could have higher value than another set according to v_*, but be deemed less valuable according to the defender's v. As a result, we can be certain that the approach described in Theorem 3.1 is not optimal.

The idea here is to examine how various RL approaches such as Q-learning, policy gradient, and actor critic can "discover" more optimal approaches against these defenders.

3.1 Challenges

In the original paper first introducing Erdös-Selfridge-Spencer games as test environments for RL algorithms (Raghu et al. 2017), the focus was mainly on training defender agents. Defenders have an action space of size two, while the attackers, which is what we will mainly be training, has an $O(2^n)$ action space.

While the paper provides a method to train attackers, it employs a techniques through which the attacker's action space is made linear. This reduction not only substantially increased the proportion of optimal moves (making the game easier for the attacker), but the reduced action space is also only guaranteed to contain the optimal move against the optimal defender.

Furthermore, rewards are sparse, and we only get feedback whenever a piece gets tenure. Since most of the attacker's action are bad and lead to no reward, it can be difficult to gather good training feedback.

4 Algorithms

We were able to analytically prove properties about biased defenders. We provide two algorithms which maximizes the real value of the surviving pieces at the end of each turn against any nearsighted or farsighted defender.

Two core ideas behind the algorithms are laid out in the following lemmas:

Lemma 4.1. *If a set of pieces S which are all further away from tenure than a piece at level i has $v_*(S) >= v_*(i)$, then there is a subset S' of S that satisfies $v_*(S') = v_*(i)$.*

Proof. Showing lemma 4.1 is equivalent to showing that for a fraction p of the form $1/2^i$ where $i \in \mathbb{Z}^+$ and a multiset M of fractions which contains only elements of the form $1/2^{i+j}$ where $j \in \mathbb{Z}^+$. If the elements in M adds up to a value greater than p, then there exist a subset of M which adds up to exactly p.

We prove this result by first create a new multiset M', by dividing each element in M by p. Notice the problem is equivalent to showing that a subset of M' adds up to 1.

Since, the sum of M' must be greater than 1, and M' contains only powers of $1/2$, lemma 1.X (my proof of the splitting lemma) completes the proof. □

Lemma 4.2. *Let v be the value function of a farsighted defender. Let A, B be two sets of pieces. If every piece in B is closer to tenure than any piece in A, and B has greater value according to v, then B has greater value according to v_* as well.*

Proof. If A is greater than B according $v*$, then we can continually apply lemma 4.1, and map each piece in B to a subset of A, without ever using the same piece from A more than once (and there will be pieces left over in A, not mapped to anything in B). In each of these mappings, each piece from the subset of A will be valued according to $v*$ as being worth $1/2^i$ of the value of the piece in B, where i is the number of levels the piece in A is away from the piece in B.

According to v, however, the same piece would be worth more than $1/2^i$ the value of the piece in B, since $v(i) < 2v(i+1)$ for any farsighted defender. Thus, the pieces in the set from A will be valued less by v than the piece in B. Applying this to every set in the mapping, we conclude A will be valued higher than B by v. We complete the proof by contraposition. □

Similarly, we also have:

Lemma 4.3. *Let v be the value function of a nearsighted defender. Let A, B be two sets of pieces. If every piece in B is further from tenure than any piece in A, and B has greater value according to v, then B has greater value according to v_* as well.*

4.1 Farsighted Defenders

Theorem 4.4. *The following algorithm maximizes the real value of the surviving pieces at the end of each turn against any farsighted defender.*

```
Algorithm 1: Minimizing $v_*$ Value of Destroyed Set Against Nearsighted Defenders

  Input: a board position $S$ with $k$ levels, and the value function of a nearsighted defender $v$
  Result: a partition of the board $(S_1, S_2)$ which guarantees the subset that the defender will
          destroy has the lowest possible value according to $v_*$

  Beginning from the row furthest from tenure, and going from left to right, we start by uniquely
    labeling each piece on the board with an integer starting at one.
  $\gamma \leftarrow \frac{v(S)}{2}$;
  $resulting\_set \leftarrow$ an array of zeroes of length $k$;
  $cumulative \leftarrow$ an array of tuples $(cd, level)$.
  The tuple at the $i$-th index has for $cd$ the sum of the biased values of all the pieces up to and
    including the $i$-th piece, and $level$ is the level of that $i$-th piece;
  $current\_index \leftarrow cumulative.length - 1$;

  while $\gamma > 0$ and $current\_index > 0$ do
      if $cumulative(current\_index - 1).cd < \gamma$ then
          $best\_set[cumulative(current\_index - 1).level]{+}{+}$;
          $\gamma {-}{=} v(cumulative(current\_index - 1).level)$;
      end
      $current\_index{-}{-}$;
  end

  return $(resulting\_set, (S - resulting\_set))$;
```

Proof. Label each of the pieces with an integer in the same manner that is described in the algorithm. Essentially, the algorithm looks for the piece with the largest index which satisfies the condition that the sum of the biased values of the current piece and all the pieces that come after it is greater than γ.

We can then be certain that the current piece can be contained in the set that we want the defender to destroy, by showing that there is no reason to include any elements with a smaller index. By contradiction, assume there is some set S that contains one or more element of smaller index and has minimal value according to v_* and is greater than γ according to v.

We will show that we can replace those elements with index smaller than that of the current piece with elements with larger index than the current piece to form a set with equal or less value according to v_* and equal or more value according to v. Let T be a set containing the current element and all elements with larger index than the current element. Subtract $S \cup T$ from S, T to form S' and T' respectively. Since we are claiming S is optimal, then $v(S') <= v(T')$.

Consider, only a single element inside S' which has index smaller than the current piece...

□

4.2 Nearsighted Defenders

Theorem 4.5. *The following algorithm maximizes the real value of the surviving pieces at the end of each turn against any nearsighted defender.*

Algorithm 2: `Minimizing` v_* `Value of Destroyed Set Against Nearsighted Defenders`

```
Input: a board position $S$ with $k$ levels, and the value function of a nearsighted defender $v$
Result: a partition of the board $(S_1, S_2)$ which guarantees the subset that the defender will
        destroy has the lowest possible value according to $v_*$

$\gamma \leftarrow \frac{v(S)}{2}$;
$best\_set \leftarrow S$;
$best\_value \leftarrow v_*(S)$;
$current\_set \leftarrow$ an array of zeroes of length $k$;
$current\_value \leftarrow 0$;

for $i \leftarrow 0$ to $k$ do
    for $piece \leftarrow 0$ to $S[i]$ do
        if $v(i) < \gamma$ then
            $current\_set[i]{+}{+}$;
            $\gamma {-}{=} v(i)$;
            $current\_value {+}{=} v_*(i)$;
        end
        else
            if $current\_value + v_*(i) < best\_value$ then
                $best\_value \leftarrow current\_value + v_*(i)$;
                $best\_set \leftarrow current\_set$;
                $current\_set[i]{+}{+}$;
            end
            $break$;
        end
    end
end

return $(best\_set, (S - best\_set))$;
```

Before we go about proving Theorem 4.5, we establish:

Lemma 4.6. *Let there be no pieces on the board which have greater value than γ according to some nearsighted v. Let i be the smallest i such that the i-th level is not empty. To form a subset of M with value of at least γ according to v, while minimizing the value of the subset according to v_*, we are guaranteed to be able to include a piece from level i.*

Proof. Let M not contain one of the pieces on level i. Since $v(S) >= \gamma >= v(i)$. Then we can apply lemma 4.3 and conclude $v_*(S) > v_*(i)$. Then, by lemma 4.1, we know there is a subset M' of M with a value of exactly $v_*(i)$. If we were to replace M' in M with a piece from level i, we do not change the value according to v_* but we increase or maintain the value according to v (Modus Tollens of lemma 4.3). □

We now prove Theorem 4.5:

Proof. The idea behind the algorithm is that as we iterate through the board, starting with the more valuable pieces (according to v_*), which we are certain we can add to current_set due to lemma 4.6. Every time we come across a piece whose value exceeds the remaining γ, we flag... □

5 Machine Learning

5.1 Q-Learning

The traditional Q-learning approach using a Q-table is only feasible for small tables.

5.1.1 Deep Q-Learning

We propose a method for using a neural network to represent the Q table, using a method to compress the action space?

5.2 Policy Gradient

Take a series of actions, and determine feedback based on cumulative reward.

5.3 Actor Critic

One neural network to approximate value function, and another to select actions. Multiple continuous action space problem?

6 Observations

L. Moura: Train attacker for a predetermined number of turns (or time), and observe how well the trained defender performs relative to expected potential, and the "optimal" algorithm. A good metric is:

$$\frac{optimal\_score - obtained\_score}{r_*(starting\_state)}$$

Metrics are gathered both during training, and after training. During training, track the training gain by measuring how much the q values are being changed (make the algorithm automatically increase the number of levels, once we have convergence), and track the progression of the previous metric. After training (without exploring), play a series of games and plot the potential, the obtained score, and the score obtained by the optimal algorithm.

Investigate relationship between γ and the difference between potential score and optimal score?

Is there anything else?