Stochastic Resonance in Human Cognition: ACT-R Versus Game Theory, Associative Neural Networks, Recursive Neural Networks, Q-Learning, and Humans

Robert L. West (robert_west@carleton.ca)[1], **Terrence C. Stewart,** (tcstewar@connect.carleton.ca)[1], **Christian Lebiere** (clebiere@maad.com)[2], **Sanjay Chandrasekharan** (schandr2@connect.carleton.ca)[1]

[1]Institute of Cognitive Science, Carleton University, Ottawa, Ontario, Canada, K1S 5B6
[2]Micro Analysis and Design, Inc., 6800 Thomas Blvd, Pittsburgh, PA 15208 USA

Abstract

We examined the effect of cognitive noise on human game playing abilities. Human subjects played a guessing game against an ACT-R model set at different noise levels. Counter to the normal effect for noise (i.e., to increase randomness) increasing noise over certain ranges increased the win rate in both the ACT-R model and in the humans. We then attempted to model the human results using ACT-R, Q-Learning, neural networks, and Simple Recursive Neural Networks. Overall, ACT-R produced the best match to the data. However, none of these models were able to reproduce a secondary counter intuitive human win rate effect.

Noise, or randomness, plays an important role in cognitive modelling. In problem solving it is often necessary to add noise to a model to get it to explore possible solutions rather than freezing onto a single approach. In memory models, noise often plays a role in modelling errors of omission and commission (e.g. Anderson & Lebiere, 1998). Noise is also used to model the ability of humans to purposefully behave stochastically (e.g., Treisman & Faulkner, 1987). In these cases, the role of noise is to create and/or increase randomness in behaviour. However, adding noise to a component within a system can also have the opposite effect. That is, adding noise can, under the right conditions, *decrease* randomness (i.e. the system's behaviour moves away from chance).

The best-known example of this is stochastic resonance (SR). SR refers to a class of models that produces the effect of reducing randomness by adding noise. Importantly, SR has been implicated in neural functioning (see chapter 22 of Ward, 2002 for a review) and has also been shown to influence decision making in perceptually based tasks (see chapter 21 of Ward, 2002, for a review). However, there is no agreed upon, precise definition of when a system should be classed as SR. For experimental results it is often the case that a result is assumed to be SR if adding noise to a system reduced the level of randomness of the system in some way. This is the sense in which we use the term SR. However, the important point is not the technical definition but whether or not noise can function in this way for the cognitive system, as it is known to do for the neural and perceptual systems.

Games, Randomness, and Cognitive Noise

In game theory, the ability to behave randomly or pseudo-randomly often plays a central role. This is because increasing the level of randomness in a player's moves decreases the ability of the opponent to predict these moves. If we assume that increasing noise in a cognitive model will always increase the level of randomness in its behaviour then there is a direct link between cognitive noise levels and the level of randomness in a game. However, if adding noise can, under certain conditions, reduce the level of randomness, then the relationship between cognitive noise and randomness is not so straightforward.

We investigated this by looking at the relationship between cognitive noise and the ability to predict your opponent in the game of Paper, Rock, Scissors (henceforth PRS). PRS was chosen for this study because the game theory solution is very simple; just play randomly, 1/3 paper, 1/3 rock, 1/3 scissors. The reason for this is that any deviation from this strategy would leave the player open to exploitation from an opponent who could detect the deviation. The expected outcome for this strategy over time is for both players to play at chance; 1/3 wins, 1/3 losses, and 1/3 ties. If adding noise to the cognitive system of a player increases the randomness of their playing then adding noise should cause the rate of win, losses and ties to move towards the chance rate. In contrast, an SR effect would cause one or both players to move away from the chance rate as more noise is added. Typically, such an effect would occur over only a limited range of the noise parameter.

Another reason that PRS is a good choice is that the cognitive processes underlying PRS play have been previously studied. Human PRS play has been successfully modelled using neural networks (West & Lebiere, 2001) and ACT-R (Lebiere & West, 1999). In both cases the basic strategy was the same: to attempt to win through the detection of sequential dependencies. Specifically, each player tries to predict what their opponent will play next by detecting sequential dependencies in past moves. Both the neural network model and the ACT-R model, when compared to human data, indicated that people use their opponent's last two moves to predict the current move. We refer to this as a lag 2 model. Simpler models, which use only the last move, were termed lag 1 models.

The effect of cognitive noise on this strategy seems straightforward: as noise is added to the sequential dependency mechanism the player should become less able to predict their opponent's moves. Also, as their moves are increasingly determined by the noise they should become increasingly hard to predict. Eventually the cognitive system will become completely swamped with noise and all the moves will be random. That is, the win/loss/tie rates for both players will converge towards the chance rates. With sufficient noise this outcome is unavoidable. However, if an

SR effect exists then the relationship between the level of noise and the level of randomness will not be monotonic. That is, for some regions of noise, increasing noise will cause the win/loss/tie rates to move away from chance.

Humans Versus ACT-R

Testing for stochastic resonance in humans is difficult because it is problematic to add precise amounts of noise to their mental processes. We approached this issue by having human subjects play against an ACT-R model with different amounts of noise. Previous experiments (Lebiere & West, 1999; West & Lebiere, 2001) have shown that humans tend to beat the ACT-R model when it is set to detect sequential dependencies at lag 1, while they tend to tie or lose if the model uses the last two moves (lag 2) to predict the next move. They also found that human players were much more motivated when they were winning, so we chose the lag 1 version for this experiment.

The ACT-R model learns sequential dependencies by observing the relationship between what happened and what came before on each trial. After each round, a record of this is stored in the ACT-R declarative memory system. Each time the same sequence of events is observed it strengthens the activation of that sequence in memory. Thus, activation levels reflect the past likelihood of a sequence occurring. Noise is added (via the standard ACT-R activation noise parameter) when the model attempts to retrieve the sequence with the highest activation level in order to predict the opponent's next move. For example, if the opponent's last move was P and the model was set to use information from just the previous move (lag 1), the model would choose from PR, PS, and PP based on activation levels, then use the retrieved sequence to predict the opponent's next move. Thus, if PS had the highest activation this would predict that the opponent will play S next, and so the model would play R (which beats S). The model would then see what the opponent actually did and store a record of it (e.g., assume the opponent played S, the model would then store PS), which would strengthen the activation of that sequence.

Previous research has shown that when two players use the strategy of detecting sequential dependencies against each other, the result is that both players produce a series of short-lived sequential dependencies (West & Lebiere, 2001). Adding noise to the ACT-R model increases the likelihood that a mistake will be made, in that it increases the chance that the most active sequence may not be chosen. Thus, if the model 'knows' the right answer, adding noise increases the chance that it will fail to retrieve it. When noise causes a failure to retrieve the most active sequence, it also introduces noise (i.e., false information) into the signal sent to the opponent. This causes the opponent to store false information that is not predictive of the player's sequential dependencies (which are embodied in the activation levels), and so introduces noise into the opponent's learning process. Thus we tested for two sources of noise: internally generated noise in the ACT-R model, and noise in the signal provided to the human players.

For all the experiments, each human subject played against the ACT-R model at different noise levels. Each game was 150 trials and the order was randomized. To test the significance of the noise manipulation we used a least squared regression to examine the human scores across trials and the ACT-R score across trials. The slope of the regression produced an estimate of the win and loss rates for each noise level.

The thick lines in Figure 1 and Figure 2 show the results for our first experiment. In Figure 1, we see that as the model noise increased from 0 to 0.5, the human win rate decreased towards chance. At the same time, Figure 2 shows that the model's win rate (which is the same as the human's loss rate) increased dramatically. Importantly, between a noise level of 0.2 and 0.5, increasing the amount of noise caused the win rate to significantly increase in a direction away from chance ($p<0.001$). This is a clear example of an SR effect within the model. The model was able to predict the human players better as noise was added, within this noise range.

The data also revealed an SR effect within the human players. When the model's noise was increased from 0.5 to 0.75, the humans significantly increased their win rate away from chance ($p<0.001$). The increased noise in the ACT-R model caused increased noise in the information being received by the human players, which in turn caused an *increase* in their ability to predict the model's performance.

To replicate the human SR effect, we repeated the experiment focusing in on the noise range that produced this effect. As the thin line in Figure 1 illustrates, the effect for the humans was even stronger in this experiment, possibly because we came closer to capturing the peak of it. Again, the movements away from chance were significant ($p<0.001$).

Finally, to be sure of the effect, we re-ran the experiment using only experienced players ($n = 8$) who had been able to win in the previous experiment, and focused in on two noise levels that would maximize the effect. Again we found a significant increase in the human hit rate ($p<0.001$). In this case, probably due to the use of experienced players, every subject individually produced the effect. As far as we know, this is the first direct demonstration of SR effects at the cognitive level in human subjects.

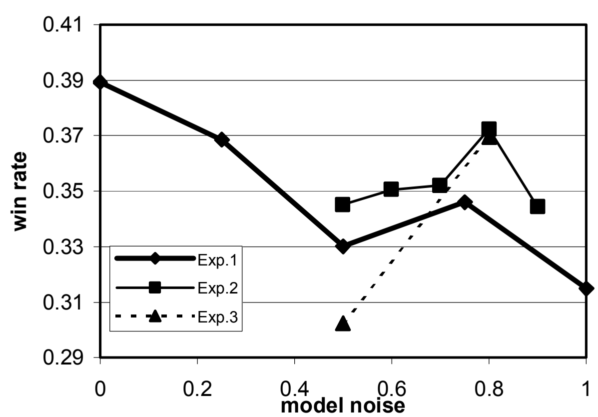

Figure 1: Human win rate (model loss rate) at different levels of model noise for three different experiments.

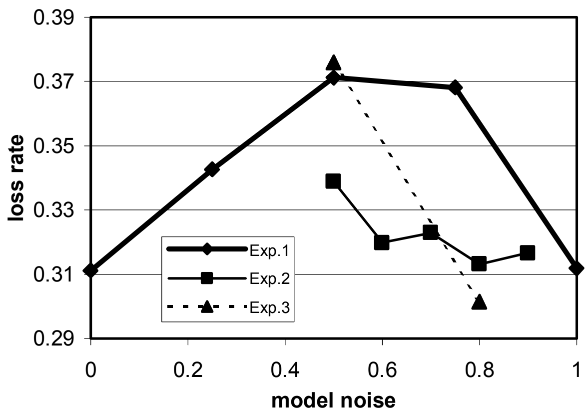

Figure 2: Human loss rate (model win rate) at different levels of model noise for three different experiments.

Modeling Human Performance

We developed a number of fundamentally different models of the human performance versus ACT-R, and tested them at various parameter settings. In practice, this approach can have a number of different outcomes. First, a model may simply not match well with the human data, no matter what changes are made to its parameter settings. This falsifies the model. Second, a model may match well over a wide range of plausible parameters settings (or, indeed, over all parameter settings). It is our experience that this happens surprisingly often (see Stewart, West, & Coplan, 2004 for an example). Third, a model may match the human data well, but only over a particular narrow range of settings. If there is no way to explain or justify those parameter settings then there is a possibility that the fit is due to capitalizing on chance and it is difficult to draw any conclusions about the validity of the model. Fourth, a model may inherit recommended settings for its parameters that have been found to work in a wide range of situations. In this case, there is a prediction that the standard settings should work well in the new situation, and as the parameters are moved away from that norm, the accuracy should decrease.

We compared the human data shown in the previous section to models falling into five major classes: Game Theory, ACT-R, Associative Neural Net, Simple Recursive Neural Net, and Q-Learning. To do the comparisons, we ran the various models against the same opponent that the humans played: an ACT-R lag 1 model with varying degrees of noise. For each level of noise in the opponent, we ran 100 simulations of the two agents playing 150 rounds of PRS.

For each model class, we created a large number of sub-models by adjusting the internal parameters. A variety of settings for each parameter were chosen, and the models were run for each combination of settings. All models were implemented in Python, and the source code is available at <http://ccmlab.ca/prs>.

The Game Theory Model

This is the simplest model, and inspired by the pure game-theory solution to PRS. This model chooses its actions randomly, without regard to the actions of the opponent. This is not expected to be a good match to the human results, but is included so as to have a baseline for comparison. It has no parameters.

The ACT-R Model

This model is as previously described. It has a single parameter (the level of noise), and the general recommendation is to set this value to 0.25. This gives us a prediction that the model should be optimal at or near that setting.

We also examined a number of variations on the basic ACT-R model. The original ACT-R PRS model was created in ACT-R 4. An important aspect of this model was that it used the architecture in a very direct way to detect sequential dependencies. ACT-R 5 introduced a change in the architecture such that implementing the version 4 model in ACT-R 5 could not be achieved in a simple and direct way. So we created a version 5 model that used the ACT-R 5 architecture in the most direct way. The difference amounts to this: in version 4 only the chunk describing what actually happened is strengthened, while in version 5 the chunk describing what the model thought was going to happen is also strengthened. This makes sense, as both of these chunks play an important role and are focused on.

In both ACT-R 4 and 5, there is the option to enable 'partial matching', allowing for memory retrieval errors. We varied this and found that it caused either no significant effect or a deleterious effect. These results are not otherwise reported.

We also tried making use of ACT-R's 'optimized learning' system. This approximation of the learning system is used in ACT-R models to save computing time, but, similar to Sims and Gray (2004), we found that it significantly altered the results. Because there is no theoretical story behind the optimized version we only report on the results from the full, non-optimized version.

Associative Network

This model has been used previously to model PRS playing (West & Lebiere, 2001). Here, a network is used whose weights form a payoff matrix for performing a given action given the previous moves by the opponent. The weights are then modified based on whether or not this choice results in a win. The rewards and punishments were set equal to the game payoffs (i.e., +1 for winning and -1 for losing) so the only free parameters are the number of rounds of history to use (i.e. the 'lag' of the network, in the same sense as the ACT-R model), and whether the system treats ties as neutral (payoff = 0) or as losses (payoff = -1). West and Lebiere (2001), using different experimental manipulations, found that a lag 2 network that treated ties as losses most closely modeled the human data.

Simple Recursive Neural Network

An SRNN is a variant of the standard neural network that is specifically designed to predict the next element in a sequence (Elman, 1990). It does this by adjusting its connection weights via the back-propagation of error learning algorithm (Rumelhart et al, 1986), and by having a separate set of inputs that are set to the values of its own internal hidden nodes. This allows the network to learn its own representation of past events, and thus to find patterns that are not artificially limited to being of a certain length. This is in contrast to the models seen thus far which are set to be either lag 1 or lag 2.

The important parameters for an SRNN are the learning rate and number of hidden nodes. We also varied the number of times the network was trained on the previously seen data. This allowed the network to adjust more quickly to short-term patterns. Payoffs were set in the same way as the associative network.

Q-Learning

Here, we made use of the classic reinforcement-learning algorithm as defined in (Watkins, 1989). This is an action-selection algorithm that makes decisions based on a current sensory state (in this case, the last 1 or 2 moves by the opponent) and an experientially learned estimation of the long-term rewards (as measured by wins and losses) for overall strategies. Importantly, it is capable of learning strategies that involve short-term loss for long-term wins.

However, it has three parameters (the learning rate, the future-discount rate, and the exploration rate), and these do not have suggested values. Furthermore, we had the option of treating ties in the game as either losses or neutral (as in the SRNN and associative networks), and we could set it to be either lag 1 or lag 2. This resulted in a large number of possible parameter settings.

Modeling Results

To establish that a given model matches with the known human data, we performed equivalence testing. This is similar to a standard t-test, but the null hypothesis is that the two means are different, rather than being equal. In the results that follow, the p-values indicate the probability that the model data and the human data have means that differ by more than 2.5%.

For each model, we measured the average win and loss rates when that model played against the ACT-R lag 1 model with the 9 different levels of noise studied in the human data. This gave us 18 p-values per model. We combined the p-values using Fisher's rule, resulting in a single p-value indicating the probability that the model and the humans had different mean scores. This means that a p-value of less than 0.05 indicates 95% certainty that the model is within 2.5% of the human data.

In total, we investigated 223 separate parameter settings.

The Game Theory Model

Since it is well known that humans are generally bad at performing randomly (e.g. Neuringer, 1986), it was expected that this model would act as a benchmark for comparison. As expected, the match was not significant ($p>0.05$).

Q-Learning

No matter what combinations of parameter settings were tried, we were unable to find a Q-Learning model that matched the human data, according to our criteria. All p-values were above 0.05. However, the best version of the model came close to significance. It was a Lag-2, treating ties like losses, with learning rate of 0.5, exploration of 0.1, and future discount rate of 0.95, which achieved a p-value of 0.052. Most settings were significantly worse.

Associative Network

This model was also unable to match the human data at the 0.05 significance level. Its best result was also with a Lag of 2, and treating ties as losses (the same result as found in West & Lebiere, 2001).

Simple Recursive Neural Network

For the SRNNs, we found one model that matched the human data, according to our criteria. With 3 hidden nodes, a learning rate of 0.1, and repeating the training 100 times, the model achieved a p-value of 0.02. This gives us a 98% certainty that the model plays within 2.5% of the human performance. However, since there were 50 SRNN models investigated, the fact that one was found to match with 98% confidence would be expected, even if none of the models matched. This means that we should be wary of accepting the SRNN as a model of human performance on this task.

ACT-R

Almost half of the ACT-R models investigated were found to be good matches ($p < 0.05$) to the human data. This is a remarkable result, indicating that we can model the human data accurately *without parameter tweaking*. However, the best matches were in the range of 0.25-0.28. This compares favorably to the recommended noise setting of 0.25.

Table 1: The Top 10 ACT-R Models

p-value	Noise	Lag	Version
<0.01	0.28	2	5
<0.01	0.25	2	5
<0.05	0.28	2	4
<0.05	0.28	1	4
<0.05	0.5	2	4
<0.05	0.3	2	5
<0.05	0.3	2	4
<0.05	0.25	1	4
<0.05	0.5	1	5
<0.05	0.7	1	4

The SR Effects

All of these results were based on an overall fitting of the model data to the human data. As discussed earlier, in the human versus ACT-R data there were two SR effects. As noise was added, the first effect was a relatively large effect benefiting the ACT-R player. The second effect was a relatively small effect benefiting the human player. Disappointingly, none of the models we tested produced the second effect.

To determine if this effect could be produced with highly specific parameter settings, we ran a (1,5)-Evolutionary Strategy (a relative of a standard Genetic Algorithm, but more suited for parameter optimization) on all the models to try to get this effect. However, this failed to get the effect in any of the models, indicating that none of the models is ultimately correct. All of the models became swamped by the noise in the output of the lag 1 ACT-R model and went to chance rates around the point where the human SR effect occurred.

Discussion

In this study we tested four different classes of cognitive models. Qualitatively, all of them (with the right parameter settings) were able to match the human win rate for the first SR effect to some degree. That is, it was possible to produce a model that initially could beat the lag 1 ACT-R model, but lost this ability as noise was added. One way to interpret this is that the model is able to predict the opponent (the ACT-R lag 1) and so wins above chance; however, the opponent cannot predict the model's moves and so wins at chance. Because one wins above chance and the other at chance, ties occur at a level below chance. As noise is added to the opponent it becomes more difficult for the model to predict it and its win rate and the tie rate converge towards chance. In this situation, everything moves towards chance as noise is added. This is a standard randomness effect as opposed to an SR effect. However, this is not what happened for the ACT-R lag 1 model that the human subjects played against. The human win rate went down, but the model's win rate (the human loss rate) increased away from chance, producing an SR effect (see Figure 2).

The mechanism for this first SR effect may reside in the fact that the interaction between two players using the sequential dependencies strategy causes the players to generate short lived sequential dependencies (West & Lebiere, 2001). If the opponent can detect one of these, it can be exploited, but after it disappears the opponent needs to let it go and find the next one. Thus unlearning is as important as learning. Under these conditions, during the unlearning stage it can actually be an advantage not to select the most active chunk, because it now represents a wrong prediction. Of course it is also a disadvantage not to select the most active chunk once a sequential dependency has been learned and is still valid. However, if learning is more transitory than unlearning, the overall effect would be to increase the win rate.

Another possibility is that the dynamic interaction that creates the sequential dependencies is affected. It is important to realize that the sequential dependencies are not generated by the individual players but through the interaction between them. Thus it is possible for changes in the behaviour of one player to affect the sequential dependencies outputted by the other player. In most cases adding noise would increase the chance of choosing the next most active chunk. It is theoretically possible that this could affect the interaction such that the opponent would output stronger sequential dependencies and thus be easier to predict. The dynamic interaction between the players forms a complex coupled system that is not easily unpacked.

With this in mind, Figure 3 shows the results for the best performing ACT-R model (version 5, lag 2, noise = .28). It models the human win rate well, but the loss rate is much flatter than the human loss rate. However, the small initial rise away from chance in the loss rate is significant ($p<0.001$), so the model did succeed in replicating the effect, although it produced only a muted version of it. The problem seems to be that the randomness effect was too large and the SR effect was too small. Interestingly the best performing models for the Associative Networks and Q-Learning (not shown) also produced a muted version of the first SR effect.

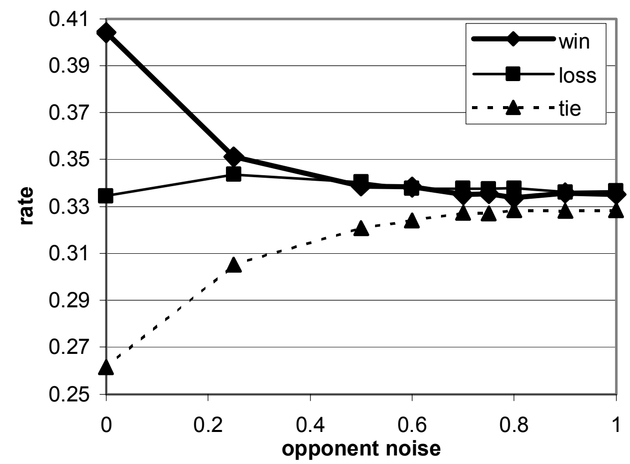

Figure 3: Performance of the best ACT-R model (lag 2, noise 0.28, version 5) playing against the same opponent the human subjects played against.

Overall, when the models were examined across the entire data set, only ACT-R and SRNN were able to outperform the random game theory model in terms of matching the human data. However, SRNN could do this only for a very limited set of parameter settings that did not have a theoretical justification, and also failed to reproduce the first SR effect. Thus it is questionable whether the SRNN model should be regarded as better than the Associative Network and Q-Learning models. In contrast, ACT-R was able to outperform the random model over a wide range of parameter settings, worked best for parameter settings at or near the value found to work in most ACT-R models, and produced the first SR effect.

The fact that the ACT-R model performed well in this study, and also in other studies using different games (Lebiere, Wallach, & West 2000; Lebiere, Gray, Salvucci, & West, 2003), indicates that it is accurately capturing a significant portion of the cognitive functions involved in human game playing. However, the ACT-R model we used was falsified, along with all the other models, because it could not produce the second SR effect. This is part of the normal process of developing and refining models. Some (e.g., Roberts & Pashler, 2000) have suggested that cognitive models are not falsifiable and therefore not scientific. Our results show that this is not the case, as the current version of the ACT-R model has clearly been falsified. However, we hope that with further study we will be able to develop an ACT-R model that will produce both SR effects and thus shed more light on the phenomenon.

The second SR effect requires an explanation. It was interesting that the second effect occurred as the first effect was ending. We speculate that this marked a transition from the first SR effect to a normal randomness effect in the ACT-R model. One possibility is that at this point the increasing randomness component in the signal from the ACT-R model caused the human subjects to be less 'locked in' when they detected a sequential dependency. We have some simulation results indicating that this can produce an advantage. Another possibility, as with the first SR effect, is that the dynamic interaction that created the sequential dependencies was affected, causing the ACT-R model to output stronger sequential dependencies. Further analysis (beyond the scope of this paper) is required to understand the mechanisms for both the first and second SR effects.

Another interesting question is the extent to which SR effects can occur in ACT-R models. The ACT-R lag 1 model produced a large SR effect when interacting with the human players, but only a muted effect when playing against the ACT-R lag 2. This raises the question of whether a large effect is possible using ACT-R to model both players. Figure 4 shows a lag 2 ACT-R model set at different noise levels playing against a lag 1 ACT-R model fixed at a somewhat low noise level (noise = 0.10). This result demonstrates that large SR effects are possible using just ACT-R players (this is also demonstrated in Lebiere and West, 1999).

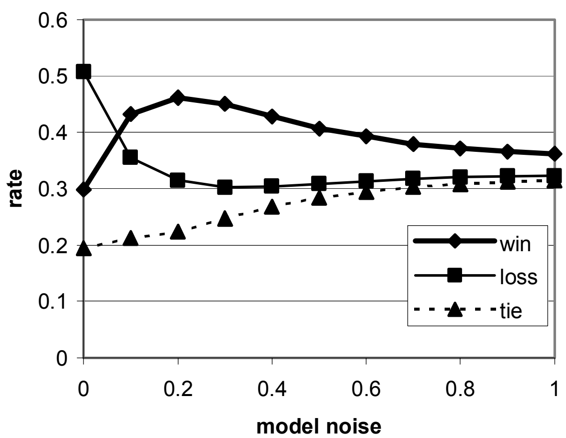

Figure 4: Performance of an ACT-R lag 2 model against the same opponent the humans played against. Unlike the other graphs, here we vary the amount of noise in the lag 2 model, not in the opponent.

Conclusions

As far as we can ascertain, this is the first conclusive demonstration that adding noise can produce an SR-like effect in the human cognitive system. Although we could not model both SR effects found in the human data, three out of the five cognitive models we tested did produce the first SR effect. This indicates that SR effects are a property of current models of cognition. This means that adding noise to a cognitive system should not automatically be assumed to increase the randomness of that system's behaviour. This is particularly true for systems involved in dynamic interactions with competition and feedback.

References

Anderson, J. R. & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

J.L. Elman, (1990). Finding structure in time, *Cognitive Science*, 14, 179-211.

Kitajo, K., Nozaki, D., Ward, L., & Yamamoto, Y. (2003). Behavioral Stochastic Resonance within the Human Brain. *Physical Review Letters*, 90:21.

Lebiere, C., Gray, R., Salvucci, D., & West, R. L. (2003). Choice and Learning under Uncertainty: A Case Study in Baseball Batting. *Proceedings of the 25th Annual Meeting of the Cognitive Science Society*, 704-709.

Lebiere, C., Wallach, D., & West, R. L. (2000). A memory-based account of the prisoner's dilemma and other 2x2 games. P*roceedings of International Conference on Cognitive Modeling*, 185-193. NL: Universal Press.

Lebiere, C., & West, R. L. (1999). Using ACT-R to model the dynamic properties of simple games. *Proceedings of the Meeting of the Cognitive Science Society*. Hillsdale, NJ: Earlbaum.

Neuinger, A. (1986). Can people behave "randomly"? The role of feedback. *Journal of Experimental Psychology: General*, 115, 62-75.

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review*, 107, 358-367.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986) Learning representations by back-propagating errors. *Nature*, 323, 533-536.

Sims, C. & Gray, W. D. (2004). Episodic versus semantic memory: An exploration of models of memory decay in the serial attention paradigm. *Proceedings of the 6th international conference on cognitive modeling.* 279-284. Mahwaj, NJ: Earlbaum.

Stewart, T.C., West, R., and Coplan, R. (2004). A Dynamic, Multi-Agent Model of Peer Group Formation. *Proceedings of the 6th international conference on cognitive modeling* 290-295. Mahwaj, NJ: Earlbaum.

Treisman, M. & Faulkner, A. (1987). Generation of random sequences by human subjects: Cognitive operations or psychophysical process? *Journal of Experimental Psychology: General*, 116, 337-355.

Ward, L. (2002). *Dynamical Cognitive Science*. MIT Press.

Watkins, C. J. C. H. (1989). Learning from Delayed Rewards. Doctoral dissertation, Cambridge University, Cambridge, England.

West, R. L., & Lebiere, C. (2001). Simple games as dynamic, coupled systems: Randomness and other emergent properties. *Cognitive Systems Research*, 1:4, 221-239.