n-Move Memory Evolutionarily Stable Strategies for the Iterated Prisoner's Dilemma

Andrew Clemens

August 15, 2013

Abstract

Cooperation games are competitions where there is a conflict between personal and societal gain. No matter what others choose to do, personal gain is maximized by defecting; however, mutual defection is worse personally than mutual cooperation. The iterated prisoner's dilemma is one of the most simple and widely studied such games. This is a round-robin style tournament where in each round, players can choose to cooperate with or defect from their opponent, and previous actions are remembered and used to determine a player's next move. In the evolutionary form of the iterated prisoner's dilemma, players that do poorly will die out, and players that do well will be copied. Their copies are sometimes mutated, resulting in new strategies that are usually worse but are sometimes better. It's been shown several times that there are no evolutionarily stable strategies (ESS) in this game; however, Lorberbaum et. al. showed that this finding is a mathematical technicality which only happens because these strategies are too perfect. ESS's start to appear when we introduce a constant e, where $0 < e < \frac{1}{2}$, and instead of the strategies cooperating with a probability of 0 or 1, they now cooperate with a probability of e or $1 - e$. More specifically, they showed that there are exactly three one-move memory ESS's: Pavlov, which generally cooperates if and only if both players simultaneously either cooperate or defect; Grudge, which cooperates if and only if both players simultaneously cooperate; and AllD, which generally defects after all histories. AllD is always an ESS, but the other two rely on certain payoff conditions to be evolutionarily stable. This paper hopes to generalize the results of Lorberbaum by allowing the strategies to look back more than one round, concluding that a modified Pavlov strategy exists as an ESS for any payoff conditions.

1 Introduction

A commonly analyzed game in game theory is known as the prisoner's dilemma. This is a two player game where each player has the choice of either cooperating or defecting. There are four different payoffs they can get, depending on what they do or what their opponent does. They are:

- a reward R, which both players get if they both cooperate;
- a punishment P, which both players get if they both defect;
- a temptation payoff T, which a defector gets against a cooperator; and

- a sucker's payoff S, which a cooperator gets against a defector.

The trick here is that the payoffs satisfy $T > R > P > S$, which means that both players can maximize their payoff by defecting, regardless of what their opponent does. Since $T > R$ and $P > S$, whatever a player's opponent does, it is always better for the player to defect. This is a problem because mutual cooperation is better than mutual defection. By doing what's best for them personally, both players will receive a punishment, when they could have gotten a reward. Betrayal stifles collaboration.

In the iterated version of the prisoner's dilemma, this problem is resolved. The game is played multiple times between two players and the actions in the previous rounds are remembered. Defection is no longer always the best strategy, because the point is to get the opponent to cooperate, something constant defection won't do. When Robert Axelrod held a competition, allowing anyone to submit their strategies for this game, and held a round-robin tournament for the strategies he got, the winner was a strategy called Tit-for-Tat, which cooperates on the first round and then simply repeats the actions of the opponent on the previous round. However, the purpose of this paper is not to find the best strategy, in a round-robin tournament, but to find strategies that are evolutionarily stable. Although tit-for-tat is not an ESS, the idea of rewarding cooperation and punishing defection does form the basis of many ESS's.

One could simulate the evolution of strategies for this game, in order to look for strategies that are evolutionarily stable. First, a random population of strategies is generated. Then the strategies play against each other in a round-robin style tournament. Strategies that do poorly die out, and strategies that do well copy and mutate, which introduces new strategies. Eventually, the population will become filled with several copies of identical strategies, and when a different strategy gets introduced it dies out immediately. When this happens we have what is called an evolutionarily stable strategy, or ESS.

The iterated prisoner's dilemma actually has no ESS's. This simulation will just cause the population to change constantly without settling on one strategy. It was shown by Lorberbaum that if random noise is introduced, and we have a probability that the strategies will make mistakes (cooperate when they mean to defect or vice versa), ESS's do appear. Say we have a probability constant e, where $0 < e < \frac{1}{2}$, then the strategies cooperate with probability e or $1 - e$ instead of probability 0 or 1. Lorberbaum only looked at strategies that can look at the previous round and make decisions based on that, and he found that of the 16 possible such strategies, three of them were ESS's:

- Pavlov-e: cooperates with probability $1 - e$ if and only if both it and it's opponent cooperated on the previous round or defected on the previous round, otherwise it defects with probability $1 - e$ (only an ESS if $2R > T + P$)

- Grudge-e: cooperates with probability $1 - e$ if and only if both it and it's opponent cooperated on the previous round, otherwise it defects with probability $1 - e$ (only an ESS if $R + 2S < 3P$)

- AllD-e: defects with probability $1 - e$ after all histories (ESS under all payoff conditions)

2 n-Move-Memory Strategies

We can generalize Lorberbaum's results by allowing the strategies to have a longer memory.

Definition 1. *An n-move-memory strategy is a strategy that is allowed to look at the past n rounds to make its decision.*

Because each round has 4 possible plays, and each history has two possible responses, the number of n-move-memory strategies is 2^{4^n}. This number gets very large very quickly, as n increases, and that can make these strategies difficult to work with. There are only 16 1-move-memory strategies so it's easy to find all ESS's, but there are about 65,000 2-move-memory strategies, and the number of 3-move-memory strategies is about 18 billion billion.

2.1 A Strategy's Genome

We characterize a strategy by giving it a set of instructions, or what to play after every possible history. We'll use the example of a two-move-memory strategy X. There are 16 possible histories, so we'll first list out all of them, and after those we'll say what the strategy will do after that particular history.

X	D	D	C	X	D	D	D
opp	D	D		*opp*	D	C	
X	D	D	D	X	D	D	C
opp	C	D		*opp*	C	C	
X	D	C	D	X	D	C	D
opp	D	D		*opp*	D	C	
X	D	C	C	X	D	C	D
opp	C	D		*opp*	C	C	
X	C	D	D	X	C	D	C
opp	D	D		*opp*	D	C	
X	C	D	C	X	C	D	D
opp	C	D		*opp*	C	C	
X	C	C	C	X	C	C	C
opp	D	D		*opp*	D	C	
X	C	C	C	X	C	C	C
opp	C	D		*opp*	C	C	

By assigning $C = 1$ and $D = 0$, we can create a genome for strategy X like this:

$$X = \{1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1\},$$

These genomes can easily be generalized into any n-move-memory strategy.

2.2 Calculating the Expected Payoff

We generalize the method from Nowak 1995 to calculate the expected payoff between two strategies. We'll use the example of these two random 2-move-memory strategies:

$$X = \{1,0,0,1,0,0,1,1,0,1,1,1,0,0,0,1\},$$

and

$$Y = \{0,1,0,0,0,0,1,0,0,1,1,0,1,0,1,1\}.$$

We want to find the average long-term payoff per round between these two strategies, where the number of rounds approaches infinity and the probability of error approaches zero. There are 3 runs that these two strategies can get caught up in. They are, along with the payoffs X will receive from them:

$$\begin{array}{lc}
\mathbf{A} & \begin{array}{c|cccccc} X & D & D & C & D & C & C \\ \hline Y & D & D & D & C & C & D \end{array} & \frac{2P+2S+T+R}{6} \\
\mathbf{B} & \begin{array}{c|cc} X & C & D \\ \hline Y & C & D \end{array} & \frac{R+P}{2} \\
\mathbf{C} & \begin{array}{c|c} X & C \\ \hline Y & C \end{array} & R
\end{array}$$

There are twelve ways a perturbation can happen in state **A**. Seven of them lead back to state **A**, four lead to state **B** and one leads to state **C**. In state **B**, there are four ways a perturbation can happen, and they all lead to state **A**. There are two ways a perturbation can happen in state **C**, and they both lead to state **A**. So, the transition matrix between the states looks like this:

$$\begin{array}{ccc}
 & & \text{from} \\
 & & \begin{array}{ccc} \mathbf{A} & \mathbf{B} & \mathbf{C} \end{array} \\
\text{to} & \begin{array}{c} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \end{array} & \begin{pmatrix} \frac{7}{12} & 1 & 1 \\ \frac{1}{3} & 0 & 0 \\ \frac{1}{12} & 0 & 0 \end{pmatrix}
\end{array}$$

This matrix has a right eigenvector of $\{\frac{12}{17}, \frac{4}{17}, \frac{1}{17}\}$, and taking the dot product of that with the three payoffs, we get our expected payoff for strategy X: $V(X|Y) = \frac{6P+4S+2T+5R}{17}$.

2.3 Finding a Strategy's Best Response

Every strategy X has a best response $Y*$. This is the strategy that satisfies, for any other strategy Y and Z, $V(Y*|X) \geq V(Y|X)$ and if $V(Y*|X) = V(Y|X)$, then $V(Y*|Z) \geq V(Y|Z)$. If $Y* = X$, then X is an ESS. We'll be looking for ESS's by finding strategies with themselves as their own best response.

Because the number of possible strategies becomes very high when n is increased, a brute force algorithm for finding a strategy's best response becomes difficult. To simplify the search, we need to make rules that will narrow it down. Most of these follow naturally from the rules set by Lorberbaum, while others take some altering.

Rule 1. *The best response to any n-move-memory strategy is another n-move-memory strategy.*

Proof. Say X is an n-move-memory strategy with optimal opponent Y. X has 4^n states, corresponding to how X behaves after each of the 4^n histories. Say for some history h, X will play $X(h)$. Y has some best response to this, and that response will not change for any of the plays before h. This is true for all h, meaning Y would want to look back the same number of rounds as X would, making Y an n-move-memory strategy. □

Rule 2. *The best response to n-move-memory strategy Y only needs to determine if it will cooperate or defect based on the previous $n-1$ rounds, as well as what Y will do in the current round.*

Proof. Say h_n is the n-move history, or the previous n plays. Y will use this history to decide it's next play. After it's done that, the least recent play in h_n doesn't matter, and doesn't decide how well the opponent does. So, the opponent should do whatever will give him the best payoff in the long run, which would be determined only by the history h_{n-1} and Y's next play. □

Now, the plays Y needs to look at are the previous $n-1$ rounds, or $2n-2$ plays, as well as whatever X is planning on doing in the current round. This is a total of $2n-1$ plays, each with 2 plays that could happen on them and 2 responses for Y. This cuts down the number of possible best responses to an n-move-memory strategy to $2^{4^{2n-1}}$, or the square root of the total number of possible strategies.

3 Pavlovian strategies

Definition 2. *A Pavlovian strategy P_n is an n-move memory strategy that cooperates if and only if for some k where $0 \leq k \leq n$, both strategies cooperated on the past k moves and defected on all $n-k$ moves before that.*

P_1 is the Pavlov strategy from above. Pavlovian strategies do well because they can "communicate" with each other, and tell each other that they're the same strategy, and they want to cooperate to improve their own population. Additional similar ESS's can be found by looking for other strategies that do this.

Theorem 1. *If $(n+1)R > T + nP$, then P_n is an ESS.*

Proof. Say $Y \in S$, and $Y \neq P_n$. We know that $V(P_n|P_n) = R$. If Y always generally plays the same moves as P_n, then $V(Y|P_n) = R$. (Now show the other part of ESS)

Now say at some point Y's behavior deviates from that of P_n; that is, Y either cooperates where P_n defects or Y defects where P_n cooperates. First, say Y cooperates and P_n defects. Then Y will receive payoff S, and for the next n rounds P_n will defect, giving Y a payoff no higher than nP. So for those $n+1$ rounds, Y will receive an average payoff of $\frac{S+nP}{n+1}$, which is certainly less than R because $S < R$ and $P < R$. This means that there is no way Y can get a higher payoff than P_n afte0r Y cooperates and P_n defects.

Now, say Y defects where P_n cooperates. From this, Y receives payoff T and P_n will defect for the next n rounds, giving Y a payoff of no higher than nP. That means for those $n+1$ rounds, Y receives an average payoff of $\frac{T+nP}{n+1}$. By our assumption, $(n+1)R > T + nP$, or $\frac{T+nP}{n+1} < R$, meaning there's no way Y can get a higher payoff than P_n after Y defects and P_n cooperates. □

Note that $(n+1)R > T+nP \iff n > \frac{T-R}{R-P}$. So, no matter what the payoff conditions are, we can find an n high enough so that P_n is an ESS.

4 Identifier strategies

The reason Pavlov strategies do well is because they are very good at determining if their opponent is another idential Pavlov strategy, and cooperating or defecting accordingly. There are other ways that strategies can do this. Pavlov strategies are one example of an identifier strategy, so called because they can identify if their opponent is an identical opponent to themself. Each identifier strategy starts out with some kind of pattern of defection and cooperation, which it only continues if the opponent matches that pattern (otherwise it defects), and when that pattern ends and the two players are sure that they're the same strategy, they play some other pattern with each other. These patterns do not necessarily need to be as long as the strategy's memory, but the lengths of them do have limitations.

An identifier strategy will be denoted as $I_n^{p,q}$. n is the length of the strategy's memory. p in base 2 is the pattern the strategy will follow, starting after n defections, only continuing to follow the pattern if the opponent does also. q is the pattern the strategy will repeat after p. We can see that $I_n^{0,1} = P_n$ and $I_0^{0,0} = AllD$.

For example, take $I_3^{5,2}$. This strategy has a 3-move-memory. Because $5 = 000101_2$, the strategy first defects three times, then cooperates, then defects, then cooperates. If the opponent does the same thing, then this tells $I_3^{5,2}$ that it could be playing against an identical strategy, so it goes into the q phase. $2 = 10_2$, so the strategy will then cooperate, then defect, and then continue this forever, as long as the opponent does the same thing. After any communication errors, the game will always default back to the q phase, so $V(I_3^{5,2}|I_3^{5,2}) = \frac{P+R}{2}$. Generally:

$$V(I_n^{p,q}|I_n^{p,q}) = \frac{P\left(\lfloor 1+\log_2 q\rfloor - \mathrm{weight}(q)\right) + R \times \mathrm{weight}(q)}{\lfloor 1+\log_2 q\rfloor},$$

where weight(q) is the Hamming weight of q.

Under what conditions $I_n^{p,q}$ is an ESS would be a possibility for future study.

5 Grudgian strategies

Definition 3. *A Grudgian strategy G_n is an n-move-memory strategy that cooperates if and only if the play from n rounds previous (or the least recent play that G_n can remember) was G_n and the opponent both cooperating. Otherwise, G_n defects. What the players did on the other $n-1$ rounds is irrelevant.*

In a game between G_n and another G_n if either player defects, then both will continue to defect after multiples of n periods. The only circumstances in which both players will cooperate is if it is one of the initial $n-1$ rounds, it is n rounds after both have cooperated, or there is an extremely improbable simultaneous communication error which we're calling too improbable to consider. After a while, a communication error will have occured enough times that neither side is cooperating, and the long-term payoff the players will receive is the punishment. So, $V(G_n|G_n) = P$.

G_n where $n > 1$ could be thought of as a "delayed" Grudge strategy. To simplify the way that G_n plays against its opponents it makes sense to think of a game against G_n as n separate games against G_1, all mixed together. This way, it's easier to find a best opponent.

5.1 Responses to G_n

As a notation, during a game with G_n as a player, we say that we are in state T_m if G_n cooperated for m of the past n rounds ($0 \leq m \leq n$). As will be made clear later, the different possible permutations of T_m are interchangeable, so we only need to look at the $n+1$ possible states.

Definition 4. *G_n^α is a response to G_n. If the players are in state T_m, then G_n^α operates as G_n if $m \geq \alpha$ and AllC if $m < \alpha$.*

(Note that $G_n^0 = G_n$ and $G_n^{n+1} = AllC$.)

The idea behind G_n^α is that, depending on the payoff conditions, it might be beneficial to an opponent to try and get G_n to cooperate only if G_n is cooperating below some threshhold. Otherwise it might be a waste for an opponent to get G_n to cooperate further.

Now to find $V(G_n^\alpha|G_n)$, we have to create a general transition matrix A. Entry $A_{i,j}$ of this matrix will be the probability of progressing from state T_j to state T_i. If $i < j-1$ or $i > j+1$, $A_{i,j} = 0$. Otherwise:

$$A_{i,j} = \begin{cases} \frac{i+1}{n}, & \text{if } i = j-1 \\ \frac{n-i}{n}, & \text{if } i = j \text{ and } j \geq \alpha \\ \frac{n-i}{2n}, & \text{if } i = j \text{ and } j < \alpha \\ 0, & \text{if } i = j+1 \text{ and } j \geq \alpha \\ \frac{n-i+1}{2n}, & \text{if } i = j+1 \text{ and } j < \alpha \end{cases}$$

Next is to find the eigenvector, or the vector $\overrightarrow{v}$ such that $A\overrightarrow{v} = \overrightarrow{v}$. So for all i, where $0 \leq i \leq n$, we want $\overrightarrow{v}_i = \sum_{j=0}^{n} \overrightarrow{v}_j A_{i,j} = \overrightarrow{v}_{i-1} A_{i,i-1} + \overrightarrow{v}_i A_{i,i} + \overrightarrow{v}_{i+1} A_{i,i+1}$. Or:

$$\begin{aligned} \overrightarrow{v}_i &= \frac{1}{1 - A_{i,i}} \left[A_{i,i-1}\overrightarrow{v}_{i-1} + A_{i,i+1}\overrightarrow{v}_{i+1}\right] \\ &= \begin{cases} \frac{(i+1)\overrightarrow{v}_{i+1}}{i} & \text{if } i > \alpha \\ \frac{(-i+n+1)\overrightarrow{v}_{i-1} + 2(i+1)\overrightarrow{v}_{i+1}}{2i} & \text{if } i = \alpha \\ \frac{(-i+n+1)\overrightarrow{v}_{i-1} + 2(i+1)\overrightarrow{v}_{i+1}}{i+n} & \text{if } i < \alpha \end{cases} \end{aligned}$$

First, we know that $\overrightarrow{v}_i = 0$ when $i \geq \alpha$ because G_n^α won't bother to increase cooperation when cooperation is already above α, so we can add that to the equation.

$$\overrightarrow{v}_i = \begin{cases} 0 & \text{if } i > \alpha \\ \frac{(-i+n+1)\overrightarrow{v}_{i-1}}{2i} & \text{if } i = \alpha \\ \frac{(-i+n+1)\overrightarrow{v}_{i-1} + 2(i+1)\overrightarrow{v}_{i+1}}{i+n} & \text{if } i < \alpha \end{cases}$$

We'll show through an inductive argument that for any i where $0 < i \leq \alpha$, $v_i = \frac{(-i+n+1)\overrightarrow{v}_{i-1}}{2i}$. The base case is $i = \alpha$, for which we already know this is true. We'll go backwards from there. Assume that for some k, where $0 < k \leq \alpha$, $v_k = \frac{(-k+n+1)\overrightarrow{v}_{k-1}}{2k}$. We also know that $\overrightarrow{v}_{k-1} = \frac{(-k+2+n)\overrightarrow{v}_{k-2}+2k\overrightarrow{v}_k}{k-1+n}$, and substituting what we assume $\overrightarrow{v}_k$ to be, we get:

$$\overrightarrow{v}_{k-1} = \frac{(-k+n+2)\overrightarrow{v}_{k-2} + (-k+n+1)\overrightarrow{v}_{k-1}}{k+n-1}$$

After simplifying we get:

$$\overrightarrow{v}_{k-1} = \frac{(-k+n+2)\overrightarrow{v}_{k-2}}{2(k-1)}$$

So by the property of mathematical induction, $v_i = \frac{(-i+n+1)\overrightarrow{v}_{i-1}}{2i}$ for any $0 < i \leq \alpha$.

Now we'll want to find what $\overrightarrow{v}_i$ is in terms of $\overrightarrow{v}_0$. This is easy from looking at the first few values:

$$\begin{aligned}
\overrightarrow{v}_0 &= \overrightarrow{v}_0 \\
\overrightarrow{v}_1 &= \overrightarrow{v}_0 \frac{n}{2} \\
\overrightarrow{v}_2 &= \frac{(n-1)\overrightarrow{v}_1}{4} = \overrightarrow{v}_0 \frac{n(n-1)}{2 \times 4} \\
\overrightarrow{v}_3 &= \frac{(n-2)\overrightarrow{v}_2}{6} = \overrightarrow{v}_0 \frac{n(n-1)(n-2)}{2 \times 4 \times 6} \\
&\vdots \\
\overrightarrow{v}_i &= \overrightarrow{v}_0 \frac{n(n-1)(n-2)\cdots(n-(i+1))}{2 \times 4 \times \cdots \times 2i} \\
&= \overrightarrow{v}_0 \frac{1}{2^i}\binom{n}{i}
\end{aligned}$$

Remember, this is only for $0 \leq i \leq \alpha$. For $i > \alpha$, $\overrightarrow{v}_i = 0$.

To normalize $\overrightarrow{v}$, it would be easier to just set $\overrightarrow{v}_0 = \frac{1}{N}$, and then create a normalizer N.

$$\begin{aligned}
1 &= \sum_{i=0}^{\alpha} \overrightarrow{v}_i \\
&= \sum_{i=0}^{\alpha} \frac{1}{N}\frac{1}{2^i}\binom{n}{i} \\
N &= \sum_{i=0}^{\alpha} \frac{1}{2^i}\binom{n}{i}
\end{aligned}$$

The payoffs for G_n^α in state T_m is $\frac{mR+(n-m)P}{n}$ if $m \geq \alpha$ and $\frac{mR+(n-m)S}{n}$ if $m < \alpha$. Now we can find the value of $V(G_n^\alpha|G_n)$:

$$V(G_n^\alpha|G_n) = \sum_{m=0}^{\alpha-1}\left(\vec{v}_m \frac{mR+(n-m)S}{n}\right) + \sum_{m=\alpha}^{n}\left(\vec{v}_m \frac{mR+(n-m)P}{n}\right)$$
$$= \frac{\left[\sum_{m=0}^{\alpha-1}\left(\frac{1}{2^m}\binom{n}{m}\frac{mR+(n-m)S}{n}\right) + \left(\frac{1}{2^\alpha}\binom{n}{\alpha}\frac{\alpha R+(n-\alpha)P}{n}\right)\right]}{\sum_{i=0}^{\alpha}\frac{1}{2^i}\binom{n}{i}}$$

This equation either grows or shrinks as α grows, depending on the payoff conditions. More specifically, if $R+2S<3P$, $V(G_n^\alpha|G_n) > V(G_n^{\alpha+1}|G_n)$, and if $R+2S>3P$, $V(G_n^\alpha|G_n) > V(G_n^{\alpha-1}|G_n)$.

Rule 3. *A best response to G_n will generally cooperate when G_n generally cooperates, and depending on the payoff conditions should either generally cooperate or generally defect when G_n generally defects.*

Proof. Say Y is the best response to G_n. Y would rather get the reward than the punishment, and defecting when G_n cooperates will get him the temptation payoff immediately but every n rounds after that will get him the punishment, when he could have gotten the reward for those rounds.

For the second part of the rule, G_n doesn't look at the previous $n-1$ rounds. □

This rule implies that G_n's best response is either G_n or *AllC*.

Theorem 2. *If $R+2S<3P$, G_n is always an ESS.*

Proof. G_n's best response is either G_n or *AllC*. $V(G_n|G_n) = P$, and $V(AllC|G_n)$ is equivilant to $V(AllC|G_1) = \frac{R+2S}{3}$, because a game against G_n is the same as n games against G_1. Therefore, G_n's best response is itself when $P > \frac{R+2S}{3}$ or $R+2S<3P$. □

References

[1] Nowak, Martin A., Karl Sigmund, and Esam El-Sedy. "Automata, Repeated Games and Noise." *Journal of Mathematical Biology* 33.7 (1995): 703-722. Print.

[2] Lorberbaum, Jeffrey P., Daryl E. Bohning, Ananda Shastri, and Lauren E. Sine. "Are There Really No Evolutionarily Stable Strategies in the Iterated Prisoner's Dilemma?" *Journal of Theoretical Biology* 214.2 (2002): 155-69. Print.

[3] Axelrod, Robert M. *The Complexity of Cooperation: Agent-based Models of Competition and Collaboration.* Princeton, NJ: Princeton UP, 1997. Print.

[4] Sigmund, Karl. *The Calculus of Selfishness.* Princeton: Princeton UP, 2010. Print.

Reviewer Comments
August 18, 2013
David Housman

Andrew Clemens, “n-Move Memory Evolutionarily Stable Strategies for the Iterated Prisoner's Dilemma,” August 15, 2013.

This draft introduces some interesting new ideas (e.g., identifier strategies and grudge except below a given threshold of cooperation strategies), but there would need to be a large number of additions and modifications for this paper to become worthy of distribution.

I am now fairly convinced that Lorberbaum’s proof is incorrect. Hence, it would be valuable to carefully reconsider the 1-period memory strategies.

Alternatively, it also could be of interest to run simulations of 3-person iterated prisoners’ dilemma games.

When you get back to campus, we should discuss whether you still wish to continue with this project or pursue other interests during the game theory course.

Location Abbreviations: P = page, L = line, S = section, T = theorem

<table>
<tr><th>Location</th><th>Recommendation</th></tr>
<tr><td>Abstract</td><td>Replace “This paper hopes to generalize” with “This paper generalizes”.</td></tr>
<tr><td>P2 L2</td><td>Replace “The trick here is that the payoffs satisfy T > R > P > S, which means that both players can maximize their payoffs by defecting, regardless of what their opponent does.” with “The payoffs satisfy T > R > P > S.”</td></tr>
<tr><td>P2 L9</td><td>Replace “problem is resolved” with “problem is sometimes resolved.”</td></tr>
<tr><td>P2 L9-20</td><td>Somewhere in here you need to explain what the payoffs are in the iterated game. Axlerod used the average payoff over a fixed number of rounds. Lauberbaum and you take the limit as the number of rounds approaches infinity.</td></tr>
<tr><td>P2 L14</td><td>Provide a reference to Axlerod’s book.</td></tr>
<tr><td>P2 L-16</td><td>Provide a reference to Lowerbaum’s paper.</td></tr>
<tr><td>P3 S2.1</td><td>You need to explain why a strategy does not need to specify what will be done in the first two periods. This is because of the payoff function you are using. If you were using an average of a fixed finite number of periods or the discount average over an infinite number of periods with a less than one discount rate, then it would be important for the strategies to specify what happens in the first two periods.</td></tr>
<tr><td>P3 S2.1</td><td>Slightly more explanation is required. Does<table><tr><td>X</td><td>D</td><td>C</td></tr><tr><td>Opp</td><td>C</td><td>D</td></tr></table>mean the focal player defected in the most recent round or two rounds ago?</td></tr>
<tr><td>P3 S2.1</td><td>Either the table or the vector is incorrect because the table has 9 cooperates and the vector has 8 cooperates. Once the correction is made, more explanation</td></tr>
</table>

<table>
<tr><td></td><td>will be needed so that the reader knows the order in which the table is translated into the vector.</td></tr>
<tr><td>P4 S2.2</td><td>I was able to find strategies X and Y that have the properties described (the three runs, the transition counts, and the same number of cooperates), but this required a lot of work on my part. You could help the reader by never introducing the vector notation—I don’t think you ever use it again, and instead displaying both strategies X and Y in the table in the following manner:
<table>
<tr><td>Period</td><td>-2</td><td>-1</td><td>0</td></tr>
<tr><td>X</td><td>D</td><td>C</td><td>D</td></tr>
<tr><td>Y</td><td>C</td><td>D</td><td>D</td></tr>
</table>
You would need to provide some explanation, but the beauty of this approach is that it becomes far easier for a reader to verify your claims.

Of course, the payoff is really the limit as the probability of communication error ε approaches 0. Perhaps this needs to be explained.

Some additional explanation needs to be given for why the right dominant eigenvector is being found and what it means.</td></tr>
<tr><td>P4 S2.3</td><td>Not every strategy has a best response as you define it. Consider the game
<table>
<tr><td></td><td>A</td><td>B</td><td>C</td></tr>
<tr><td>A</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>B</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>C</td><td>1</td><td>1</td><td>0</td></tr>
</table>
No strategy is a best response to any other strategy. For example, A is not a best response to A because $V(B|A) > V(A|A)$, B is not a best response to A because $V(C|A) = V(B|A)$ and $V(C|B) > V(B|B)$, and C is not a best response to A because $V(B|A) = V(C|A)$ and $V(B|C) > V(C|C)$.

Furthermore, best responses need not be unique. Consider the game
<table>
<tr><td></td><td>A</td><td>B</td><td>C</td></tr>
<tr><td>A</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td>B</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>C</td><td>1</td><td>0</td><td>0</td></tr>
</table>
B and C are both best responses to A.

Thus, the first sentence of this section must be removed and definite articles “the” should often be replaced with indefinite articles “a” or “an.”

What is true is that if Y1 and Y2 are both best responses to X, then $V(Y1|Z) = V(Y2|Z)$ for all strategies Z (can you prove this?), that is, Y1 and Y2 do equally well in all environments. However, other strategies may do more or less well against Y1 and Y2 (e.g., see that A does better against C than against B).

All this suggests that the first paragraph should just define best response and ESS. It would be good to add an explanation for why these are reasonable</td></tr>
</table>

	definitions.
P4 S2.3	Always Defect and Always Defect except in round 17 are both best responses to Always Defect (using the limit average payoff). Thus, Rule 1 is false. Rule 1 with the initial "The" replaced with "A" may be true, but given that best responses may not exist, we cannot assume the existence of such a best response as is done in the first sentence of the current proof. In fact, I think that even this modified Rule 1 is false when $2R > T + P$. Let X be the strategy D if both players agreed in the previous period and D otherwise. Let Y1 be the strategy X if both players agreed in the previous period and D otherwise. Let Y2 be the strategy C if both players defected in the previous period and D otherwise. Then $V(Y1\|X) = V(Y2\|X) = T$. For any other 1-move-memory strategy Y, $V(Y\|X) < T$. Let Z1 be the strategy D if both players defected in the previous round and C otherwise. Then $V(Y1\|Z1) = (R + 2T)/3 < T = V(Y2\|Z1)$. Let Z2 be the strategy D if the player cooperated and the opponent defected in the previous round and C otherwise. Then $V(Y1\|Z1) = R > (R + T + P)/3 = V(Y2\|Z1)$. Hence, there is NO 1-move-memory strategy that is a best response to X. Thus, either there is NO best response strategy or any best response strategy must have more than 1 move of memory. In either case, this shows that Lorberbaum's Rule 1 is false! Of course, all of the above payoffs are really limits as the probability of communication error ε approaches 0.
P5 T1	Provide some explanation for why $V(P_n\|P_n) = R$. Again this is for $\varepsilon \to 0$. To find the actual value for a particular $\varepsilon > 0$ might require finding the dominant eigenvector of the transition matrix for the 4^n possible states. Here is a shorter version of one part of the proof: Suppose Y is a strategy such that $V(Y\|P_n) > R$. Since R is the second largest single period payoff, in the long run Y must obtain T with positive probability. When Y obtains T, P_n responds with at least n defections. For P_n to C requires Y to D during n periods. Hence, the most Y can obtain on average is $(T + nP) / (n+1)$, which is less than R according to the argument you already have. As you have noted, the remainder of the proof involves showing that if Y is a strategy satisfying $V(Y\|P_n) = R$ and Z is any strategy, then $V(P_n\|Z) \geq V(Y\|Z)$. Since $V(Y\|P_n) = R$, the earlier argument tells us that Y must almost always be receiving R. This means that Y must choose C while P_n chooses C, and since communication errors are possible, Y must choose D n times in a row "soon" after a period when one or both players choose D. This gives some handle on what Y must look like. Perhaps more could be said. Then perhaps one could assume that $V(P_n\|Z) < V(Y\|Z)$ for some strategy Z, and then come to a contradiction..
P6 S4	The idea of identifier strategies is interesting, and they can be clearly be implemented using finite state machines; however, it is not clear to be that they can usually be implemented as finite memory strategies because they must

	know whether they should be following pattern p or pattern q. Shouldn't $I_n^{0,1} = P_{n+1}$? And shouldn't a communication error reset back to the beginning rather than to the q phase?
P7 middle	$A_{i,j}$ must be incorrect because $\sum_i A_{i,j}$ should equal 1 for each j but it does not. It appears that you may be assuming that the C's in a T_m are randomly distributed across the n periods. I am not going further until there is more explanation and correction.