Non-stationary Policy Learning in 2-player Zero Sum Games

Content Areas: machine learning, Markov decision processes, reinforcement learning

Abstract

A key challenge in multiagent environments is the construction of agents that are able to learn while acting in the presence of other agents that are simultaneously learning and adapting. These domains require on-line learning methods without the benefit of repeated training examples, as well as the ability to adapt to the evolving behavior of other agents in the environment. The difficulty is further exacerbated when the agents are in an adversarial relationship, demanding that a robust (i.e. winning) non-stationary policy be rapidly learned and adapted.

We propose an on-line sequence learning algorithm, ELPH, based on a entropy pruning technique that is able to rapidly learn and adapt non-stationary policies. We demonstrate the performance of this method in a non-stationary learning environment of adversarial zero-sum matrix games.

Introduction

A significant challenge in multiagent environments is to learn and adapt in the presence of other agents that are simultaneously learning and adapting. This problem is even more acute when the agents are competing in some task. In competitive environments, each agent is trying to optimize its return at the expense of the other agents, therefore any single agent's success depends on the actions of the other agents. Optimal behavior in this context is defined relative to the actions of the other agents in the environment on a moment to moment basis. An agent's policy must continuously change as the other agents learn and adapt. Assuming the other agents have similar goals (i.e. to win), this results in the need to learn non-stationary policies over the space of stochastic actions.

Many previous machine learning approaches apply to single agent domains in which the environments may be stochastic, but the learned policies are stationary. (Sutton & Barto 1998; Agrawal & Srikant 1995) However, approaches that learn only stationary policies are insufficient for multiagent, non-stationary environments. We present ELPH, a novel on-line learning method that learns quickly and is highly adaptive to non-stationary environments. We demonstrate these abilities in the non-stationary policy learning context of the two-player zero sum game Rock-Paper-Scissors, employing competition against both synthetically generated agents and human opponents.

Zero sum matrix games

Matrix games (Owen 1995) are two player games in which each player selects simultaneously from some set of actions, $a_i \in A$. Each player's payout or reward can be represented by an $n \times n$ matrix, R_{ij} in which the rows i represent the player's action and the columns j represent the opponent's action. A *zero sum* matrix game is one in which each player's reward matrix is the negative of the other ($R_1 = -R_2$). In the game of Rock-Paper-Scissors each player selects from $a_i \in \{rock, paper, scissors\}$ with reward as follows:

$$R_1 = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}, R_2 = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

Here, paper beats rock, scissors beats paper, and paper beats rock. Nobody receives a payout for ties (Fudenberg & Levine 1999).

In each of these cases, there is no optimal policy for either player that is independent of the other. For example, if $player_1$ employs a policy of playing all *rock*, then the optimal policy for $player_2$ is to play all *paper*. Assuming each player is playing rationally and adapting his strategy, a game-theoretic result for 2-player zero sum games is that each player converges to a unique Nash equilibrium. In this case, the equilibrium policy is to play randomly.

Bowling (Bowling & Veloso 2002) has shown that the WoLF (Win or Learn Fast) principle applied to an incremental gradient ascent over the space of possible policies will converge to the optimal policy. However, using incremental gradient ascent is problematic when faced with an on-line adversarial learning environment in which the policy space gradient is non-stationary and the current operating policy must be adapted quickly (within a few plays).

The ability for gradient ascent to "learn fast" depends entirely on the selection of the learning rate applied to the policy update. If the learning rate is small, adapting to the opponent's strategy will be slow. If the learning rate value is large, facilitating more rapid learning, convergence might be compromised. Although incremental gradient ascent can be proven to converge to an optimal policy in the long run, the ability for an opponent to change policies more quickly can render this method ineffective.

ELPH: Sequence learning with an Entropy Learning Pruned Hypothesis space

In situations where an opponent agent is likely to change policies frequently and without warning, it is essential that an agent (1) learn on-line, (2) learn as rapidly as possible and (3) adapt quickly to changing opponent strategies.

We propose a novel on-line learning algorithm, that observes and learns temporal sequences over a short-term observation history using an entropy measure to discard all but highly predictive sequences. This method is called ELPH (Entropy Learning Pruned Hypothesis space). The method exhibits an ability to both rapidly learn predictive sequences (using as little as a single example) and quickly adapt to non-stationarity in the underlying process statistics. In a very general sense, the strategy is to intentionally overfit the observations and subsequently discard non-predictive and/or inconsistent hypotheses in real-time.

Unlike order-n Markov chain methods, in which learning occurs over a space of uniform n-grams, ELPH learns over a space of *hypotheses* (HSpace).

Given a short temporal history of the n most recent observations, an individual *hypothesis* consists of a unique subset of the ordered contents of the observation history together with the set of events that have, in the past, immediately followed the pattern contained in the observation subset.

Consider some event e_t, occurring at time t, which is immediately preceded by a finite series of temporally ordered observations $(o_{t-n}, \ldots, o_{t-1})$. If some subset of those observations consistently precedes the event e_t, then it can be subsequently used to predict future occurrences of e_t. In general, if the observed system takes the form of a Markov chain of order 1, then the single observation o_{t-1} can be used to predict the probability of the event e_t. However, given an arbitrary series of observations, it is not necessarily true that the sequence results from a Markov process of order 1. For example, it may be that a single observation like o_{t-4} or a combination of two specific observations like $\{o_{t-6}, o_{t-4}\}$ might suffice to accurately predict the observed event.

At each time step, ELPH attempts to learn which of the possible subsets of the observation history are consistently good at predicting the current event e_t. It does this by adding a hypothesis to the HSpace for each possible subset of the observation history corresponding to the currently observed event, e_t. Without loss of generality, assuming an observed history length of 7 there are $2^7 = 128$ such hypotheses:

$$\begin{aligned} \{o_{t-1}\} &\Rightarrow e_t \\ \{o_{t-2}\} &\Rightarrow e_t \\ &\vdots \\ \{o_{t-6}, o_{t-4}\} &\Rightarrow e_t \\ &\vdots \\ \{o_{t-7}, o_{t-6}, \ldots, o_{t-1}\} &\Rightarrow e_t \end{aligned}$$

Each of the individual hypotheses are inserted into the HSpace subject to the following rules:

1. If the hypothesis pattern is not in the HSpace, it is added with an associated prediction-set containing only the event e_t with its event frequency set to 1.
2. If the hypothesis already resides in the HSpace, then e_t is compared with the stored predictions in the associated prediction-set. If found, the proposed hypothesis is *consistent* with past observations and the event frequency corresponding to e_t is incremented.
3. If the hypothesis already resides in the HSpace but the observed event e_t is not found in the associated prediction-set, the novel prediction is added to the prediction-set with an event frequency of 1.

The combinatorial explosion in the growth of the HSpace is controlled through a process of active pruning. Since we are only interested in those hypotheses that provide *high-quality* prediction, inconsistent hypotheses or those lacking predictive quality can be removed. For any given hypothesis, the prediction-set represents a histogram of the probability distribution over those events that have followed the specified pattern of observations. The entropy of this distribution is a measure of the prediction uncertainty and can be considered an inverse qualitative measure of the prediction. Using the individual event frequencies, f_{e_i}, the entropy of the prediction set can be computed as,

$$H = -\sum_{e_i} \frac{f_{e_i}}{f_{e_{tot}}} \log_2 \left(\frac{f_{e_i}}{f_{e_{tot}}} \right)$$

where $f_{e_{tot}} = \sum_{e_i} f_{e_i}$is the sum of all the individual event frequencies. If a specific hypothesis is associated with a single, consistent prediction, the entropy measure for that prediction-set will be zero. If a specific hypothesis is associated with a number of conflicting predictions, then the associated entropy will be high. In this sense, the "quality" of the prediction represented by the specific hypothesis is inversely related to the entropy measure.

Those hypotheses that fail to provide consistent prediction accuracy are pruned. If the entropy of a specific hypothesis exceeds a predetermined threshold, H_{thresh}, it fails the "predict with high probability" test and is no longer considered a reliable predictor of future events, so it is removed from the HSpace. Over time, only those

hypotheses deemed accurate predictors with high probability are retained. Entropy threshold pruning also facilitates rapid adaptation in non-stationary environments. When the underlying process statistics change, the resultant increase in prediction-set entropy causes existing hypotheses to be removed and replaced by low-entropy hypotheses learned following the change.

Using entropy as a qualitative prediction measure also provides a mechanism to infer future events from the current observation history. To make a prediction given a sequence of observations, we locate the hypotheses in the HSpace which are consistent with the current contents of the observation history and rank them. For ranking predictions, a simple entropy computation is not sufficient because it is biased toward selecting those hypotheses with a small number of occurrences. For example, a hypothesis that has only occurred once will have a single prediction-set element, producing a computed entropy value of zero. Instead we use a more *reliable entropy* measure, obtained by recomputing the prediction-set entropy with the addition of a single, hypothetical false-positive element representing an implicit prediction of "something else". This reliable entropy automatically discounts infrequently occurring hypotheses.

Given an observation history of length n, the maximum number of matching hypotheses is $2^n - 1$. The most frequently occurring prediction (maximum likelihood) from the single hypothesis with the lowest reliable entropy is the best prediction that can be made from the current observations.

Statistical structure in the observation space leads to efficient pruning: If the temporal stream of observations is truly random, resulting in the inability to predict future events, then ELPH will continually prune and add new hypotheses (i.e. thrash). However, most interesting domains possess regularities our algorithm should efficiently exploit.

Using the ELPH algorithm to observe the actions of another agent, we can learn the predictive elements of that agent's policy. In the case of game playing, this capability can be used to exploit the learned policy of the opponent to select superior plays in those cases where the opponent is acting predictably. The overall strategy is to ascertain predictability *bias* in the opponent's play, predict what the opponent is most likely to do next, and choose a play that is superior to that predicted for the opponent. If the opponent exhibits predictable behavior, the policy learning agent can exploit that bias and achieve a statistical edge.

Methods and Experimental Results

We pitted the ELPH algorithm against a collection of both synthetically generated agents and human players in the game of Rock-Paper-Scissors. The synthetic agents produced a series of 1,000 plays according to a stochastic policy or a non-stationary series of either pure (deterministic) or mixed (stochastic) policies. ELPH was used to learn the non-stationary policy of the agents, and play accordingly.

Synthetic stochastic agents

The simplest agent class is one in which all plays are purely random. In this case, we simply generated a series of 1,000 plays from a uniformly random transition matrix U. This agent is playing a stationary policy at the Nash equilibrium. In this case, the best policy that ELPH can employ is also random. The number of wins/losses as well as ties should be roughly equal for both ELPH and the synthetic agent. Figure 1 shows that ELPH and the opponent approximately break even, each winning 1/3 of the trials in this case.

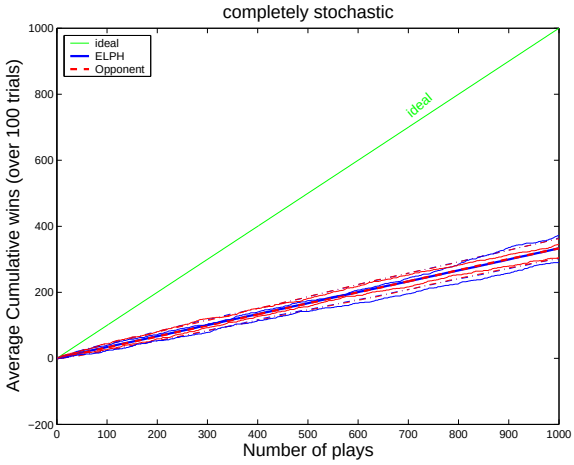

Figure 1: ELPH accumulated wins over time against a purely stochastic synthetic agent. Results are for 100 trials of 1000 plays. We show the mean and plus/minus twice the standard deviation.

Synthetic non-stationary deterministic agents

This synthetic agent class generated a series of 1,000 plays from a non-stationary collection of randomly chosen deterministic policies. A specific policy was chosen at random and used to generate n consecutive plays, where n was also chosen randomly from a Poisson distribution with $\mu = 20$. After generating the n plays, a new policy and new n were chosen. This process was repeated until a total of 1,000 plays were generated.

Each specific policy was constructed by filling a 3×3 matrix with exactly one "1" in each row, where the column position of each "1" was chosen at random uniformly from the set $\{1, 2, 3\}$ corresponding to the states $\{rock, paper, scissors\}$. All the remaining matrix entries were set to 0.

The resulting transition matrices, though deterministic, are not ergodic. They may be cyclic and/or reducible. They could produce degenerate cases such as a fixed play (i.e. rock, rock, rock ...).

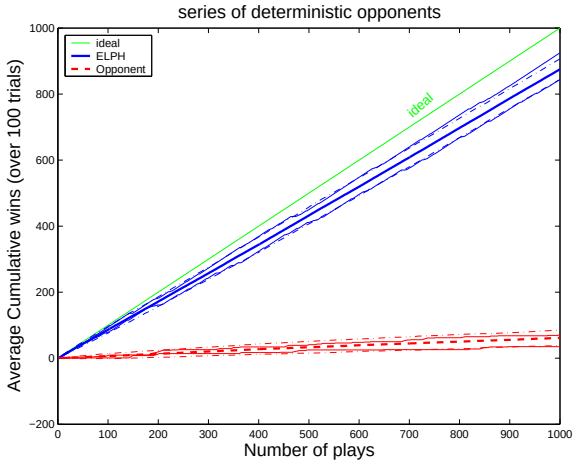

Figure 2: ELPH accumulated wins over time against a synthetic agent playing from a non-stationary set of randomly selected deterministic policies. Results are for 100 trials of 1000 plays. Each policy was played for a randomly selected time determined from a Poisson distribution with $\mu = 20$.

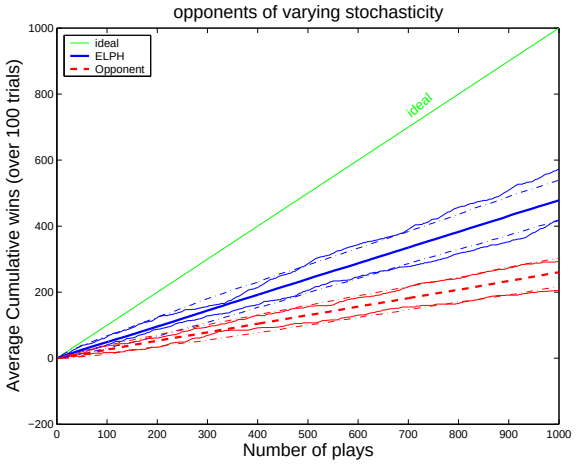

Figure 3: ELPH accumulated wins over time against a synthetic agent that plays from a non-stationary set of stochastic policies of the form (1). Results shown for 100 trials of 1000 plays. Each policy was played for a randomly selected time chosen from a Poisson distribution with $\mu = 20$.

When ELPH was matched against the non-stationary deterministic synthetic opponent, it was able to both quickly learn the opponent's active policy and rapidly adapt to the individual policy changes. The procedure used to generate the synthetic agent's play results in a uniform distribution of actions from an overall frequency point of view, but due to the deterministic nature of each individual policy, there is significant smaller scale structure. ELPH exploits this structure by rapidly adapting to the policy changes and quickly learning the new policy. This behavior is detailed in Figure 2, where ELPH wins nearly 90% of the plays, even though the agent is changing policies approximately every 20 plays.

Synthetic non-stationary stochastic agents

This synthetic agent was constructed like the preceding one, but using stochastic policies, as follows.

Randomly select a purely deterministic transition matrix D according the same procedure as before. Define U to be the uniformly random transition matrix (all entries equal to $1/3$). Construct T as the convex sum of D and U:

$$T = \lambda(D) + (1 - \lambda)(U), \qquad (1)$$

where λ is a pseudo-random number chosen uniformly from the interval $(0, 1]$.

The matrix T will always be ergodic even though, for λ near 1, the matrix will be highly deterministic.

For this agent, we experimented with different ways of switching between the stochastic and deterministic processes.

Figure 3 details results for play against a non-stationary agent that is randomly selecting mixed policies at the same rate as in Fig. 2. Here the performance is degraded, but ELPH is still able to exploit the times when the λ value is high. ELPH is still able to win more plays than the opponent in every case.

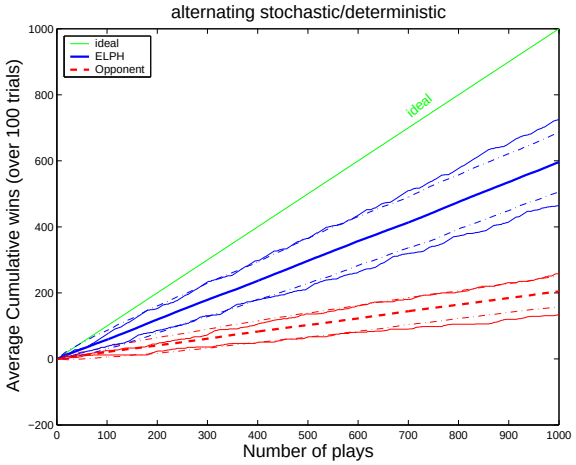

Figure 4: ELPH accumulated wins over time as in Fig. 3, but with $\lambda = 0, 1$ chosen randomly, each with probability $1/2$. Each policy was played for a randomly selected time chosen from a Poisson distribution with $\mu = 20$.

Figure 4 shows the results when λ is restricted to take on extreme values 0 and 1. Figure 5 illustrates how the ELPH performance decreases when opponent's policy changes more often, but ELPH still outperforms the opponent.

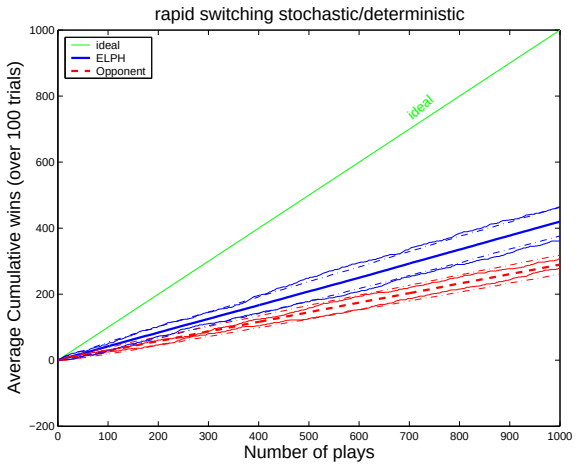

Figure 5: Same as Figure 4 except that each policy was played for a randomly selected time chosen from a Poisson distribution with $\mu = 3$.

Human opponents

Play against human opponents involved an interactive version of Rock-Paper-Scissors. The game was set to end when either player's accumulated reward exceeded 100. If a human opponent plays rationally, according to the Nash equilibrium, he/she should ultimately play randomly to maximize return. However, humans have great difficulty acting randomly. The observed behavior appears to rather be one of constantly trying "different" approaches in an effort to "fool" the opponent. The working hypothesis in this case is that humans will exhibit biased play which can be exploited by an agent that is able to quickly learn and adjust to the non-stationarity of the overall policy.

A multiple context ELPH approach was used to learn two separate temporal observation streams in parallel. The first stream consisted of the consecutive plays of the opponent and was used to predict the opponent's subsequent play. The second stream was used to predict the opponent's next play based on the sequence of ELPH's plays. In this way, if the opponent exhibits biased patterns related to his/her own play, the first stream provides predictors, whereas if the opponent attempts to exploit perceived patterns related to the machine's play, that bias will be detected and exploited. The approach is simple. Observe, make two predictions of the opponent's next play based on the separate input streams, and select the play that has the lowest reliable entropy measure.

The results against human opponents are less pronounced, but demonstrate ELPH performance when confronted by a non-stationary policy in which the time scale and selection process is completely unknown. In this case ELPH is able to exploit predictive bias in the human opponent's play. Figure 6 details one representative match. As shown in this example, an advantage was gained following approximately $35 - 40$ plays. It adapts to the changing play of the opponent and quickly exploits predictive patterns of play.

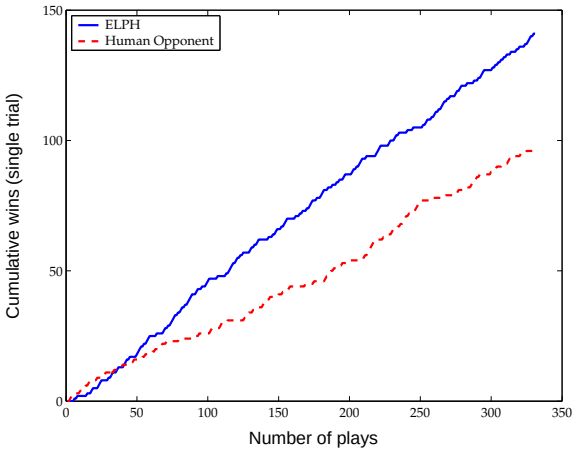

Figure 6: ELPH accumulated wins over time against a human opponent. Results shown for a single match.

Discussion

Statistical estimation methods such as WoLF-PHC act according to their best estimate of the optimal policy and then they modify that estimate based on the observed outcome. In multiagent environments, that outcome (reward) is a function of the opponent's policy, and is assumed to change over time. Owing to this dependency, these learning methods are, in effect, forming an *indirect* estimate of the opponent's policy over time. The indirect estimate of the opponent's policy is learned through an exploratory or "probative" process of trying some action and observing the opposing agent's response. In a non-stationary domain, this process may never fully arrive at an adequate estimate of the opponent's (instantaneous) policy. These methods also require some stated prior estimate on the policy space. In most cases, the initial estimate of the optimal policy is assumed to be a uniform distribution over actions. If infinitesimal gradient ascent is employed, it usually takes an unacceptably long time to converge to the optimal policy.

ELPH, on the other hand attempts to learn the opponent's policy directly, without exploration. It learns over a space of observed behavior hypotheses. This is a decidedly distinct way of approaching the problem. Assuming the opponent is playing according to some policy (rational or otherwise), ELPH generates a collection of hypothetical states from the observed action sequence and selects those that prove consistent with the estimate of the opponent's policy. ELPH requires no notion of "winning" or "losing". It simply adapts by abandoning inconsistent predictions and acquiring new ones based on the shift in policy space.

For WoLF-PHC, convergence to an optimal policy is guaranteed in the long term. For two-player zero sum

games, it will ultimately converge to the Nash equilibrium and play randomly. In the domain presented here, WoLF-PHC was unable to gain any advantage when playing against same synthetic opponents as ELPH due to the fact that the sequence of plays was too short.

The ELPH hypothesis pruning mechanism can keep up with the changes in the opponent. This yields substantial advantages in the non-stationary domains presented here. When the opponent plays his/her optimal strategy (random play), ELPH will respond by playing randomly. But ELPH will quickly pick up on some non-random structure in the opponent's play.

Related work

The ELPH algorithm can be viewed as a method to learn a sparse representation of an order-n Markov process via pruning and parameter tying. Because sub-patterns occur more frequently than the whole, the reliable entropy measure preferentially prunes larger patterns. Because prediction is then performed via the best sub-pattern, this effectively ties probability estimates of all the pruned patterns to their dominant sub-pattern.

Previous approaches to learning sparse representations of Markov processes include variable memory length Markov models (VLMMs) (Guyon & Pereira 1995; Ron, Singer, & Tishby 1996; Singer 1997; Bengio *et al.* 1998) and mixture models that approximate n-gram probabilities with sums of lower order probabilities (Saul & Jordan 1998). VLMMs are most similar to our approach in that they use a variable length segment of the previous input stream to make predictions. However, VLMMs differ in that they use a tree-structure on the inputs, predictions are made via mixtures of trees, and learning is based on agglomeration rather than pruning. In the mixture approach, n-gram probabilities $p(o_t|o_{t-1} \ldots o_{t-n})$ are formed via additive combinations of 2-gram components. Learning in mixture models requires the iterative EM method to solve a credit assignment problem between the 2-gram probabilities and the mixture parameters. ELPH does not require any iterative algorithm at each step.

Rock-paper-scissors is one of the stochastic games used by Bowling and Veloso (Bowling & Veloso 2002) as a demonstration of their WoLF algorithm. WoLF (Win Or Learn Fast) applies a variable learning rate to gradient ascent over the space of policies, adapting the learning rate depending on when a specific policy is winning or losing. The WoLF principle is to learn quickly when losing and more cautiously when winning. In contrast to this work, ELPH completely ignores the reward or whether it is winning or losing. ELPH simply makes predictions based on past observations and discards past knowledge if it fails to predict future play. ELPH makes no assumption on the rationality of the opponent's policy. If the opponent exhibits *any* predictability in play, ELPH will exploit it and choose an action that will better the opponent with a frequency matching the statistical bias. If the opponent plays purely randomly, then ELPH is capable of playing to a draw.

Conclusions and Future Work

We have described an approach for learning to predict temporal sequences that is robust to non-stationary generative processes, and demonstrated a simple application of the approach in playing 2-player zero-sum matrix games. ELPH is shown to exhibit both rapid learning and rapid adaptation to non-stationary policies, even when the policy and the time period are chosen randomly.

Future work on ELPH is focused on extension to more complex domains and on learning higher-order sequences that repeat in time periods greater than 7 events.

References

Agrawal, R., and Srikant, R. 1995. Mining sequential patterns. In Yu, P. S., and Chen, A. S. P., eds., *Eleventh International Conference on Data Engineering*, 3–14. Taipei, Taiwan: IEEE computer Society Press.

Bengio, Y.; Bengio, S.; Isabelle, J.-F.; and Singer, Y. 1998. Shared context probabilistic transducers. *NIPS'97* 10.

Bowling, M., and Veloso, M. 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence* 136:215–250.

Fudenberg, D., and Levine, D. K. 1999. *The Theory of Learning in Games.* Cambridge, Massachusetts: MIT Press.

Guyon, I., and Pereira, F. 1995. Design of a linguistic postprocessor using variable memory length Markov models. *International Conference on Document Analysis and Recognition* 454–457.

Owen, G. 1995. *Game Theory.* Academic Press.

Ron, D.; Singer, Y.; and Tishby, N. 1996. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning* 25.

Saul, L. K., and Jordan, M. I. 1998. Mixed memory Markov models: decomposing complex stochastic processes as mixtures of simpler ones. *Machine Learning* 1–11.

Singer, Y. 1997. Adaptive mixture of probabilistic transducers. *Neural Computation* 9.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction.* Adaptive Computation and Machine Learning. MIT Press.