Symbolic Noise Detection in the Noisy Iterated Chicken Game and the Noisy Iterated Battle of the Sexes

Tsz-Chiu Au
Department of Computer Science
University of Maryland
College Park, MD 20742, U.S.A.
chiu@cs.umd.edu

Sarit Kraus
Department of Computer Science
Bar-Ilan University
Ramat Gan, 52900, Israel
sarit@cs.biu.ac.il

Dana Nau
Department of Computer Science
University of Maryland
College Park, MD 20742, U.S.A.
nau@cs.umd.edu

Abstract

Symbolic noise detection (SND) has been shown to be highly effective in the Noisy Iterated Prisoner's Dilemma, in which an action can accidentally be changed into a different action. This paper evaluates this technique in two other 2×2 repeated games: the Noisy Iterated Chicken Game (ICG) and the Noisy Iterated Battle of the Sexes (IBS).

We present a generalization of SND that can be wrapped around *any* existing strategy. To test its performance, we organized ICG and IBS tournaments in which we solicited several dozen strategies from different authors, and we tested these strategies with and without our SND wrapper. In our tests, SND identified and corrected noise with 71% accuracy in the ICG, and 59% accuracy in the IBS. We believe the reason why SND was less effective in the ICG was because of a tendency for IBS strategies to change more frequently from one pattern of interactions to another, causing SND to make a higher number of wrong corrections. This leads us to believe that SND will be more effective in any game in which strategies often show a stable behavior.

Introduction

The performance of mutliagent systems often depends on the robustness of interaction among agents. But errors can occurs during the interaction, and that could break the premises the agents make about their interaction. This problem is compounded by the fact the agents are self-interested and do not completely trust each other; agents can no longer trust each other because of the mistakes that the other agents make, albeit the mistakes is not intentional but accidental. How to cope with such mistakes is a critical factor in the maintenance of cooperation among agents.

Our previous work on the study of this issue focus on a famous normal-form game called the *Iterated Prisoner's Dilemma* (IPD), which is well known as an abstract model for studying cooperative behavior between two self-interested parties. We studied an important variant of the IPD is the *Noisy* IPD, in which there is a small probability, called the *noise level*, that accidents will occur (Molander 1985). Strategies that do quite well in the ordinary (non-noisy) IPD may do quite badly in the Noisy IPD (Axelrod & Dion 1988; Bendor 1987; Bendor, Kramer, & Stout 1991; Molander 1985; Mueller 1987; Nowak & Sigmund 1990). For example, if two players both use the well-known Tit-For-Tat (TFT) strategy, then an accidental defection may cause a long series of defections by both players as each of them punishes the other for his non-intentional defecting.

A technique called *symbolic noise detection* (SND) has been shown to be quite effective in coping with noise in the Noisy IPD (Au & Nau 2007; 2006). The basic idea is to use a deterministic model of the other player to identify actions affected by noise during a game. In Category 2 of the 20th Anniversary IPD competition,[1] seven out of nine programs using symbolic noise detection are among the top ten. They lost only to a group of programs that work in a conspiracy to push one program to the top by giving as many points as possible to this program while sacrificing the performance of the rest.[2]

It is natural to ask how helpful SND can be in other kind of games, and whether there are particular kinds of games in which it is especially helpful. Studying SND is important since to the best of our knowledge there is no other general procedure that can handle noise efficiently in such games. This paper addresses these questions by studying SND in the noisy, repeated version of two other 2×2 games: the Game of Chicken (Deutsch 1973; Smith 1982) and the Battle of the Sexes (Luce & Raiffa 1957). The Chicken Game models the situations in which two self-interested parties compete for a resource, but if neither of them concedes, both of them could get none of the resource. A typical example of this type of game is one that two people drive head-to-head towards each other at a very high speed, so that the first driver who swerves is the loser. The Battle of the Sexes models the situations in which two parties need to coordinate with each other to accomplish a task, but they favor different actions. For example, a husband and a wife prefer to go to a football game and an opera, respectively. However, if they go to

[1] The results of the competition can be found on the competition's homepage at http://www.prisoners-dilemma.com.

[2] The other two programs in the top nine used a very different strategy called the *master-and-slaves* strategy. The rules of the competition allowed each participant to submit up to 20 programs, and some participants submitted 20 programs that operated conspirators in which 19 programs (the "slaves") sacrificed their own performance in order to feed as many points as possible to the 20th program (the "master").

different places, both of them would not be happy.

We choose these games because they model many interesting real-life social situations in which dilemma occurs. Solutions to these games have practical implications. Moreover, both the Chicken Game and the Battle of the Sexes relates to IPD by an appropriate scaling of the payoff matrix (Carlsson & Jönsson 2002). Thus, these games facilitate our studies of how well SND works as the parameters of the payoff matrix changes.

Our approach for evaluating SND was to organize tournaments similar to the past IPD tournaments. We organized two tournaments called the *the Noisy Iterated Chicken Game (ICG)* and *the Noisy Iterated Battle of the Sexes (IBS)*, and asked students of an AI class to participate. We also devised a wrapper that can be placed on each of the students' programs. The function of the wrapper is to correct any observed action that is regarded to be affected by noise according to the principle of SND. Then we put the wrapper around students' programs and repeated the tournaments. Our objective was to compare the performance of the strategies (i.e., student's programs) before and after using the wrapper. In particular, we conducted experiments to study the relationships of three variables: (1) the average score of a strategy, (2) the difference of the average scores of a strategy before and after using the wrapper, and (3) the accuracy of correction made by the wrapper.

The contributions of this paper are:

- We provide a general procedure called the *Naïve Symbolic Noise Filter* (NSNF) that can be placed around any existing strategy.
- In our competitions, NSNF was highly accurate in predicting the other player's next moves—96% and 93% of predictions are correct in ICG and IBS, respectively. Not all of these predictions would prompt NSNF to make corrections; NSNF corrects an observed action only when the corresponding prediction is different from the observed action. NSNF also did a decent job in correcting actions affected by noise—out of all corrections made by NSNF, 71% and 59% of them successfully rectified actions that have actually affected by noise in ICG and IBS, respectively.
- In both ICG and IBS, NSNF increased the scores of most programs, and the average increase was higher in ICG than in IBS. One reason for this is that NSNF often has a higher accuracy of noise correction in ICG.
- If we look at each strategy individually, the accuracy in correction does not strongly correlate with the increases in average scores of the strategies due to the use of NSNF. Some strategies, especially those in IBS, actually performed worse with NSNF than without it.
- In both games, strong players and weak players behave quite differently—strong players are more exploitive and they choose defect frequently during a game. However, in ICG this exploitive behavior causes the other player to exhibit a more clear behavior, but in IBS this does not. We explain this phenomenon via certain characteristics of decision making process of the strategies and the structure of the payoff matrix.

Our Hypothesis

Symbolic noise detection is a principle for identifying which the other player's actions has been affected by noise, using a deterministic model of the other player's behavior learnt from the current game history. This idea can be summarized as follows.

- Build a deterministic model of how the other player behaves.
- Watch for any *deviation* from the deterministic behavior predicted by the model.
- If a deviation occurs, check to see if the inconsistency persists in the next few iterations:
    1. If the inconsistency does not persist, assume the deviation is due to noise.
    2. If the inconsistency persists, assume there is a change in the behavior.

The clarity of behavior in IPD has already been discussed by Axelrod in his analysis of TFT (Axelrod 1984). In (Au & Nau 2006), it was argued that clarity of behavior is an important ingredient of long-term cooperation, and therefore most IPD agents exhibit deterministic behavior in tournaments. Thus, SND is effective in IPD because deterministic behavior is abundant in the IPD. However, if we use SND in other kind of games, the intention for cooperation may no longer be abundant; perhaps, cooperation may even be a undesirable behavior in other games. Therefore, an interesting question is to see in what kind of games SND would be effective.

In general, players in any zero-sum game tend to have little clarity in their behavior. For instance, in a game called RoShamBo (Egnor 2000), the objective is to predict the opponent's decision, and therefore players tried hard to conceal their thought patterns. Likewise, it is often hard for chess players to predict the opponent's next move with a high degree of certainty that is large enough for SND to be effective. In some non-zero-sum games, however, deterministic behaviors is more plentiful. It would be beneficial if we have a way to predict the amount of clarity in the player's behavior by just looking at the structure of a game. But this feat is currently out of our reach.

To summarize, our hypothesis is that symbolic noise detection will be the most effective in games in which the strategies are likely to exhibit deterministic behavior.

The Noisy ICG Tournament and The Noisy IBS Tournament

Our tournaments are similar to the Axelrod's IPD tournaments and the 2005 IPD tournament, except that the payoff matrices are the following ones:

Chicken Game's Payoff Matrix		Player 2 Cooperate	Player 2 Defect
Player 1	Cooperate	$(4, 4)$	$(3, 5)$
	Defect	$(5, 3)$	$(0, 0)$

Battle of the Sexes' Payoff Matrix		Husband	
		Football	Opera
Wife	Football	$(1, 2)$	$(0, 0)$
	Opera	$(0, 0)$	$(2, 1)$

In order to allow a program to play the roles of both husband and wife in the IBS, we will use the following modified payoff matrix in the IBS:

Battle of the Sexes' Modified Payoff Matrix		Player 2	
		Defect	Cooperate
Player 1	Cooperate	$(1, 2)$	$(0, 0)$
	Defect	$(0, 0)$	$(2, 1)$

At first glance, the modified Battle of the Sexes' payoff matrix seemed to be different from the usual one. In fact, the difference is just the labels in the matrices; we can obtain the modified payoff matrix by labeling Wife as Player 1, Husband as Player 2, Wife's Football as Cooperate, Wife's Opera as Defect, Husband's Football as Defect, and Husband's Opera as Cooperate. According to the modified payoff matrix, Defect is always more favorable than Cooperate for both Player 1 and Player 2. In order for the players to coordinate with each other, the players has to choose different actions in the modified payoff matrix.

A *game* consists of a finite sequence of *iterations*. In each iteration, two players, namely Player 1 and Player 2, play an ordinary Chicken Game or the Battle of the Sexes. At the beginning of a game, each player knows nothing about the other player, and does not know how many iterations he will play. In each iteration, each player chooses a *move*, which is either cooperate (C) or defect (D). A move is also called an *action*. After both players choose a move, noise may occur and alter the moves—changing 'Cooperate' to 'Defect', or 'Defect' to 'Cooperate'. If noise occurs and changes a move, the other player would see the altered move, not the move originally chosen by the player.

To distinguish the moves chosen by the players from the moves eventually seen by the other players, we call the former the *decisions* and the latter the *physical actions*. Suppose the decisions of Player 1 and Player 2 in an iteration are a and b, respectively. Then the *decision pair* of this iteration is a pair of decisions (a, b), and the *interaction pair* is a pair of physical actions (a', b'), where, (1) $a' = \{C, D\} \setminus a$ if a has been affected by noise, or $a' = a$ if otherwise, and (2) $b' = \{C, D\} \setminus b$ if b has been affected by noise, or $b' = b$ if otherwise.

Noise has the following characteristics.

- The *noise level*, the probability that noise occurs and affects a move, is 10%. Each action has an equal probability of being affected by noise, and the occurrences of noise are independent of each other.
- If Player 1 chooses cooperate but noise changes his action to defect, then (1) Player 2 sees that Player 1's action is defect, and does not know that Player 1 originally chooses cooperate; and (2) Player 1 also does not know that his action has been affected by noise—after Player 1 chooses cooperate there is no way for Player 1 to tell whether his action has been affected by noise. In short, Player 1 knows a and b', and Player 2 knows a' and b; Player 1 does not know a' and b, and Player 2 does not know a and b'.
- The payoff of an iteration is determined by the interaction pair (a', b'), not the decision pair (a, b). But this payoff is not announced to the players, and the players cannot compute the payoff since Player 1 does not know a' and Player 2 does not know b'.

The score of a player in a game is the sum of the payoff he accumulated in all the iterations of the game.

Naïve Symbolic Noise Filter

Our next step is to supplement the collected strategies with symbolic noise detection. Our approach is to develop a procedure called *Symbolic Noise Filter* (SNF) that can be placed around *any* existing strategy to filter the input (the observed action of the other player) to a strategy using SND. The benefit of this approach is that SNF, once implemented, is readily applicable to all strategies. On the other hand, this is helpful to our study because the noise filter of all strategies are the same, and it is easier to compare their performance than the custom-made noise filters.

Our study will use a simplified version of SND called *Naïve Symbolic Noise Detection* (NSND), which is like SND but does not defer judgment about whether a derivation is due to noise or not—it immediately regards a derivation is due to noise when a derivation is detected. This is different from the full-strength SND proposed in (Au & Nau 2006), which utilizes the information *before and after* a derivation to improve the accuracy of noise detection. The benefit of NSND over the full-strength version of SND is that its implementation is much simpler than SND's—there is no need to remember when derivation occurs and adjust the underlying move generator when a change of behavior is detected. Of course, the drawback is that the accuracy of NSND can be lower than SND's. But this deficiency is outweighed by its simplicity, which is important for our wrapper-approach to SND.

SNF based on NSND is called *Naïve Symbolic Noise Filter* (NSNF). Figure 1 illustrates the architecture of NSNF, and Figure 2 outlines the pseudo-code of our implementation of NSNF. Before we discuss the NSNF procedure, let us give the definitions of various terms. A *history* H of length k is a sequence of action pairs of all iterations up to and including iteration k. We write $H = \langle (a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k) \rangle$. A *strategy* is a mapping $M : \mathcal{H} \rightarrow \{C, D\}$, where $\mathcal{H} = \langle (C, C), (C, D), (D, C), (D, D) \rangle^*$ is the set of all possible histories. A *condition* $Cond : \mathcal{H} \rightarrow \{\mathsf{True}, \mathsf{False}\}$ is a mapping from histories to boolean values. For any action a and b, we define $Cond_{a,b}$ to be a condition such that $Cond_{a,b}(H) = \mathsf{True}$ if and only if either (1) $k \geq 1$, $a_k = a$ and $b_k = b$, (where $k = |H|$), or (2) $k = 0$ and $a = b = C$. In other words, $Cond_{a,b}(H)$ is true when the last action pair of H is (a, b). A *deterministic rule* is $Cond \rightarrow b$, where $Cond$ is a condition and b is an action.

The NSNF procedure in Figure 2 has two input parameters: a strategy M and a positive integer promotion_threshold. The strategy M takes a history as its input

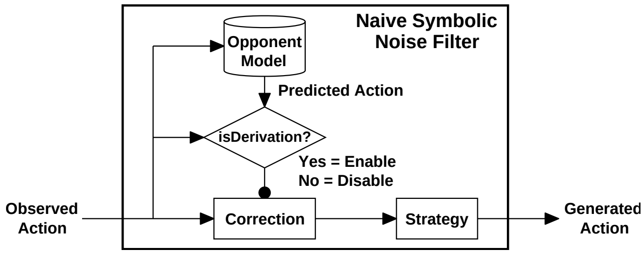

Figure 1: Naïve Symbolic Noise Filter (NSNF).

and generate an action a. The promotion threshold is to control the likelihood of picking up a deterministic behavior. Increasing the promotion threshold reduces the chance that the function **isDerivation** mistakes a random behavior as a deterministic behavior, but ignores certain genuine deterministic behavior. In our tournaments, the promotion threshold is 3.

The procedure has two variables about the current game histories: the recorded history $H_{recorded}$ and the filtered history $H_{filtered}$. The recorded history $H_{recorded}$ is a sequence of action pairs, each of them is the decision a of the strategy M and the other player's physical action b' in an iteration. The filtered history $H_{filtered}$ is like $H_{recorded}$, except that the other player's physical actions in $H_{filtered}$ have been "corrected" if the procedure decides that the physical actions have been affected by noise. The history seem by the strategy M is $H_{filtered}$ rather than $H_{recorded}$ (Line 4 in Figure 2).

This implementation of NSNF is simpler than the implementation of the Derived Belief Strategy (DBS) in (Au & Nau 2006), because it does not explicitly maintain the opponent model (i.e., the hypothesized policy in DBS) but deduces the deterministic rules on demand. The procedure simply searches backward on the current recorded history to locate a deterministic rule $Cond_{a_j,b'_j} \rightarrow b_{next}$, where (a_j, b'_j) is the second to the last action pair of the current recorded history (Line 13 to Line 23). If NSNF finds such a rule, NSNF will make a prediction based on b_{next}. If b'_{j+1}, the last observed physical action of the other player, is the same as b_{next}, no derivation is observed; otherwise, NSNF observes a derivation and regards b'_{j+1} has been affected by noise. Then it corrects b'_{j+1} using the invert function, which returns C if $b'_{j+1} = D$ and returns D if $b'_{j+1} = C$.

As an example, suppose both Player 1 (P_1) and Player 2 (P_2) uses a strategy called Tit-For-Tat (TFT), which starts with Cooperate and then repeats the other player's action in the previous iteration. If both P_1 and P_2 do *not* use the Naïve Symbolic Noise Filter, a possible history can be:

Iteration:	1	2	3	4	5	6	7	8	9	10
Physical Actions of P_1:	C	C	C	D	C	D	C	D	D	D
Physical Actions of P_2:	C	C	C	C	D	C	D	D	D	D

Here, the underlined characters refer to the physical actions that have been affected by noise. We can see that in the fourth iteration, the decision of P_1 was originally C, but was changed to D due to noise. Then this triggered the retaliation of P_2 and started a long sequence of mutual defection between the two players. The situation worsened when noise occurred again in the eighth iteration.

```
NaïveSymbolicNoiseFilter(Strategy, promotion_threshold)
1. H_recorded ← ∅               // the recorded history
2. H_filtered ← ∅               // the filtered history
3. Loop until the end of the game
4.    a ← Strategy(H_filtered)
5.    Output a and get the other player's physical action b'
6.    H_recorded ← ⟨H_recorded, (a, b')⟩
7.    If isDerivation(H_recorded) = True, then
8.       H_filtered ← ⟨H_filtered, (a, invert(b'))⟩
9.    Else
10.      H_filtered ← ⟨H_recorded, (a, b')⟩
Function isDerivation(H_recorded)
11. Let H_recorded be ⟨(a_1, b'_1), (a_2, b'_2), . . . , (a_{k+1}, b'_{k+1})⟩
12. Cond ← Cond_{a_k,b'_k} ; count ← 0
13. For j = k − 1 DownTo 1
14.    If Cond(a_j, b'_j) = True, then
15.       If count = 0, then
16.          b_next ← b'_{j+1} ; count ← count + 1
17.       Else
18.          If b'_{j+1} = b_next, then
19.             count ← count + 1
20.             If count ≥ promotion_threshold, then
21.                If b_next = b'_{k+1}, return False, else return True
22.          Else
23.             Return False
```

Figure 2: The pseudo-code of the Naive Symbolic Noise Filter. The function invert(b') returns C if $b' = D$ and returns D if $b' = C$.

But if Player 2 uses Naïve Symbolic Noise Filter, the above history would become:

Iteration:	1	2	3	4	5	6	7	8	9	10
Physical Actions of P_1:	C	C	C	D	C	C	C	C	D	C
Physical Actions of P_2:	C	C	C	C	C	C	C	D	C	C

At the end of the third iteration, NSNF can readily identify the deterministic rule $Cond_{C,C} \rightarrow C$, because the rule is true in the first three iterations and we set promotion_threshold $= 2$. In the fourth iteration, NSNF saw a derivation from the rule, causing P_2 to correctly consider D in the fourth iteration as being affected by noise. Therefore, P_2 did not retaliate in the fifth iteration. In the ninth iteration, P_1 retaliates for the defection it observes in the eighth iteration since P_1 does not use NSNF. But P_2 did not defect in the tenth iteration, because the deterministic rule $Cond_{C,C} \rightarrow C$ had held repeatedly since the fifth iteration.

In this example, NSNF prevented mutual defection in two different occasions, helping both players to maintain cooperation. Furthermore, NSNF can be effective even if only one player is using it.

Tournament Setup

We have asked students of an advanced-level AI class to participate in two tournaments: the Noisy ICG tournament and the Noisy IBS tournament. There are 37 students in the class, and all of them have submitted programs to both tournaments. We told students that the noise level is 10% and the number of iterations of each game is at least 50. Thus, students do not know the exact number of iterations, which is 300. The ICG tournament is held first. Before the IBS tournament, students were informed about the ranking of their programs in the ICG tournament. This information should not affect the IBS tournaments since the tournaments are different. In each tournament, students were given approximately 2 weeks to complete their programs.

First, we conducted experiments by repeating the ICG tournament (without NSNF) 1000 times as follows. Let $\mathcal{P}_{ICG}$ be the set of all programs for ICG. For any pair $P_i, P_j \in \mathcal{P}_{ICG}$ of programs, P_i has a chance to play with P_j in a tournament. Notice that P_j can be P_i itself. The average score of P_i is the average of the scores of P_i in the 37000 ICG games in which P_i participated. We also collected statistics about P_i such as the number of defection, etc.

Second, for each $P_i \in \mathcal{P}_{ICG}$, we augmented P_i with NSNF and denote the augmented program by $\hat{P}_i$. Then for each $P_j \in \mathcal{P}_{ICG}$, we set $\hat{P}_i$ to be Player 1 and play against P_j for 1000 times. Notice that P_j can be P_i itself (but without NSNF). The average score of $\hat{P}_i$ is the average of the scores of P_i in the 37000 ICG games in which Player 1 is $\hat{P}_i$. We also collected statistics about $\hat{P}_i$ and its NSNF such as the average number of iterations in which NSNF correctly predicted the occurrence of noise in a game, etc.

The IBS tournaments were also conducted in the same way.

Experimental Analysis of NSNF

Our analysis is divided into three sections. The first section presents some basic statistics about the tournaments. Our focus is on three dependent variables: (1) the average scores, (2) the increases in average scores due to NSNF, and (3) the accuracy of correction of NSNF. The second section tries to explain the relationships of these variables by measuring the frequency of change of decisions and the frequency of defects made by strategies. The third section compares the distribution of decision pairs in ICG and IBS.

Basic Statistics

In this paper, average scores are normalized—a normalized average score is equal to the average score divided by the maximum possible scores of a game (1500 in ICG and 600 in IBS). This allows us to put data from different tournaments in the same graph.

Average scores Figure 3 shows the normalized average scores of the strategies, with and without using NSNF, in ICG and IBS. The normalized average scores are ordered according to the ranks of the strategies in the original (without NSNF) tournaments. This shows that the differences of the normalized average scores of the best strategy and the worst strategy are small: 0.144 for ICG and 0.262 for IBS. Therefore, a small change in the average score would have a decisive effect to the rank of a strategy. On average, an increase of 0.0039 and 0.0071 in the normalized average score can change the rank of a program in ICG and IBS, respectively.

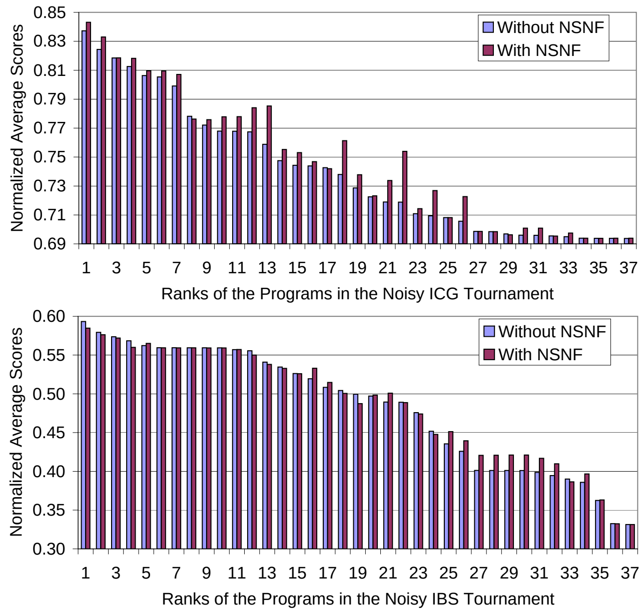

Figure 3: Normalized average scores.

Table 1: The overall normalized average scores (the average of the normalized average scores of all strategies in Figure 3.)

	Without NSNF	With NSNF	Difference
ICG	0.7407	0.7475	0.0068
IBS	0.4834	0.4869	0.0035

Table 1 presents *the overall normalized average scores*, the averages of the normalized scores of all strategies in a tournament. This result shows that NSNF does increase the overall normalized average scores of both ICG and IBS, and the increase for ICG is larger than the increase for IBS.

We computed the *the increase in normalized average scores* of a strategy due to NSNF by subtracting the normalized average score of a strategy without NSNF from the normalized average score of a strategy with NSNF, using the data in Figure 3. Then we present how the increases in normalized average scores relates to the normalized average scores in Figure 4. The relationship for ICG is quite different from the relationship for IBS. In ICG, there is no obvious relationship between the increases in normalized average scores and the normalized average. However, in IBS, the increases in normalized average scores decrease as the normalized average scores increase. In addition, some strategies do not have an increase but a decrease in the normalized average scores. This is essentially true for strategies in IBS, especially those that originally have a high average score.

Accuracy of correction We measured various statistics about the accuracy of NSNF, and the results is summarized

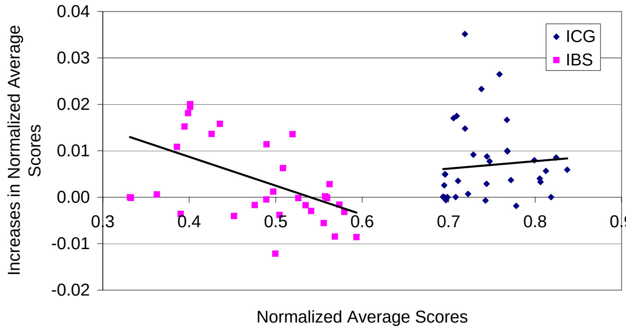

Figure 4: Increases in normalized average scores due to NSNF versus normalized average scores. Notice that 11 points are clustered at (0.7, 0.00).

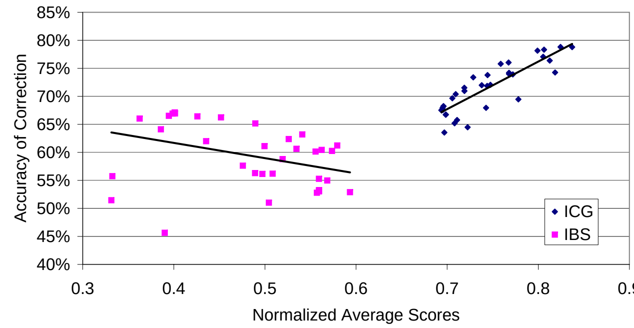

Figure 5: Accuracy of correction versus normalized average scores

Table 2: Accuracy of Predictions and Corrections of NSNF.

	ICG	IBS
Avg. no. of actions affected by noise	30.00	30.01
Avg. no. of predictions made (n_p)	188.86	172.86
Avg. no. of derivations detected	25.43	27.14
Avg. no. of corrections made	25.43	27.14
Avg. no. of false negatives (n_f^-)	0.82	1.23
Avg. no. of false positives (n_f^+)	7.36	11.08
Avg. no. of true negatives (n_t^-)	162.61	144.49
Avg. no. of true positives (n_t^+)	18.06	16.06
Accuracy of predictions	95.67%	92.88%
Accuracy of corrections	71.04%	59.19%
Effective no. of correction	10.70	4.98

in Table 2. In this table, a true positive is referred to a situation in which NSNF correctly predicts that an action is affected by noise. A true negative is a situation in which NSNF correctly predicts that an action is not affected by noise. False positives and false negatives are situations in which NSNF make wrong predictions (noise occurs and noise does not occur, respectively). The accuracy of prediction, which is equal to $(n_t^+ + n_t^-)/n_p$, is a measure of the likelihood that NSNF makes correct prediction. This is different from the accuracy of correction, which is equal to $n_t^+/(n_t^+ + n_f^+)$ and is a measure of the likelihood that NSNF makes correct prediction about the occurrence of noise. The effective number of correction is equal to $n_t^+ - n_f^+$.

Let us discuss the accuracy of NSNF in ICG first. In each iteration, NSNF predicts the next move of the other player by searching backward to see whether a deterministic behavior pertaining to the current condition of the current iteration can be picked up. Thus, a prediction is made only if an appropriate deterministic behavior is found. In ICG, NSNF were able to pick up deterministic behavior and made predictions in 188.86 out of 300 iterations; this is an interesting result, because this shows that there is a significant amount of deterministic behavior in ICG (and in IBS too).

Our statistics showed that most of these 188.86 predictions were correct—the accuracy of prediction is as high as 95.67%. Only 4.33% of these predictions were false positives (predicting there is noise when there is no noise) and false negatives (predicting there is no noise when there is noise). This indicates that the deterministic model of the other player's behavior has a high predictive power indeed.

Nonetheless, the usefulness of these deterministic behaviors depends on whether they can be used to detect noise. Thus, our concern is how often NSNF corrects actions that have actually affected by noise. On average, out of 25.43 corrections made by NSNF, 18.06 of them were correct (i.e., they are true positives). Therefore, the accuracy of corrections is 71.04%. This is a decent accuracy; at least, it shows that NSNF were not making random correction: if we compare our result to the accuracy of a random noise filter, which randomly select 25 actions out of 300 observed actions and correct them and hence only about 2.5 of these corrections are correct (i.e., the accuracy is only 10%), the accuracy of NSNF is much higher.

On the other hand, NSNF made about 28.96% wrong corrections, changing about 7.36 actions when they should have not been changed. The result of these wrong corrections is similar to the introduction of noise into the observed actions (though the introduction is not random). According to this view, the effective number of correction is 10.70.

In IBS, NSNF's prediction is also highly accurate: it is as high as 92.88%. However, the number of true positives among these predictions is slight smaller than that in ICG, resulting in an 59.19% accuracy of correction. The effective number of correction is 4.98 only, which is less than half of the effective number of correction in ICG.

Accuracy of correction vs Average Scores Figure 5 presents how the average scores relate to the accuracy of correction. Surprisingly, a positive linear correlation is observed in ICG, but a negative linear correlation is observed in IBS (after excluding a few outliners due to some weak strategies in IBS). Moreover, most strategies in ICG have a higher accuracy than all of the strategies in IBS.

But how the accuracy of correction relates to the increases in average scores? In Figure 6, we found no obvious relationship between these two variables. Even worse, around half of the strategies in IBS do not have an increase in their average scores no matter what their accuracy of correction

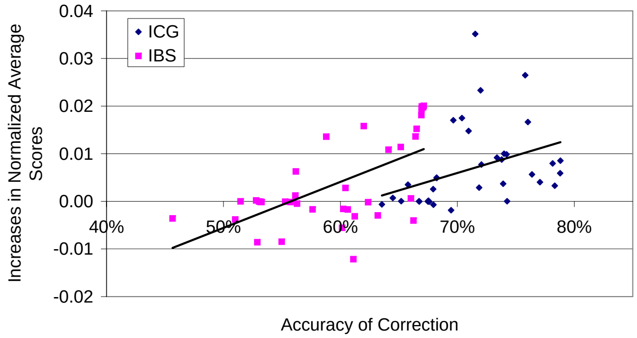

Figure 6: Increases in normalized average scores versus accuracy of correction

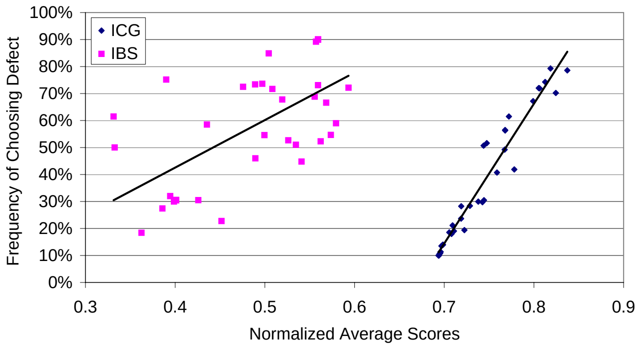

Figure 7: Frequency of choosing defect versus normalized average scores.

are. But one thing we observe in both ICG and IBS is that none of the strategies with a low accuracy of correction (when compared to other strategies in the same tournament) has a higher-than-average increase in their normalized average scores. This is why the slopes of the lines in Figures 6 are positive.

Explanations via Characteristics of Decisions Making Process

This section offers explanations on (1) the relationship between the average scores and the accuracy of correction, and (2) the relationship between the average scores and the increases in average scores. From these, we can deduce the relationship between the accuracy of correction and the increases in average scores.

Our explanations are based on two dependent variables concerning decisions made by players: (1) the frequency of choosing defect (Figure 7), and (2) the frequency of change of the player's own decisions—from cooperate to defect, or vice versa—between two consecutive iterations (Figure 8).

Average scores vs accuracy of correction We observe that the accuracy of correction increases with average scores in ICG in Figure 5. At the same time, we observe that the frequency of choosing defect increases with average scores in Figure 7. We believe this is not a coincidence; in fact, we can see that they are naturally linked together by look-

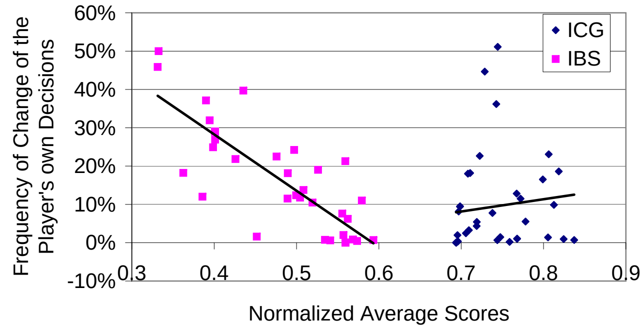

Figure 8: Frequency of changes of the player's own decisions versus normalized average scores.

ing at the structure of the Chicken Game's payoff matrix—if a player chooses defect, the other player is much better off choosing cooperate (and earning 3 points) rather than defect in order to avoid mutual defection that gives 0 point; but if a player chooses cooperate, the choice of the other player would make little difference to the other player's payoff (4 points for choosing cooperate, and 5 points for choosing defect). Thus, if a player manages to choose defect frequently, the other player's behavior would become more deterministic, because he often chooses cooperate. The linear correlation in both Figure 5 and Figure 7 seems to strongly support our explanation.

We also observe a gentle decrease in the accuracy of correction as the average scores increase in IBS in Figure 5. At the same time, we observe that the frequency of choosing defect increases with the average scores in IBS in Figure 7. These figures suggest that strategies that defects frequently have a slight lower accuracy of correction than those that cooperate frequently. Our explanation for this is based on the following observations. First, the cost of mutual defection is small according to the payoff matrix—the player loses 1 point only if he chooses defect instead of cooperate in response to defect. This cost is much lower than the cost in ICG, which is 3 points. Hence, the other players in IBS are more willing to choose defect in response to defect. See the caption of Table 4 for evidence of this point. Second, weak players (who have low average scores) tend to be those who want to play fairly—having the same cumulative payoff as the other player's. By contrast, strong players (who have high average scores) tend to be exploitive—they often defect. Data in Figure 7 and Figure 8 seem to support this description of the characters of strong and weak players. When playing with strong players, weak players would often change their decisions frequently (as they are willing to defect in response to defect) as they struggle for fairness, causing a lower accuracy of the strong players' prediction of their behavior. But from a weak players' viewpoint, the strong player's behavior is very clear.

Average scores vs increases in average scores In Figure 4, we observe that there is no obvious relationship between the average scores and the increases of average scores due to NSNF in ICG, but a negative correlation is observed in IBS. A common feature of the data for ICG and the data for IBS in Figure 4 is that strong players often have a lower increase in average scores than other players. One reason for this is that strong players defect frequently according to Figure 7. More importantly, they defect no matter which move the other player chooses. Thus, strong players are less sensitive to noise because they often choose detect no matter what. Therefore, noise would cause less damage to these players than those that change their decisions frequently. NSNF would be less beneficial to strong players, as strong players do not behave much differently even after the correction of noise. On the other hand, some weak players rely on the establishment of certain interaction patterns with the other player. This process is prone to noise, and thus NSNF can help these players a lot.

Distribution of Decision Pairs

We would like to further investigate the general effect of NSNF in different kind of games by looking at the distribution of different decision pairs as shown in Table 3.

Traditionally, researchers use Nash equilibrium and its extensions to study repeated games; for example, they use Folk Theorems and their variants to answer questions pertaining repeated games (Osborne & Rubinstein 1994). These theorems, however, do not offer any prediction about the distribution of strategies in a tournament. Many interesting phenomena have been observed in simulations, but cannot yet be explained completely by contemporary theories. A famous example is the emergence of cooperation in IPD (Axelrod 1984): about 31% of the interaction pairs in the Category 2 of 2005 IPD competition are (C, C) when there is no master-and-slaves strategy (Table 3). The similar emergence of cooperation was observed in our ICG tournaments—the most-frequent decision pairs turns out to be (C, C), rather than the two pure Nash equilibria of the Chicken Game ((C, D) and (D, C)). In IBS, however, the interaction pairs of strategies are often in two pure Nash equilibria of the Battle of the Sexes ((C, D) and (D, C)). But (D, D) is also a high-frequent decision pairs in IBS. We call (C, C) in ICG and (D, D) in IBS the *emergent decision pairs*, since they have a high frequency of occurrence, but they are not the pure Nash equilibria of the corresponding one-shot games.

NSNF affects the distribution of decisions pairs. In ICG, NSNF leads to a 5.75% increase of (D, C), most of them are converted from (C, C). In IBS, NSNF causes an 3.22% increase of (C, D), most of them are converted from (D, D). A striking similarity of these effects is that there is a decrease in the frequency of the emergent decision pairs—(C, C) in ICG and (D, D) in IBS. In future, we are interested to study whether this also occurs in the Noisy IPD and other kind of games as well.

We also look at the pattern of how often one decision pair changes to another in consecutive iterations. The numbers in Table 4 are computed by dividing the average number of the consecutive iterations that one decision pair change to another by 299, the total number of consecutive iterations in a game. Thus, the sum of all 16 numbers in a matrix is equal to 100%. Notice that the other players in IBS are much more willing to choose defect in response to defects; there is a $12.9\% = (0.39 + 4.26)/(1.52 + 0.39 + 29.77 + 4.26)$ of chance of choosing defect when the previous decision pair is (D, C), as opposed to $2.97\% = (0.18 + 0.70)/(3.47 + 0.18 + 25.48 + 0.70)$ of chance in ICG.

Table 3: Distributions of different decision pairs. The IPD's data are collected from Category 2 of the 2005 IPD competition. *"All but M&S"* means all 105 programs that did not use master-and-slaves strategies, and *"all"* means all 165 programs in the competition. Note that the numbers for IPD are not the number of decision pairs but interaction pairs.

ICG	(C, C)	(C, D)	(D, C)	(D, D)
Without NSNF	38.26%	29.84%	29.83%	2.06%
With NSNF	33.35%	28.87%	35.58%	2.20%
Difference	-4.90%	-0.97%	5.75%	0.14%

IBS	(C, C)	(C, D)	(D, C)	(D, D)
Without NSNF	5.06%	35.91%	35.94%	23.09%
With NSNF	4.94%	39.13%	35.06%	20.87%
Difference	-0.11%	3.22%	-0.88%	-2.22%

IPD	(C, C)	(C, D)	(D, C)	(D, D)
all	13%	16%	16%	55%
all but M&S	31%	19%	19%	31%

Table 5: The sum of the diagonal entries in Table 4.

	Without NSNF	With NSNF	Difference
ICG	82.18%	82.75%	0.57%
IBS	72.36%	74.71%	2.35%

From Table 4, we see that there is a large number of changes to and from the emergent decision pairs in both tournaments. More precisely, in ICG, most changes of decision pairs starts from (C, C) or ends at (C, C), whereas in IBS, most changes of decision pairs starts from (D, D) or ends at (D, D). Perhaps this has something to do with the underlying mechanism that causes the emergence of these decision pairs. However, the flow in and out of these decision pairs drop after we augmented strategies with NSNF.

Table 5 shows how often strategies remain in the same decision pairs in consecutive iterations. First, strategies in ICG more often remain in the same decision pair than strategies in IBS. Perhaps this explains why the accuracy of correction is higher in ICG than in IBS (Figure 5). Second, strategies in both ICG and IBS are more likely to remain in the same decision pairs after using NSNF. This stabilizing effect may have caused the increases in the overall average scores as shown in Table 1.

Related Work

The question of how to deal with noise in multi-agent systems is important, because mistakes in the interaction can cause mistrust and destroy cooperative relationships among agents. Early studies of the effect of noise focused on how Tit-For-Tat (TFT) performs in IPD in the presence of noise. TFT is known to be vulnerable to noise (Axelrod &

Table 4: Frequency of change of decision pairs.

ICG without NSNF

From \ To	(C,C)	(C,D)	(D,C)	(D,D)
(C,C)	**30.89%**	3.56%	3.56%	0.26%
(C,D)	3.47%	**25.48%**	0.18%	0.70%
(D,C)	3.47%	0.18%	**25.48%**	0.70%
(D,D)	0.46%	0.63%	0.63%	**0.33%**

ICG with NSNF

From \ To	(C,C)	(C,D)	(D,C)	(D,D)
(C,C)	**26.45%**	3.22%	3.44%	0.24%
(C,D)	3.13%	**24.87%**	0.19%	0.68%
(D,C)	3.37%	0.18%	**31.12%**	0.91%
(D,D)	0.42%	0.61%	0.87%	**0.31%**

IBS without NSNF

From \ To	(C,C)	(C,D)	(D,C)	(D,D)
(C,C)	**0.56%**	1.17%	1.17%	2.15%
(C,D)	1.52%	**29.75%**	0.39%	4.26%
(D,C)	1.52%	0.39%	**29.77%**	4.26%
(D,D)	1.46%	4.67%	4.67%	**12.28%**

IBS with NSNF

From \ To	(C,C)	(C,D)	(D,C)	(D,D)
(C,C)	**0.60%**	1.46%	1.02%	1.87%
(C,D)	1.64%	**33.63%**	0.39%	3.47%
(D,C)	1.43%	0.41%	**29.14%**	4.07%
(D,D)	1.27%	3.71%	4.57%	**11.33%**

Dion 1988; Bendor 1987; Bendor, Kramer, & Stout 1991; Molander 1985; Mueller 1987; Nowak & Sigmund 1990; Wu & Axelrod 1995). Researchers have proposed better strategies for noisy environments such as Tit-For-Two-Tats, Generous Tit-For-Tat (Nowak & Sigmund 1992), Contrite TFT (Sugden 1986; Wu & Axelrod 1995), and Pavlov (Kraines & Kraines 1989; 1993; 1995; Nowak & Sigmund 1993). But these strategies are not general methods that can be used to deal with noise in a wide variety of situations. The success of the *Derived Belief Strategy* (DBS), based on symbolic noise detection, in the 2005 IPD competition has demonstrated that an opponent model can be used to detect and isolate noise in the Noisy IPD (Au & Nau 2007; 2006). This paper has proposed a general SND-based approach—the Naïve Symbolic Noise Filter—that is applicable to not only DBS but also *any* strategy in 2×2 repeated games.

There is little work on the Noisy ICG and the Noisy IBS, and therefore the questions about how to cope with noise in these games are still open.

Summary

The robustness of interactions is an important issue in multi-agent systems, but mistakes due to noise can interfere with the interactions among agents and decrease the performance of the agents. Symbolic noise detection (SND) is the only known general method for handling noise efficiently in games like the Noisy IPD. The purpose of this paper has been to investigate what kinds of features of a game may cause SND to work well or work poorly.

For our investigation, we have evaluated SND in two other games: the Noisy Iterated Chicken Game (ICG) and the Noisy Iterated Battle of the Sexes (IBS), using a simplified version of SND called Naïve Symbolic Noise Filter (NSNF). We found that NSNF was highly accurate in predicting the other player's next moves (96% and 93% of predictions were correct in ICG and IBS, respectively) and did a decent job in correcting actions affected by noise (71% and 59% of corrections were correct in ICG and IBS, respectively).

Moreover, the overall average scores (the average of the scores of all strategies) increase after using NSNF in both ICG and IBS. Also, the increase in ICG is larger than in IBS. Since most strategies in ICG have a higher accuracy of noise correction than strategies in IBS, this seems to support our hypothesis that symbolic noise detection will be more effective in games in which strategies often exhibit deterministic behavior.

However, if we look at each strategy individually, the results are mixed: the accuracy of noise correction does not strongly correlate with the increases in average scores. Some strategies, especially those in IBS, perform worse than before after using NSNF.

To explain these results, we have explored the relationships between (1) the average scores, (2) the increases of average scores due to NSNF, and (3) the accuracy of noise correction of NSNF. We found that strong players and weak players behave quite differently—strong players are more exploitive, having a clear behavior of choosing defect. While this is true for both games, this exploitive behavior has different effect to the clarity of the other players' behavior in different games. In ICG this causes the other player to exhibit a more clear behavior, but in IBS this does not. We have offered explanations to this phenomenon via the observed characteristics of decision making process of the strategies and the structure of the payoff matrix.

References

Au, T.-C., and Nau, D. 2006. Accident or intention: That is the question (in the noisy iterated prisoner's dilemma). In *AAMAS*, 561–568.

Au, T.-C., and Nau, D. 2007. Is it accidental or intentional? a symbolic approach to the noisy iterated prisoner's dilemma. In *The Iterated Prisoners' Dilemma Competition: Celebrating the 20th Anniversary*. World Scientific. To appear.

Axelrod, R., and Dion, D. 1988. The further evolution of cooperation. *Science* 242(4884):1385–1390.

Axelrod, R. 1984. *The Evolution of Cooperation*. Basic Books.

Bendor, J.; Kramer, R. M.; and Stout, S. 1991. When in doubt... cooperation in a noisy prisoner's dilemma. *The Jour. of Conflict Resolution* 35(4):691–719.

Bendor, J. 1987. In good times and bad: Reciprocity in an uncertain world. *American Journal of Politicial Science* 31(3):531–558.

Carlsson, B., and Jönsson, K. I. 2002. Differences between the iterated prisoner's dilemma and the chicken game under noisy conditions. In *Proc. of the 2002 ACM symposium on Applied computing*, 42–48.

Deutsch, M. 1973. *The Resolution of Conflict: Constructive and Destructive Processes*. Yale University Press.

Egnor, D. 2000. Iocaine powder explained. *ICGA Journal* 23(1):33–35.

Kraines, D., and Kraines, V. 1989. Pavlov and the prisoner's dilemma. *Theory and Decision* 26:47–79.

Kraines, D., and Kraines, V. 1993. Learning to cooperate with pavlov an adaptive strategy for the iterated prisoner's dilemma with noise. *Theory and Decision* 35:107–150.

Kraines, D., and Kraines, V. 1995. Evolution of learning among pavlov strategies in a competitive environment with noise. *The Journal of Conflict Resolution* 39(3):439–466.

Luce, R. D., and Raiffa, H. 1957. *Games and Decisions: Introduction and Critical Survey*. Wiley.

Molander, P. 1985. The optimal level of generosity in a selfish, uncertain environment. *The Journal of Conflict Resolution* 29(4):611–618.

Mueller, U. 1987. Optimal retaliation for optimal cooperation. *The Journal of Conflict Resolution* 31(4):692–724.

Nowak, M., and Sigmund, K. 1990. The evolution of stochastic strategies in the prisoner's dilemma. *Acta Applicandae Mathematicae* 20:247–265.

Nowak, M. A., and Sigmund, K. 1992. Tit for tat in heterogeneous populations. *Nature* 355:250–253.

Nowak, M., and Sigmund, K. 1993. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature* 364:56–58.

Osborne, M. J., and Rubinstein, A. 1994. *A Course in Game Theory*. The MIT Press.

Smith, J. M. 1982. *Evolution and the Theory of Games*. Cambridge University Press.

Sugden, R. 1986. *The economics of rights, co-operation and welfare*. Blackwell.

Wu, J., and Axelrod, R. 1995. How to cope with noise in the iterated prisoner's dilemma. *Journal of Conflict Resolution* 39:183–189.