Adaptive Learning as Potential Descent

CS 6998 – Advanced Computational Learning Theory
Final project report

Mathew Davies
mdavies@cs.columbia.edu

June 15, 2005

1 introduction

Adaptive learning techniques are used in numerous algorithms for classification, prediction and strategic game play[1], including boosting. However, these techniques are not unique to computational learning theory. Adaptive learning approaches are also used in the social sciences[2], particularly in stochastic game theory.

The goal of this paper is to show that there exist significant connections between adaptive learning in contemporary game theory, and adaptive learning in computational learning theory. For instance, the GRL model of adaptive learning for binary choice games [5], a particular case of which is the Roth-Erev (RE) stochastic learning model, is related to Hart and Mas-Collel's regret-matching algorithm for finding correlated equilibria [10]. Both algorithms (along with several other important learning procedures) are special cases of the potential-descent framework of Cesa-Bianchi and Lugosi [3]. This framework is important for at least two reasons. First, it permits the generalization of a large number of important adaptive algorithms in computer science as well as in game theory. Second, it gives a new theoretical basis for the derivation of bounds on loss and convergence which can in some cases be applied to learning models in the social sciences, as we will show with the RE model.

The connections between adaptive learning in game theory and computer science can be seen as instances of the relationship between artificial intelligence and game theory discussed by Tennenholtz [17]. In particular, Tennenholtz cites three fundamental issues of relevance in both fields: reasoning and rationality in distributed environments, learning in uncertain

[1] That is, finding solutions to repeated formal games. Although learning techniques are also used in some adversarial game-playing algorithms, the term "game" in this paper refers only to extended-form and repeated games as defined in game theory.

[2] Models of adaptive learning in the social sciences are sometimes described as "reinforcement learning" or "learning-theoretic"; see the note on terminology at the end of the introduction.

environments, and the representation of behavior and decision processes. The importance of this relationship is attested by the growing number of computer scientists doing research in game theory and decision theory, and the growing number of researches in game theory and other disciplines applying computational methods of analysis to problems in the social sciences.

This paper draws heavily on literature from the intersection of game theory and computer science. A central theme in this literature, which is not new but has gained additional substance from powerful analytical techniques in the last few years, is that boundedly rational approaches to choice in games can lead to outcomes that are better (from a social perspective) than fully rational approaches. In this paper, we attempt to illustrate this point, following one particular thread – the notion of adaptive learning – from computational learning theory, through stochastic game theory, to the dynamics of cooperation in social dilemma games. This thread and many others constitute a fabric of understanding that we believe can yet address some of the most pressing dilemmas of the modern age.

1.1 outline of the report

In section 2, several adaptive algorithms from both computer science and from economic game theory are presented in their original contexts. In section 3, these algorithms are shown to be special cases of a general potential-descent framework. In addition to summarizing results of [3], we show that the logit equilibrium (introduced in section 2) can be cast in this framework. In section 4, we briefly discuss the role rationality in the context of adaptive game play, review relevant work by Flache and Macy [5] on stochastic learning in social dilemmas, derive a version of the RE model as a potential-descent algorithm, and sketch a derivation of analytic upper bounds on the time complexity of reaching a stable equilibrium in the RE model. In section 5, we discuss some of the relationships exposed among these algorithms, and their implications to the problems of cooperation and coordination in distributed environments.

1.2 reinforcement learning, or adaptive learning?

Behavioral models in stochastic game theory can be seen as refinements of classical game theory in which forward-looking rationality is partially or wholly replaced as a determinant of choice by backwards-looking adaptation, sometimes modeled as a stimulus-response process and sometimes as a true learning process. In the social sciences, such models are often distinguished with the term "reinforcement learning", or "learning-theoretic". Because both of these terms in computer science refer to related but very distinct concepts from their counterparts in the social sciences, I adopt the term *adaptive* to describe algorithms in both computational learning theory and stochastic game theory in which hypothesis parameters (weights or probabilities) are adjusted in each round of play or training according to the results of the previous round.

While learning in computer science generally refers to well-defined computational classification or prediction models, learning in the social sciences often refers to mechanisms explaining human choice or behavior. While not unrelated, the differences between usage of the term reflect important differences between the two disciplines. In this paper, *adaptive learning* refers to binary classification and discrete-choice prediction algorithms in computer science, as well as to methods for constructing stochastic choice rules in the theory of repeated games. The relationship between adaptive learning in each field will be discussed in the conclusion.

1.3 notation

In section 2, we follow the notation used by the authors whose analyses we discuss, with small modifications for consistency. Generally speaking, component i of a vectorlike quantity, say $\mathbf{p}$, at time t is written $\mathbf{p}_t(i)$. In section 3 and subsequently, we follow the notation used by Cesa-Bianchi *et al.* In particular, the component i of a vector $\mathbf{r}$ at time t is written $r_{i,t}$ (and similarly for vectors $\mathbf{R}$, $\mathbf{p}$, etc.).

The phrases "time t" and "round t" are used interchangeably throughout. The terms *algorithm* and *procedure* are also used interchangeably, although "algorithm" is used preferentially for well-known algorithms in computer science, while "procedure" is often used when discussing algorithms or game strategies in the context of economic game theory, or when discussing algorithms and procedures from both disciplines as a class.

2 adaptive algorithms in computer science and game theory

In this section, I present the following four important adaptive algorithms, or learning procedures.

- In computational learning theory, *boosting* and *adaptive game playing* were developed as distict approaches to very different problems, yet they share significant game-theoretic connections.
- In stochastic game theory, *regret matching* is one of several solution concepts in which stochastic choice according to simple adaptive rules leads to correlated equilibria in iterated games, while the *logit equilibrium* can be interpreted as a bridge between evolutionary dynamics of a population, and learning dynamics of individuals.

These four algorithms, which as discussed below have been described and analyzed by various authors, are related – perhaps contrary to appearances! The relationships are discussed briefly, then in the next section these algorithms are recast as instances of a single general potential-minimizing framework, following the exposition in [3].

2.1 boosting (AdaBoost)

Although boosting has numerous forms and applications, surveyed by Schapire in [16], I discuss here only boosting of binary classifiers using the well-known AdaBoost (adaptive boosting) algorithm introduced by Freund and Schapire [6]. The AdaBoost algorithm "boosts" the performance of an ensemble of weak classifiers produced by a *weak oracle* (whose accuracy of classification on their *training set* V of examples is better than one half, but not necessary very much better) by using them to create an arbitrarily strong classifier. The following description of the algorithm follows [16], with minor label changes for the context of later discussion in this paper.

Algorithm (AdaBoost)

For round $t = 1, ..., T$,

- Train the weak oracle using the distribution $\mathcal{D}_t$ of the training set of N examples $(v_1, l_1), ..., (v_N, l_N) \in V \times \{-1, 1\}$
- Get a weak classifier $h_t : V \mapsto \Re$
- Choose $\alpha_t \in \Re$
- Update:

$$\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i) \exp\left(-\alpha_t l_i h_t(v_i)\right)}{Z_t} \text{ for } i = 1, ..., N \tag{2.1}$$

  where Z_t is an appropriate normalization constant

Output $h^*(v) = \textbf{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(v)\right)$

Let ϵ_t be the accuracy of the classifier h_t on the training set, let $\gamma_t \equiv 1/2 - \epsilon_t$ be the *advantage* of hypothesis h_t, and let γ be the maximum value such that $\gamma_t \geq \gamma$. Then the error of the final classifier h^* is at most $e^{-2T\gamma^2}$. The typical value for α_t is

$$\alpha_t = \frac{1}{2} \ln \frac{(1 - \epsilon_t)}{\epsilon_t}$$

As Schapire notes, AdaBoost is a procedure for finding a linear combination of classifiers which attempts to minimize the expression $\sum_{i=1}^{N} \exp\left(-l_i h^*(v_i)\right)$, or

$$\sum_{i=1}^{N} \exp\left(-l_i \sum_{t=1}^{T} \alpha_t h_t(v_i)\right) \tag{2.2}$$

which upper bounds the total number of misclassifications of h^* on the training set, by doing a kind of steepest-descent search while varying α_t and h_t [16]. We return to this point in the next section, where AdaBoost is recast as an explicit potential-minimization algorithm.

2.2 adaptive game playing with multiplicative weights (MW)

The following presentation of MW is adapted from Freund and Schapire [7], the originators of the algorithm. MW itself is a generalization of the Weighted Majority algorithm of Littlestone and Warmuth [13], which was developed in the context of decision theory.

MW is designed for the problem of playing repeated two-player games. The first player is designated the *learner*, whose goal is to learn a strategy minimizing the loss received from the game in each round. The second player may be thought of as another player, or generically the *environment* (a known player, an aggregation of results of other players' choices, or some unknown agent). In the case where the entire matrix $\mathbf{M}$ of the game is known and the game is maximally adversarial, linear programming may be used to find the minmax value of the game. However, in cases where $\mathbf{M}$ is not known in advance or is too large to compute, or the environment is not necessarily adversarial (i.e. the goal of the environment is not necessarily to maximize the player's loss), then another approach is desirable. MW can be used in exactly this setting.

In the following description, let $\mathbf{p}$ and $\mathbf{q}$ denote the mixed strategies of the learner and the environment, respectively, over the set of actions (pure strategies) available to them, not necessarily the same. Let $\mathbf{p}_t(i)$ denote the learner's probability of playing action i in round t of play. Let $\mathbf{M}(\mathbf{p}, \mathbf{q}) = \mathbf{p}^T \mathbf{M} \mathbf{q}$ denote the learner's expected loss when the two mixed strategies are used, and $\mathbf{M}(i, \mathbf{q})$ or $\mathbf{M}(\mathbf{p}, j)$ denote the expected loss when the player or the environment, respectively, uses a pure strategy and the other a mixed strategy. $\mathbf{M}(i, j)$ denotes the loss received by the learner playing action i, when the environment plays action j. Finally, let the entries of $\mathbf{M}$ be scaled so that $\mathbf{M}(i, j) \in [0, 1]$ for all i, j.

Algorithm (MW)

For round $t = 1, ..., T$,

- The learner chooses $\mathbf{p}_t$, as described below.
- The environment chooses $\mathbf{q}_t$, possibly with knowledge of $\mathbf{p}_t$
- The learner observes the loss $\mathbf{M}(i, \mathbf{q}_t)$ for each row i, and suffers the aggregate loss $\mathbf{M}(\mathbf{p}_t, \mathbf{q}_t)$.

The goal of this algorithm is to minimize the total loss $L \equiv \sum_{t=1}^{T} \mathbf{M}(\mathbf{p}_t, \mathbf{q}_t)$. Note that this loss can be much better than the minmax value of the game, if the environment is not maximally adversarial.

In the learner's choice step, the mixed strategy $\mathbf{p}$ is initially uniform, and thereafter updated according to the rule

$$\mathbf{p}_{t+1}(i) = \mathbf{p}_t(i) \frac{\beta^{\mathbf{M}(i, \mathbf{q}_t)}}{Z_t} \tag{2.3}$$

where $\beta \in [0, 1)$ is a constant of the algorithm, and Z_t is again an appropriately chosen normalization factor, so that $\mathbf{p}$ represents a proper probability distribution.

Schapire notes, in his overview of boosting [16], that the similarity between the MW update rule and AdaBoost's update rule is not inconsequential: boosting can be viewed as repeated play of a particular zero-sum game $\mathbf{M}'$, where the training examples correspond to player choices, and the space of weak classifiers to the environment's choices. The entry $\mathbf{M}'(i, j)$ is 1 if h_j classifies example x_i correctly, and 0 otherwise. The boosting algorithm's choice of sample distribution becomes the choice of $\mathbf{p}$ over rows of the game matrix, while the environment's choice of classifier becomes the choice of $\mathbf{q}$. Hence, MW can be seen as a kind of generalization of AdaBoost. In fact, both algorithms are instances of the potential-descent approach discussed in the next section.

The MW algorithm is important because not only does it give a method for playing repeated games when the game itself and the environment's strategy is unknown, but this method also guarantees that

- The expected per-trial average loss (in comparison to the best fixed strategy) is $O(\sqrt{\frac{\ln N}{T}})$, which approaches zero in the asymptotic limit as $T \to \infty$.
- The actual per-trial loss is at most $O(\sqrt{\frac{1}{T}})$ from the optimal loss possible.
- In the asymptotic limit ($T \to \infty$), no other adaptive algorithm can beat the above bound on convergence; MW is in this sense optimal.

These results, proved in [7], will be discussed further in the next section.

We now turn to two decision procedures important in game theory. Following their sources, we discuss these procedures somewhat informally, rather than as precise algorithms, concentrating on the probabilistic choice rules that form the substance of each decision procedure.

2.3 regret matching

Numerous authors cited in this paper discuss the concept of *regret*, which is related to loss as follows. Suppose a player in a game chooses action i rather than j. The regret of i over j (often denoted $R_t(i, j)$) is the difference in the expected loss of playing j instead of i in round t. Hart and Mas-Colell describe the following procedure, known as *regret-matching* [10], which adapts choice probabilities in each round to match the cumulative regret of each choice. They prove that regret-matching will eventually bring the game to a correlated equilibrium. The correlated equilibrium, an important concept in game theory, is defined here (following [10]).

For some game Γ, let $s \in S$ be a strategy profile of a set A of players, where $S = \prod S^a$ is the cartesian product of the sets of strategies S^a available to each player a. Let $i, j \in S^a$ be actions (pure strategies) available to a, and let $u^a(s)$ be the utility of s to player a. For

convenience, write $s \equiv (s^a, s^{-a})$, where s^a denotes player a's action, and s^{-a} the actions of the remaining players. Then a probability distribution Ψ of strategies in S is a *correlated equilibrium* of Γ if for every player $a \in A$, and every pair of actions $i, j \in S^a$, the following inequality holds

$$\sum_{s \in S : s_a = i} \Psi(s)[u^a(j, s^{-a}) - u^a(s)] \leq 0$$

This inequality simply asserts that, for any choice i, the expected payoff to a, given the choices of the other players, is at least as good as any other choice j would have been.

In the following discussion, in order to simplify matters, we discuss the strategy of a single player, and omit the superscript a; all values of utility, etc. are implicitly with respect to this player. If we define the loss L of a strategy s as $L(s) = C - u(s)$, where C is the maximum utility over strategies, then we can define the regret of an action i over j in round t (with respect, implicitly, to the unchanged actions of other players) precisely as

$$r_t(i, j) = L(i) - L(j) \tag{2.4}$$

If the loss of playing j would have been less than for playing i, the player feels "regret" for having played i instead of j.[3] The cumulative regret is then $R_t(i, j) = \sum_{t=1}^{T} r_t(i, j)$.

The player starts with a uniform strategy. After playing action j, the probability of choosing an action $i \neq j$ is then updated in each round according to

$$p_{t+1}(i) = \frac{1}{\mu} R_t(i, j) \tag{2.5}$$

while $p_{t+1}(j) = \sum_{i \neq j} p_{t+1}(i)$, and the constant $\mu > 0$ controls the rate at which learning takes place. Hence, regret-matching emphasizes playing actions for which the regret (of not playing them) is largest. Hart and Mas-Colell show, using Blackwell's approachability theorem,[4] that in the limit of large T the regret-matching strategy drives the cumulative regret (over all rounds) down to zero, bringing the strategy distribution to a correlated equilibrium. Thus, the regret-matching strategy is *Hannan consistent*, a concept discussed below and in section 3.2. Hart and Mas-Collel have described a large class of Hannan-consistent adaptive strategies [11]. In section 4, we will argue that the Roth-Erev payoff-matching model can be seen as a kind of modified regret matching which leads to an approximate correlated equilibrium.

2.4 logit equilibrium

Regret matching is one of a number of backward-looking adaptive strategies leading to correlated equilibrium. Here, we briefly discuss a kind of backward-looking equilibrium, the

[3] Hart and Mas-Colell constrain $r_t(i, j)$ to be positive, so that regret is defined to be zero when the loss of j exceeds the loss of i.

[4] See [2], as cited in [10].

logit equilibrium, that is important in evolutionary game theory. The following description of logit equilibrium and its relationship to evolutionary dynamics is due to [8].

Suppose an individual in a game chooses from among N actions according to a strategy vector $\mathbf{p}$, but also maintains beliefs about the opponent's expected actions. Let π^e be the expected payoff, where the time index t is left implicit. The *logit probabilistic choice rule* (LPCR) specifies that the stragegy $\mathbf{p}$ in any round is determined by the expected payoff according to

$$p_i = \frac{\exp\left(\pi^e(i)/\mu\right)}{Z}, i = 1, ..., N \tag{2.6}$$

where $\mu > 0$ is a scaling factor that determines the importance of payoff size as a determinant of choice, and Z is an appropriate normalizing factor over the N actions.

If we make the not-unreasonable assumption that the expected payoff for i is determined as a weighted average of T previous actual payoffs π, that is,

$$\pi^e(i) = \sum_{t=1}^{T} p_i \pi(i, y_t)$$

where y_t was the opponent's action at round t, then the LPCR can be recast as a predictor for a gradient-descent algorithm. This will be accomplished in the next section, where we will see that the similarity of the LPCR to the update rules described for the other algorithms above is not coincidental.

The logit probabilistic choice rule can be interpreted as a bridge between evolutionary and cognitive approaches to bounded rationality in games. Consider a population of players who in each round t make a decision $x(t)$ according to the distribution $F(x,t)$; the population density is given by $f(x,t)$. Assuming that evolutionary pressures will push players in the direction of higher payoffs, decisions evolve at a rate of change proportional to the time derivative of expected payoff, plus a term for random shocks analogous to Brownian motion: $dx = \pi^{e\prime}(x,t)dt + \sigma dw(t)$. This process of individual adjustment translates into the following differential equation for the population distribution of choices.

$$\frac{\partial}{\partial t}F(x,t) = -\left(\frac{d}{dt}\pi(x,t)\right)f(x,t) + \mu\frac{d}{dt}f(x,t)$$

This is the Fokker-Planck equation of statistical physics, in a new context. The logit probabilistic choice rule (2.6) corresponds to the equilibrium state of the Fokker-Planck equation, i.e. when the rate of change of $F(x,t)$ with time is zero. Hence, payoff-maximizing (or loss-minimizing) behavior can be interpreted as having the same effect (in equilibrium) as evolutionary pressure. In a somewhat similar manner, the class of Hannan consistent strategies mentioned in section 2.3 above can be shown to attain a correlated equilibrium state corresponding to classical utility maximization, without the assumption any forward-looking optimization on the part of the learner. The relationship of adaptive learning to bounded rationality is discussed further in section 4.1.

3 adaptive learning as potential descent

We now turn to the potential-descent framework of Cesa-Bianchi and Lugosi [3], from which each of the above learning procedures can be derived as special cases.[5] The derivation facilitates a comparison of loss bounds of the game-theoretic procedures, as well as illuminating the relationship among them. In the next section, we use this framework to recast the well-known Roth-Erev payoff-matching model, and to derive a new upper bound on the time complexity of reaching equilibrium in the model.

The following framework, which is presented in a greatly compressed form, can be applied to generate a wide class of adaptive algorithms for any abstract repeated decision problem involving a decision-maker and a decision environment, as long as certain constraints (discussed below) are satisfied.

- Start with an abstract decision space $\mathcal{X}$, and outcome space $\mathcal{Y}$, and a potential function $\Phi : \Re^N \mapsto \Re^+$.
- At each step $t = 1, 2, ...$ of the algorithm, the current state of the problem is represented by a point $\mathbf{R}_{t-1} \in \Re^N$, where $\mathbf{R}_0 = \mathbf{0}$.
- The decision maker observes a *drift function* $\mathbf{r}_t : \mathcal{X} \times \mathcal{Y} \mapsto \Re^N$. The decision maker then selects some decision $x_t \in \mathcal{X}$ and receives an outcome $y_t \in \mathcal{Y}$; the state of the problem is then "drifted" to the new point $\mathbf{R}_t = \mathbf{R}_{t-1} + \mathbf{r}_t$.
- The goal is to find a stochastic *predictor* $\mathbf{p}_t$, updated in each round t, which will minimize $\Phi(\mathbf{R}_t)$ for a given t', not necessarily known to the decision maker, and possibly as $t' \to \infty$.

The vector $\mathbf{r}_t$ may be thought of as the "regret" or loss of the decision made in round t, while $\mathbf{R}_t$ is the *cumulative regret* of the sequence of decisions. The purpose of the potential is simply to facilitate the construction of a predictor that is guaranteed to move the state of the problem in the direction of lowest regret. As Cesa-Bianchi *et al* show, this can be achieved fairly easily when the following two requirements are satisfied.

1. Generalized Blackwell's condition:

$$\sup_{y_t \in \mathcal{Y}} \nabla\Phi(\mathbf{R}_{t-1}) \cdot \mathbf{r}_t(x_t, y_t) \leq 0$$

2. Additive potential:

$$\Phi(u) = \sum_{i=1}^{N} \phi(u_i) \text{ for all } \mathbf{u} = (u_1, ..., u_N) \in \Re^N, \text{ where } \phi : \Re^+ \mapsto \Re^+$$

[5] Cesa-Bianchi *et al* credit work of Hart *et al* cited in this paper, as well as work by Grove, Littlestone and Schuurmans, and others, toward the development of this framework; see [3].

As mentioned earlier, a gradient-descent approach was used by Blackwell [2], and applied to sequential decision problems by Hart *et al* in [10] and [11]. A similar approach was used independently by Grove, Littlestone and Schuurmans to recast the Perceptron and Winnow algorithms. In fact, the weighted majority, p-norm and classical Perceptron, zero-threshold Winnow, and numerous other algorithms can be derived as instances of this framework.

The key to this approach is to find an auxiliary function $f : \Re^+ \mapsto \Re^+$ and a function $C : \Re^N \mapsto \Re^+$ "bounding" f such that

$$f(\Phi(\mathbf{R}_t)) \leq f(\Phi(\mathbf{0})) + \frac{1}{2}\sum_{s=1}^{t} C(\mathbf{r}_s)$$

Typically, one can find a bounded (constant) function such that $f(\Phi(\mathbf{R}_t)) \leq f(\Phi(\mathbf{0}))+ct$. If $f \circ \Phi$ is strictly convex, this is sufficient to conclude that $\mathbf{R}_t/t \to \mathbf{0}$ as $t \to \infty$, independently of the sequence of outcomes y_t.

Hence, the gradient-descent approach gives not only a loss-minimizing predictor, but also a straightforward way to analyze the expected cumulative loss when the above conditions are satisfied. This result is given as Corollary 1 of the main result of [3], which states (in a more general form than given here) that the maximum component $R_{m,t}$ of the cumulative regret $\mathbf{R}_t$, when using a polynomial potential of the form

$$\Phi(\mathbf{u}) = \sum_{i=1}^{N} (u_i)^2$$

is bounded according to

$$R_{m,t} = \sqrt{\sum_{s=1}^{t} \|\mathbf{r}_s\|^2}. \tag{3.1}$$

Although not reproduced here, an analogous result for an exponential potential is given as Corollary 2 of the main result of [3].

We now outline how the procedures for boosting (Adaboost), adaptive game playing (MW), regret matching and logit equilibrium can be derived as instances of the potential descent framework.

3.1 boosting as error potential minimization

Consistent with Shapire's remarks in [16], boosting can be viewed as a process in which the parameters α_t and the weights of training examples are chosen in order to minimize the training error of the final predictor h^*. Therefore let $\alpha_t \in \Re$ be the decision taken, and $h_t \in H$ be the outcome of the decision, where H is the space of classifiers that could be returned by the weak oracle. So $\mathcal{X} = \Re$ and $\mathcal{Y} = H$. Now considering the same set V of training examples used in the prior description of AdaBoost, let $\mathbf{r}_t$ be defined

componentwise by $r_{i,t}(\alpha_t, h_t) = -\alpha_t l_i h_t(v_i)$. Let $\Phi(\mathbf{R}_t) = \sum_{i=1}^N \exp(R_{i,t})$, where $R_{i,t} = \sum_{s=1}^t r_{i,s} = -l_i \sum_{s=1}^t \alpha_s h_s(v_i)$. The generalized Blackwell condition is then

$$\nabla\Phi(\mathbf{R}_{t-1}) \cdot \mathbf{r}_t = -\alpha_t \sum_{i=1}^N l_i h_t(v_i) \nabla_i \Phi(\mathbf{R}_{t-1}) \leq 0$$

Now consider the expression

$$\nabla_i \Phi(\mathbf{R}_{t-1}) = e^{R_{i,t-1}} = \exp\left(\sum_{s=1}^{t-1} r_{i,s}\right) = \exp\left(\sum_{s=1}^{t-1} -l_i \alpha_s h_s(v_i)\right)$$

We can see that the predictor

$$p_{i,t} = \frac{\nabla_i \Phi(\mathbf{R}_{t-1})}{\sum_{k=1}^N \nabla_k \Phi(\mathbf{R}_{t-1})} = \frac{\exp \sum_{s=1}^{t-1} -l_i \alpha_s h_s(v_i)}{\sum_{k=1}^N \exp \sum_{s=1}^{t-1} -l_k \alpha_s h_s(v_k)} \tag{3.2}$$

is exactly the update rule of AdaBoost, where $p_{i,t}$ is identified as $\mathcal{D}_t(i)$, and the recursive definition has been unraveled. In section 3.2, we show that this predictor does in fact satisfy the Blackwell condition.

Schapire notes in [16] that AdaBoost adjusts α_t and h_t on each round to minimize the expression $Z_t \equiv \sum_{i=1}^N \mathcal{D}_t(i) \exp(-\alpha_t l_i h_t(v_i))$. Over T rounds, AdaBoost thus minimizes $\prod_{t=1}^T Z_t = \frac{1}{N} \sum_{i=1}^N \exp(-l_i h^*(v_i))$, which upper bounds the training error. Rewriting this upper bound slightly, we see that AdaBoost is actually finding a minimum for the bound on the total number of misclassifications,

$$N \prod_{t=1}^T Z_t = \sum_{i=1}^N \exp\left(-l_i \sum_{t=1}^T \alpha_t h_t(v_i)\right) = \Phi(\mathbf{R}_T)$$

which is exactly the potential at round T. Hence we can think of Φ as an "error" potential which the boosting procedure descends, as noted earlier at (2.2).

3.2 potential descent in game theory: MW and regret matching

Here we review the derivation of both MW and the regret-matching procedure of Hart and Mas-Colell as regret minimization via potential descent, following [3].[6] Both procedures turn out to have the same bound on expected loss, a consequence of the fact that both procedures are Hannan consistent. The concept of *Hannan consistency*, which is discussed in depth in [11], formalizes the notion of a sequence of actions whose outcome is, in the long run,

[6] Cesa-Bianchi *et al* discuss a very general notion of regret which includes many other kinds of regret-minimizing Hannan-consistent procedures; for simplicity, we discuss only the regret matching procedure of Hart and Mas-Colell.

as good as the outcome of the best possible fixed strategy. A Hannan-consistent strategy[7] therefore drives to zero the average (per-round) cumulative regret $\mathbf{R}_t$ defined in section 3. In other words, game strategies generated by a correct potential-descent procedure – i.e. one which satisfies the necessary preconditions, in particular the generalized Blackwell's condition – will automatically be Hannan consistent.[8]

Let $\mathcal{X} = 1, ..., N$ be a set of pure strategies available to a player, with $\mathcal{Y}$ the strategies available to the adversary. In any round that the player chooses $x \in \mathcal{X}$ and the adversary chooses $y \in \mathcal{Y}$, the player receives the *payoff* $\pi(x, y) \geq 0$. If the maximum payoff for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is C, we can define a loss function $L : \mathcal{X} \times \mathcal{Y} \mapsto [0, 1]$ by

$$L(x, y) = \frac{C - \pi(x, y)}{C}$$

Now suppose that in round t of a repeated game, the player chooses according to the predictor $\mathbf{p}_t$, and the adversary chooses y_t. Define the regret $\mathbf{r}_t : \mathcal{X} \times \mathcal{Y} \mapsto \Re^N$, whose jth component

$$r_{j,t}(\mathbf{p}_t, y_t) = \left(\sum_{i=1}^{N} p_{t,i} L(i, y_t) \right) - L(j, y_t) \tag{3.3}$$

$$= \sum_{i=1}^{N} \left(p_{t,i} L(i, y_t) - L(j, y_t) \right) \tag{3.4}$$

is the expected improvement (decrease) in loss for choosing j, in comparison to the expected loss of the predictor, assuming the opponent were to make no change in strategy. In other words, r_j at any round is the "regret" the player expects for *not* playing j, and $\mathbf{r}$ is the expected regret of $\mathbf{p}$. We will show that this definition of regret is a generalization of the definition by Hart and Mas-Colell given above, in section 2.3.

Given an appropriate potential Φ, a predictor that prefers actions whose expected regret increases most quickly – i.e., for which the gradient of the potential is maximum – should lead to minimization of the cumulative regret. Hence, we can define $\mathbf{p}_t$ as the normalized gradient vector whose ith component is

$$p_{i,t} \equiv \frac{\nabla_i \Phi(\mathbf{R}_{t-1})}{\sum_{k=1}^{N} \nabla_k \Phi(\mathbf{R}_{t-1})} \tag{3.5}$$

To verify that the generalized Blackwell condition holds for this predictor, consider the jth term in the dot product $\nabla \Phi(\mathbf{R}_{t-1}) \cdot \mathbf{r}_t$,

$$[\nabla \Phi(\mathbf{R}_{t-1}) \cdot \mathbf{r}_t]_j = \nabla_j \Phi(\mathbf{R}_{t-1}) \sum_{k=1}^{N} \frac{\nabla_k \Phi(\mathbf{R}_{t-1})}{Z_{t-1}} (L(k, y_t) - L(j, y_t))$$

[7] We informally use "Hannan consistent" as an adjective for learning procedures that generate Hannan-consistent strategies.

[8] This fact is not surprising, considering that Cesa-Bianchi *et al* developed the potential-descent framework partly as a generalization of Hart *et al*'s results for Hannan consistent strategies.

where $\mathbf{r}_t$ has been expanded to its explicit form and Z_t represents the normalization factor in the denomenator of the predictor (3.5) above. For clarity, substitute for the gradient component i the symbol $G_i \equiv \nabla_i \Phi(\mathbf{R}_{t-1})$. Then

$$\nabla\Phi(\mathbf{R}_{t-1}) \cdot \mathbf{r}_t = \sum_{j=1}^{N} \sum_{k=1}^{N} \frac{G_j G_k}{Z_{t-1}} (L(k, y_t) - L(j, y_t))$$

which, due to the symmetry of the indices of summation j, k on the difference $(L(k, y_t) - L(j, y_t))$, is identically zero.

Remark. The above predictor (3.5) is exactly the predictor (3.2) used in the gradient-descent version of AdaBoost. Hence, AdaBoost's classifier-building procedure can be seen as a regret-matching procedure exactly analogous to those discussed here, where the regret is defined in terms of the error of prediction on the training set V.

It is now an easy matter to derive the update rule for the MW algorithm, equation (2.3) above, using the potential function $\Phi(u) = \sum_{i=1}^{N} e^{\eta u_i}$, where $\eta \equiv \log \beta$. The loss expected for playing action i against an adversary's strategy $\mathbf{q}$ is exactly the loss defined for MW earlier, $L(i, \mathbf{q}) = \mathbf{M}(i, \mathbf{q})$; in this case, the jth component of the regret vector simplifies to $r_{j,t}(\mathbf{p}_t, \mathbf{q}_t) = L(j, \mathbf{q})$. The jth component of the cumulative regret is thus $R_{i,t} = \sum_{s=1}^{t} L(j, \mathbf{q})$, and so the predictor (3.5) above becomes

$$p_{i,t} = \frac{\beta^{\sum_{s=1}^{t-1} \mathbf{M}(i,\mathbf{q})}}{Z'_t}$$

where Z'_t is the appropriate normalization over the index i on the player's choices. This is just an unravelled version of the recursive definition of the MW update rule (2.3), above. Using the results of [3] for the exponential potential returns the result of Schapire and Freund recorded in section 2.2, namely that difference in loss per trial from the optimal is at most $O(\sqrt{\frac{\ln N}{T}})$ at round T.

The regret-matching procedure of Hart and Mas-Colell can be derived by letting the decision space $\mathcal{X}$ be the set of pairs of actions (i, j) available to the learner. The regret defined by

$$r_{(i,j),t}(\mathbf{p}_t, y) = p_{j,t}(L(j, y) - L(i, y)), \tag{3.6}$$

which is interpreted as the regret for choosing i instead of j, is simply the expected value of the regret (2.4) defined by Hart and Mas-Colell. The polynomial potential defined by $\Phi(\mathbf{u}) = \sum_i (u_i)_+^2$, where a_+ denotes $\max\{a, 0\}$, with the predictor (3.5), gives a predictor whose ith component is

$$p_{i,t+1} = \frac{(R_{(i,j),t})_+}{Z_t}$$

which differs from the update rule (2.5) of Hart and Mas-Colell only in the absence of the (fixed) constant factor. Because the regret (3.6) is just a form of the regret (3.4) above, we are assured that the Blackwell condition holds. Cesa-Bianchi et al show, using their result (3.1) for a slightly different polynomial potential than the one given here, that the maximum component of the regret $\mathbf{R}_t$ is bounded by

$$R_{m,t} \leq \sqrt{t \ln N}$$

which is exactly the bound on the per-round cumulative regret of the MW algorithm. A similar $O(\sqrt{t})$ result will be very easily derived for the Roth-Erev payoff-matching model in section 4.3, where we discuss the relationship between regret matching and payoff matching.

3.3 logit equilibrium

The logit equilibrium can also be expressed as an outcome of an adaptive potential-minimizing procedure. In this case, $\mathcal{X}$ and $\mathcal{Y}$ are the space of choices available to an individual and the individual's environment, respectively. Let $\pi(x, y) \in \Re$ be the observed payoff from choices $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, and let $\pi(x, y) > 0$ for all $x, y \in \mathcal{X} \times \mathcal{Y}$. Let the regret $\mathbf{r}_t : \mathcal{X} \times \mathcal{Y} \rightarrow \Re$ in each round be defined by $r_{i,t} = -p_{i,t}\pi(i, y_t)/\mu'$, where μ' is a positive constant and $p_{i,t}$ is, as usual, the ith component of the predictor at round t; in other words, the ith component of the regret is the expected regret of playing action i. Then the cumulative regret is defined by

$$R_{i,t} = -\sum_{s=1}^{t} p_{i,s}\pi(i, y_s)/\mu'$$

Now, as suggested in the above discussion of the LPCR, assume that the *expected* payoff π^e is simply the expected value of the payoff over the previous T rounds (ignoring all rounds before the Tth previous one, if any), where T is some positive integer; that is,

$$\pi^e(i) = \frac{1}{T}\sum_{s=1}^{T} p_{i,s}\pi(i, y_s)$$

So $R_{i,T} = -T\pi^e(i)/\mu' = \pi^e(i)/\mu$, where in the rightmost expression the constant T has been folded into the constant μ. Let $\mathbf{p}_T = \nabla\Phi(\mathbf{R}_{i,T})/Z_T$, where Z_T is the appropriate normalization over the N actions, and let the potential $\Phi(\mathbf{u}) = \sum_{i=1}^{N} e^{-u_i}$. Dropping the time subscript for convenience, the predictor's ith component is given by

$$p_i = \frac{e^{\pi^e(i)/\mu}}{Z}$$

which is exactly the LPCR (2.6) defined earlier.

It is easy to see that the generalized Blackwell condition is satisfied, since $\nabla\Phi(\mathbf{R}_{t-1})\cdot\mathbf{r}_t = -\sum_{i=1}^{N} \exp\left(\pi^e(i)/\mu\right) p_{i,t}\pi(i, y_t)$, and the exponential, probability and payoff expressions are all nonnegative. Hence, the predictor given by the LPCR always tends towards the minimum of the potential, and therefore towards the maximum payoff.

4 adaptive learning in social dilemma games

Now that we have seen its utility in both computer science and in stochastic game theory, we use the adaptive learning framework of Cesa-Bianchi and Lugosi to give a new derivation of the well-known Roth-Erev (RE) *payoff-matching* model, described below. This derivation will permit an instructive comparison of the RE model and its cousins to the Hannan-consistent procedures outlined in the previous section.

First, we discuss rationality in the context of adaptive learning. Then we present the basic RE model, together with a very brief overview of cooperation in social dilemmas by Flache and Macy [5] based on simulations of the RE model for three games (Prisoner's Dilemma, Stag and Chicken). Next, we derive the RE model as a potential-descent procedure, with some caveats. Following our derivation, we sketch how the potential-descent framework can be used to derive analytical upper bounds on the expected lock-in time (and consequently the loss) to attain a stable equilibrium strategy.

4.1 adaptive learning and bounded rationality

Since the first work on bounded rationality by Herbert Simon in the 50s, economists and game theorists (and more recently, computer scientists) have recognized the problematic nature of classical assumptions of unbounded rationality. In game theory, regret matching and other such Hannan-consistent adaptive algorithms are important because they can generate rational (utility-maximizing) outcomes from reactive behavior, rather than proactive reasoning. In fact, many adaptive learning procedures can generate outcomes that effectively maximize utility without forward-looking computation; this is precisely the motivation for much of the interest in adaptive algorithms in the social sciences. Work in computer science has an analogous interest in understanding what minimal level of rationality can generate still useful outcomes.

In fact, there exist many adaptive procedures which are not Hannan consistent, but in effect trade the guarantee of convergence to a correlated equilibrium for something better, namely the possibility of finding equilibrium states which are not correlated equilibria, but can be preferable to any CE both individually and collectively. The canonical example of such a "nonrational" equilibrium is mutual cooperation in the prisoner's dilemma. The following discussion shows that the RE learning procedure can attain exactly this socially optimal equilibrium.

4.2 the Roth-Erev payoff-matching model

Our presentation of the RE model follows [5], with a few modifications to suit the context of this paper. The RE model uses a stochastic choice rule to play 2-player binary choice repeated games. The RE model has been analyzed for social dilemma games in particular, and so the present discussion will be restricted to social dilemma games in which the choice available to each player is either *cooperate* (C) or *defect* (D).

Let the *effective payoff* $\pi_i(x, y)$ have the value $\pi(x, y) - A$ when $x = i$, and 0 otherwise (this definition is just for convenience of notation in the derivation given below). Now define the *stimulus* $w_i \in [0, 1]$ of action x_t at time t by the quantity

$$w_{i,t}(x_t, y_t) = \frac{\pi_i(x_t, y_t)}{(\pi_m - A)}$$

where $\pi_i(x_t, y_t)$ is the player's effective payoff for action x_t (with the opponent's action being y_t), π_m is the maximum payoff for any pair of choices, and payoffs are evaluated against an *aspiration level* $A \geq 0$, here taken to be a constant. The Roth-Erev model assumes that the probability of an action $i \in \{C, D\}$ at time t is proportional to the *propensity* $q_{i,t}$ of i, defined as $q_{i,t} = \sum_{s=1}^{t-1}(w_{i,s})_+$,[9] The probability of an action i at time t is then

$$p_{i,t} = \frac{q_{i,t}}{q_{i,t} + q_{j,t}} \tag{4.1}$$

where $i, j \in \{C, D\}$ and $j \neq i$.

We now visit the analysis of learning and cooperation in Flache *et al*'s "general reinforcement learning" (GRL) procedure in [5]. The GRL procedure is a parametric generalization of the RE model and the related Bush-Mosteller model, such that both models can be generated from extrema of the parameters, as instances of a general class. The results described in [5] apply to both special cases, so we keep the RE model as a prototype of this general class.

It was found using agent-based computer simulation that, for this class of strategies, there were just two kinds of equilibria in the three social dilemma games examined.

- A *self-reinforcing equilibrium* (SRE) exists when both players converge on the same strategy distribution, such that the probability of the (same) strategy i approaches 1 geometrically quickly for both players.

- A *self-correcting equilibrium* (SCE) exists when any tendency towards an SRE is interruped by an unexpected deviation from the strategy of the SRE, which destabilizes the SRE for both players and tends to knock them both to roughly equal propensities for either cooperation or defection.

An SRE is only possible when the aspiration A is neither too low nor too high. In particular, in the Prisoner's Dilemma, when $A > \pi(C, C)$, the only SRE is mutual defection, as in the classical case. But when $A \leq \pi(C, C)$, not only is the SRE at mutual cooperation possible, but it is the only SRE – mutual defection is no longer an equilibrium. In all three of the social dilemma games investigated, when aspiration is not too high, players are guaranteed

[9]The original RE model uses a "clipping" rule to bound the propensity contribution from a negative outcome to some small positive constant ν. For simplicity, we take $\nu = 0$; Flache and Macy's generalization of RE, discussed below, does away with the clipping rule entirely.

to enter a cooperative SRE through a process called *stochastic collusion*, a kind of random walk out of the SCE into an SRE. Thus, the RE model (and others of its class) offer an adaptive procedure which can escape from socially deficient equilibria.

A drawback of simulation-based analysis of the RE model is that it tends to preclude any analysis that does not depend on the numerical results of simulation. In fact, to our knowledge, very little non-simulation-based analysis of complexity in the RE model exists.

4.3 the RE model as potential descent

The potential-based approach affords an easy analytical treatment of the dynamics of games in the RE model in terms of expected loss, and time complexity to attain convergence to an SRE. To derive this procedure as a potential-descent-based predictor, let $\mathcal{X} = \mathcal{Y} = \{C, D\}$. For any $x_t \in \mathcal{X}$ and $y_t \in \mathcal{Y}$, define the regret $\mathbf{r}_t : \mathcal{X} \times \mathcal{Y} \mapsto \Re^2$ in terms of the stimulus w (defined above), by

$$r_{i,t} = -(w_{i,t}(x_t, y_t))_+$$

As usual, the cumulative regret is defined by $R_{i,t} = \sum_{s=1}^{t} r_{i,s}$. Now define the potential $\Phi(\mathbf{u}) = u_i^2 + u_j^2$, where $i, j \in \{C, D\}$ and $j \neq i$. It is apparent that $-\nabla_i \Phi(\mathbf{R}_{t-1}) = -2R_{i,t-1} = 2q_{i,t}$, and so the predictor

$$p_{i,t} = \frac{-\nabla_i \Phi(\mathbf{R}_{t-1})}{\sum_{k \in \{C,D\}} -\nabla_k \Phi(\mathbf{R}_{t-1})}$$

returns exactly the usual Roth-Erev predictor (4.1) defined earlier.

Before examining the Blackwell condition for this predictor, we note the relationship of the RE model to the regret matching procedure discussed earlier. It is easy to interpret the potential-descent version of RE as a regret-matching procedure, where the regret defined in (4.3) is not the internal regret of Hart and Mas-Colell, but rather a kind of external regret that measures the outcome of a choice relative to the aspiration level A, rather than to the expected outcome of a fictitious best strategy. While RE does not produce Hannan consistent strategies, it does produce strategies which are in some correlated ε-equilibrium. In particular, if $\varepsilon = \pi_m - A$ is the difference between the maximum payoff π_m and the aspiration level, assuming $\pi_m > A$, then any strategy that achieves the aspiration level will be in the correlated ε-equilibrium. The use of the concept of aspiration effectively redefines the equilibrium, so that any payoff $\pi \geq A$ is "good enough". Satisficing behavior thus can attain equilibria that cannot be attained by rational maximizing.

As mentioned above, RE is a special case of a general reinforcement learning model (GRL). GRL can also be specified parametrically to produce the Bush-Mosteller (BM) model, which treats learning as a kind of stimulus-response process. Hart *et al* have shown that their regret-matching procedure can be modified so that no knowledge of what payoffs would have been is needed, but only a "memory" of what average payoffs were obtained for actions actually taken in the past [12]. This modified form of regret matching is thus also

a stimulus-response procedure: actions are taken with a likelihood in proportion to payoffs actually obtained for them in the past. Many other such stimulus-response procedures exist, both Hannan-consistent and otherwise.

Returning to the analysis of the potential-descent RE, the Blackwell condition in this case becomes

$$R_{C,t-1} \cdot r_{C,t} + R_{D,t-1} \cdot r_{D,t} \geq 0$$

Inspecting the expression for the effective payoff, we can distinguish two cases:

- If $\pi \leq A$ or $\pi \geq A$ always, then the Blackwell condition is satisfied, and we can guarantee a regret-minimizing predictor; this corresponds to the existence of an SRE, as described above. However, only when $\pi \geq A$ is mutual cooperation an SRE; otherwise, there may be SREs that are socially suboptimal.

- Otherwise, when the sign of the regret in round t sometimes disagrees with the cumulative regret, we cannot be sure that the predictor's choice will minimize regret. This corresponds to an SCE, where there is no clear advantage to any pure strategy, i.e. any action is roughly as likely to fail as to succeed.

Hence, when the game is caught in an SCE, the RE predictor is of no help; but as players commence a stochastic walk out of the SCE, the predictor becomes more and more likely to lead to an SRE. But how long can we expect to take to attain an SRE?

4.4 analytic upper bounds for convergence to mutual cooperation in the RE model

We can address the convergence question as follows. We will derive an expression bounding the probability of a player's cooperation p_C at time t, which will give the total time $T(\epsilon)$ for p_C to get within ϵ of 1 in an unbroken walk to a mutually cooperative SRE. We then compute the probability of T back-to-back cooperations, and from this derive an upper bound on the number of tries it would take to reach SRE in the worst case. Although we do not do so here, a similar method can be used to estimate a bound for the loss a player suffers before reaching within ϵ of the SRE with probability δ, by counting the expected number of unilateral and mutual defections that occur during SCE.

Assume that each player starts with roughly equal probabilities of cooperating or defecting at time $t = 0$, and assume that in the next step both players will cooperate, taking the first step along a random walk hopefully ending in an SRE. We assume (pessimistically) that any defection will end the random walk, and so $R_{D,t} = 0$ for $t > 0$. Since the probability p_i of an action matches the history of payoffs (rather, of stimuli), the probability of cooperating at step t will be $p_{C,t} = q_{C,t}/Z_t = R_{C,t}/Z_t$, where $Z_t = R_{C,t} + R_{D,0}$ is the normalization. The payoffs for cooperation will increase as long as the walk into SRE continues. Using Cesa-Bianchi's result (3.1) for the polynomial potential, a bound on the size

of $R_{C,t}$ is given by

$$R_{C,t} \leq \sqrt{\sum_{s=1}^{t} |\mathbf{r}_s|^2} \leq \sqrt{t}$$

(so long as the player is approaching SRE). The result on the right-hand side follows because the norm of every $\mathbf{r}_s$ is at most 1, and so the magnitude of the sum is at most t. Hence the bound on the probability of cooperation will be

$$p_{C,t} \leq \frac{\sqrt{t}}{\sqrt{t}+c} = \left(1 + \sqrt{\frac{c}{t}}\right)^{-1}$$

where the constant c is the initial regret of defection $R_{D,0}$. If we define $1-\delta'$ as the threshold of an SRE, where δ' is a positive confidence value ($< 1/2$), then it will take at least $T' = c\left(\frac{1-\delta'}{\delta'}\right)^2$ steps to reach convergence on the SRE. However, at every step, there is a nonzero chance of defecting, and even a single defection may upset the SRE; hence, the chance of *not* defecting for T rounds, which we call d, is only

$$d = \prod_{t=1}^{T'} \left(1 + \sqrt{\frac{c}{t}}\right)^{-1}$$

The chance that neither player defects through round T' is therefore d^2. Any time either player in a walk to SRE defects (which happens with probability $1-d^2$ on each attempt), the game may fall back to an SCE. To ensure lock-in with confidence δ (which we can identify as δ' for convenience), we may have to try at least n times to reach SRE before succeeding, where $nd^2 \geq 1-\delta$. So a walk to SRE may require $n \geq \frac{1-\delta}{d^2}$ attempts before lock-in is reached with probability $1-\delta$. If $\hat{T}$ is the mean number of steps in a (failed) walk to SRE before falling back to SCE, then the total time complexity T of reaching SRE will be at most

$$T = O\left(\hat{T}\frac{1-\delta}{d^2}\right)$$

We have sketched the derivation of a weak, but analytic (although not necessarily closed-form) upper bound on the time complexity of reaching SRE, which is largely independent of the details of the RE model. To our knowledge, no comparable analysis of the time complexity of attaining mutual cooperation in the RE model has been done.

5 discussion and conclusions

We have demonstrated several significant connections between computational learning theory, game theory, and the social sciences. In particular, we have shown that several different approaches to learning and solving games – approaches from both computer science and

from economic game theory – all are types of Hannan-consistent regret-matching algorithms, which can be derived as potential-descent algorithms that minimize regret. Recent work by Hart has underscored the importance of correlated equilibrium as a solution concept; in particular, adaptive heuristics which are *uncoupled*, or which do not depend on preferences but only on observable outcomes, cannot guarantee convergence to Nash equilibria [9]. This constitutes a rigorous demonstration that boundedness of rationality in real behavior is likely to preclude the outcomes expected by classical theory.

Inspection of loss bounds on MW, regret matching, and other algorithms discussed in [3] that use the potential-descent approach shows that all of the Hannan-consistent algorithms, i.e. procedures which drive a measure of average cumulative regret to zero, have a tight upper bound on loss which is $O(\sqrt{1/T})$ per round, or $O(\sqrt{T})$ for T rounds. Optimality results proved by Freund and Schapire for the MW algorithm in [7] suggest that this upper bound may also be the lower bound for adaptive algorithms.

Stochastic learning procedures common in the social sciences may also be analyzed using potential descent and other approaches from computer science, as we have sketched for the Roth-Erev model. This analysis is important, because such stochastic learning procedures can be used to explain equilibria observed in real economic behavior which may be neither Nash equilibria, nor even purely optimal ($\varepsilon = 0$) correlated equilibria. Using the potential descent framework, it can be shown that the regret-matching procedure of Hart and Mas-Colell, which is Hannan-consistent, is qualitatively similar to the Roth-Erev and Bush-Mosteller models (generalized by Flache and Macy as the GRL model) when the loss function is modified to include the concept of aspiration used in the GRL. Inspection of a GRL-type model, such as RE, suggests that an SRE may be interpreted as a correlated ε-equilibrium, which allows for a nonzero average regret, corresponding to the difference between aspiration and the payoff of mutual cooperation in social dilemma games.

All of the adaptive algorithms for playing games in computational learning theory discussed here depended on notions of regret in nonreactive environments – i.e., they effectively assume that playing j rather than i on previous turns would have had no effect on the actions of the other players. This assumption is clearly counterfactual in all but the most trivial environments, and hence there are many environments in which failing to take into account the response of the environment leads to very poor outcomes. Very recent work by de Farias and Megiddo [4] and Cesa-Bianchi and Lugosi[10] has explored new procedures that do take into account reactive environments. The algorithm of de Farias and Megiddo, for example, uses an expert-prediction scheme that alternates between exploring (i.e. learning which weighting of the experts gives the best result) and exploiting the current configuration of experts. This approach is able to find optimal equilibria in games like the Prisoner's Dilemma, but with rigorous bounds on time complexity and loss.

The analysis of repeated social dilemma games from a computational learning perspective complements the importance of social dilemma games in the social sciences. For

[10] In a forthcoming book.

instance, a body of work summarized by Axelrod [1] has shown that strategy equilibria corresponding to quasi-stable mutual cooperation can exist even in the absence of foresight and altruism, contrary to the predictions of classical game theory. Recent work in stochastic game theory, outlined by Goeree and Holt in [8], has shown that stochastic models of adaptive behavior can predict observed economic behavior that defies classical solution concepts (e.g. Nash equilibrium), as well as yielding utility-maximizing outcomes with little or no assumptions about individual rationality. Related efforts in sociology, in particular research by Macy *et al* ([14, 15]) has explored stochastic agent-based models of behavior in social dilemma games that result in outcomes that are "better than rational" in the sense that they can avoid socially deficient equilibrium states, like mutual defection in the Prisoner's Dilemma, that in the classical view should result from rational utility maximization.

In the other direction, results from contemporary stochastic game theory complement research in computational theory, for example by showing unexpected connections between logit equilibrium and the Fokker-Planck equation, and by helping to define new and important applications for theoretical results in computer science.

Finally, we mention what may be the most intriguing implication of the results surveyed in this paper, which is the apparent deep connection between forward-looking (utility-maximizing) and backward-looking (regret-minimizing, adaptive, evolutionary) approaches. In many cases, these two very different classes of approach yield precisely the same outcomes, whereas in certain games including social dilemma games, rationally-bounded adaptive approaches can yield even better outcomes that the outcome predicted by classical (unbounded rationality) theories. This suggests that "doing better" may sometimes be better than "doing best". However, there are clearly cases when greedy improvement strategies will lead to equilibria that are social traps. Quantifying the difference between the two cases may help us solve longstanding problems of distributed and collective action in both artificial and real social systems.

6 acknowledgements

The author gratefully acknowledges the guidance of Rocco Servedio, and helpful pointers from Yoav Freund, Sergiu Hart, and Nicolo Cesa-Bianchi which were instrumental in the development of this paper.

References

[1] Robert Axelrod. *The Evolution of Cooperation.* Basic Books, 1984.

[2] David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1–8), 1956.

[3] Nicolo Cesa-Bianchi and Gabor Lugosi. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51:239–261, 2003.

[4] Daniela Pucci de Farias and Nimrod Megiddo. Combining expert advice in reactive environments. *(submitted for publication)*, 2004.

[5] Andreas Flache and Michael W. Macy. Stochastic collusion and the power law of learning. *Journal of Conflict Resolution*, 46(5):629–653, October 2002.

[6] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Computer and System Sciences*, 55(1):119–139, August 1997.

[7] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, April 1999.

[8] Jacob Goeree and Charles Holt. Stochastic game theory: For playing games, not just for doing theory. *Proc. Nat. Acad. Sci. USA*, 96:10564–10567, September 1999.

[9] Sergiu Hart. Adaptive heuristics. *The Hebrew University of Jerusalem, Center for the Study of Rationality DP-372 (October 2004); Econometrica*, forthcoming.

[10] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, September 2000.

[11] Sergiu Hart and Andreu Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 98(1):26–54, 2001.

[12] Sergiu Hart and Andreu Mas-Colell. *Economic Essays: A Festschrift for Werner Hildenbrand*, chapter : "A reinforcement procedure leading to correlated equilibrium". Springer-Verlag, 2005.

[13] Nick Littlestone and Manfred Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.

[14] Michael W. Macy and Andreas Flache. Learning dynamics in social dilemmas. *Proc. Nat. Acad. Sci. USA*, 99 (suppl. 3):7729–7236, May 14 2002.

[15] Michael W. Macy and John Skvoretz. The evolution of trust and cooperation between strangers: A computational model. *American Sociological Review*, 63:638–660, October 1998.

[16] Robert E. Schapire. The boosting approach to machine learning: An overview. In *Proceedings of the MSRI Workshop on Nonlinear Estimation and Classification*, 2002.

[17] Moshe Tennenholtz. Game theory and artificial intelligence. In M. Fisher M. d'Inverno, M. Luck and C. Preist, editors, *Foundations and Applications of Multi-Agent Systems: UKMAS Workshop 1996-2000. Selected Papers.*, volume 2403 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.