Information Theoretic Connections between Game Theory and Machine Learning

Grant Stafford, Advisor Dash Fryer

April 5, 2013

Contents

1 Introduction

1.1 Motivation

In this paper we explore the mathematical connections between the fields of **game theory** and **machine learning**. Although the fields are ostensibly different, they share deep connections through the abstract field of **information theory**. In particular, both game theoretic and learning processes share dynamic behavior described by a fundamental concept in information theory. These connections are mathematically elegant and provide interesting perspectives on problems in both domains (and others). In application, there is wide applicability of game-theoretic algorithms to diverse learning problems, as illustrated by the derivation of the effective multiplicative weights learning algorithm from matrix games.

1.2 Background

1.2.1 Game Theory

Game theory is typically defined as the study of mathematical models of conflict and cooperation between intelligent rational decision-makers.

Although there are many different models representing games between agents, most share many of the following features with refinements for incomplete information, sequential play, bounded rationality, and more.

In a **game**, each of the **players** has a finite set of **moves** (which may include "do nothing") from which they select one to play at each iteration of the game. At the conclusion of each iteration of play, a function defined by the game converts the set of chosen moves into a set of real valued **payoffs** for each player. As rational agents, each player attempts to maximize their individual payoff.

A game is **zero-sum** if, in each iteration of the game, for any choice of moves by the players, the sum of the payoffs is zero. In considering connections to machine learning, we will first consider some of the simplest games: matrix games.

1.2.2 Machine Learning

Machine learning is concerned with giving computers the ability to learn without being explicitly programmed. Although there are many different subfields of machine learning, most techniques combine ideas from statistics and artificial intelligence.

In connections to game theory we mainly concern ourselves with **supervised learning**: finding a **hypothesis**, a good approximation to some unknown target function f, given training examples $(x, f(x))$ that map inputs to target outputs. Since supervised learning relies on known examples, it implicitly relies on the **inductive learning hypothesis**: "any hypothesis found to approximate the

target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples."

A classic application of supervised learning is to **classification problems**, in which the goal is to correctly classify inputs into categories based on their features. An example of this might be classifying emails into spam and not spam categories based on their text, sender, and other features. Basic classification hypotheses can be formulated by a variety of methods, but techniques that can combine diverse classification hypotheses have often produced better results.

1.2.3 Information Theory

Information theory, developed by Shannon and Weaver [8], is a field of mathematics involved with the quantification of information. Originally developed to study problems in communication, it has since found applications in a vast number of other fields. The abstractions developed in information theory have helped solve problems in data compression, statistical analysis, and modelling of probabilistic phenomena and are central to understanding the connection between game theory and machine learning.

1.3 Related Work

Although not the first to note connections, [1] and [2], authored by the Shapire and Freund, the pioneers of the adaptive boosting (AdaBoost) technique, are some of the earliest papers explaining game theoretic interpretations of a specific learning algorithm. These sources detail game-theoretic derivations of the multiplicative weights algorithm and its relation to boosting techniques prominent in machine learning.

In relation to game theory, [3] by Fryer provides the relevant information theory definitions needed for this paper, mentioned earlier by Shannon [7] and Kullback & Liebler [6]. The proof of the connection between replicator dynamics and Bayesian inference is provided by Harper in [4] and serves as a powerful general link between learning and game dynamics.

Though beyond the scope of this paper, the field of information geometry outlined in [5] has recently produced elegant results that generalize phenomena in machine learning and game theory as well as statistical physics and other fields.

2 The Multiplicative-Weights (MW) Algorithm

2.1 Matrix Games

A **matrix game** is a two-player zero-sum game in which both players (P_1 and P_2) have a constant and fixed number of moves, allowing the payoffs be represented in a single **payoff matrix**. A **payoff matrix** for a matrix game G with m moves for player 1 and n moves for player 2 is the $m \times n$ matrix

$M = [a_{ij}]$ where a_{ij} is the payoff to player 1 if player 1 plays move i and player 2 plays move j.

In such games we typically want to analyze which moves players will choose. On each iteration of the game a player can only choose one move, but over many iterations of a game players can effectively randomize which moves they play. We define a **strategy** for a player with n moves as a probability vector with n entries $a_1, \ldots, a_n$ where a_i denotes the probability that the player selects move i. A **pure strategy** is one in which a single entry is 1 and all others are 0. Strategies with more than one positive entry are called **mixed strategies**.

The **expected payoff** for a game with P_1 strategy P and P_2 strategy Q is $\sum_{i,j} P(i)MQ(j) = P^T MQ$. It can be used to measure the effectiveness of various strategies.

For the remainder of this section, we consider matrix games as defined above, but with two slight modifications. Here the values of the matrix represent losses (rather than payoffs) for the first player and all values in the matrix are normalized to be within the interval $[0, 1]$.

2.1.1 An Example

Consider the matrix game representation of standard Rock, Paper, Scissors

$$M = \begin{array}{c} \\ R \\ P \\ S \end{array} \begin{array}{c} \begin{array}{ccc} R & P & S \end{array} \\ \begin{pmatrix} \frac{1}{2} & 1 & 0 \\ 0 & \frac{1}{2} & 1 \\ 1 & 0 & \frac{1}{2} \end{pmatrix} \end{array}$$

If P_1's strategy is to play only rock, his strategy vector is $P = [1, 0, 0]$ (a pure strategy) and if his strategy is to play each move with equal probability his strategy vector is $P = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ (a mixed strategy). If $P = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ and $Q = [\frac{1}{2}, \frac{1}{2}, 0]$ the expected loss is $\frac{1}{2}$.

2.2 The Minimax Theorem

The classic approach to finding the "solution" to matrix games has applied the famous **minimax theorem** [9]:

Minimax Theorem.

$$\underbrace{\min_P \max_Q M(P, Q)}_{minimax} = \underbrace{\max_Q \min_P M(P, Q)}_{maximin}$$

This theorem says that the minimax strategy payoff is equal to maximin strategy payoff. Intuitively, if players choose strategies to defend against their worst case scenario, there is no advantage to knowing the opponent's move before it is played. In this sense, the minimax and maximin strategies are

optimal and finding them, typically via linear programming, gives the "solution" for a matrix game. We can then define the **value of a game** v as

$$v = \min_P \max_Q M(P, Q) = \max_Q \min_P M(P, Q)$$

The minimax theorem is a powerful result but may not be able to give the optimal strategies in certain circumstances. If M is unknown or too large the minimax approach may be infeasible. Additionally, if the column player is not fully adversarial then the row player may be able to obtain a loss significantly smaller than v.

2.3 The MW Algorithm

2.3.1 Preliminaries

The MW algorithm is an alternative mechanism for selecting strategies. We will initially examine the MW algorithm from a game-theoretic point of view and note results about its effectiveness. We consider a game on a matrix M with row player R and column player C. The matrix is possibly unknown to R and the game is repeated for a number of rounds. On round $t = 1, \ldots, T$:

1. R chooses a mixed strategy P_t.
2. C chooses mixed strategy Q_t (possibly with knowledge of P_t).
3. R is observes the loss $M(i, Q_t)$ for each row i, the loss he would have suffered playing the pure strategy i.
4. R suffers loss $M(P_t, Q_t)$.

The goal of R is to minimize his total loss $\sum_{t=1}^{T} M(P_t, Q_t)$.

2.3.2 The Algorithm

To update his strategies, R maintains non-negative weights on each row of M. Let $w_t(i)$ denote the weight at time t on row i.

The algorithm starts with each weight set to 1. On each round t, R computes a new mixed strategy P_t by normalizing the weights:

$$P_t(i) = \frac{w_t(i)}{\sum_i w_t(i)}$$

Then once he has observed $M(i, Q_t)$ for each i, R updates the weights by a simple multiplicative rule:

$$w_{t+1}(i) = w_t(i) * \beta^{M(i,Q_t)}$$

Where $\beta \in [0, 1)$ is a parameter of the algorithm, the adjustment rate.

2.3.3 Performance

Although it is not immediately obvious, the MW algorithm has very desirable performance characteristics. It can potentially do much better than the minimax strategy and even in the worst case it will do approximately as well as the minimax strategy. The results below formalize the performance of the algorithm.

Theorem 1. *For any matrix M with n rows and entries in $[0,1]$ and for any sequence of mixed strategies $Q_1, \ldots, Q_T$ played by C, the sequence of mixed strategies $P_1, \ldots, P_T$ produced by algorithm MW with parameter $\beta \in [0,1)$ satisfy:*

$$\sum_{t=1}^{T} M(P_t, Q_t) \leq a_\beta \min_P \sum_{t=1}^{T} M(P, Q_t) + c_\beta \ln n$$

where

$$a_\beta = \frac{\ln(1/\beta)}{1-\beta} \qquad c_\beta = \frac{1}{1-\beta}$$

Proof. The proof from [1] is included in appendix A for completeness. □

Corollary 1. *Under the conditions of Theorem 1 and with β set to*

$$\frac{1}{1+\sqrt{\frac{2\ln n}{T}}}$$

the average per-trial loss suffered by R is

$$\underbrace{\frac{1}{T}\sum_{t=1}^{T} M(P_t, Q_t)}_{\text{actual}} \leq \underbrace{\min_P \frac{1}{T}\sum_{t=1}^{T} M(P, Q_t)}_{\text{best}} + \underbrace{\Delta_T}_{\text{regret}}$$

where the regret $\Delta_T = \sqrt{\frac{2\ln n}{T}} + \frac{\ln n}{T} = O(\sqrt{\frac{lnn}{T}})$.

R's regret is the additional loss suffered beyond the loss suffered by the minimax strategy. Since R's average per-trial regret is logarithmic in the number of rows and independent of the number of columns, the average performance of the MW algorithm will never be more than a relatively small amount worse than minimax. In addition, observe that as $T \to \infty$, $\Delta_T \to 0$. That is, after enough rounds R's average per-trial regret goes to zero.

2.4 MW Applied to On-Line Prediction

2.4.1 On-Line Prediction

To illustrate the use of the MW algorithm in learning we consider an **on-line prediction** scenario, in which a learner predicts the categories of a sequence

of examples presented one at a time by the environment while attempting to minimize prediction errors.

Formally, let X be a finite set of examples, and let H be a finite set of hypotheses $h : X \to \{0, 1\}$. Let $c : X \to \{0, 1\}$ be an unknown target concept, not necessarily in H.

On-line learning takes place in a sequence of rounds. On round $t \in \{1, \ldots, T\}$:

1. The learner observes an example $x_t \in X$.
2. The learner makes a randomized prediction $\hat{y}_t \in \{0, 1\}$ of the label associated with x_t.
3. The learner observes the correct label $c(x_t)$.

The goal of the learner is to minimize the expected number of mistakes that he makes relative to the best hypothesis in the space H.

2.4.2 Applying the MW Algorithm

To apply the MW algorithm we let the learner take the role of R, the row player, and let the environment take the role of C, the column player.

The environment's choice of a column corresponds to a choice of an instance $x \in X$ that is presented to a learner on a given iteration. The learner's choice of row corresponds to choosing a specific hypothesis $h \in H$ and predicting the label $h(x)$. A mixed strategy for the learner corresponds to making a random choice of a hypothesis with which to predict.

Because we are modelling a classification problem the environment only plays pure strategies. So we have a matrix game with $|H|$ rows representing the hypotheses $h \in H$ and $|X|$ columns representing the examples $x \in X$. We define the matrix entries to have value 1 if and only if h disagrees with the target c on instance x. So we have the **mistake matrix**

$$M(h, x) = \begin{cases} 1 & \text{if } h(x) \neq c(x) \\ 0 & otherwise \end{cases}$$

We apply the MW algorithm to the matrix M. On round t, given instance x_t, MW provides a distribution P_t over the rows of M. We randomly select $h_t \in H$ according to P_t, and predict $\hat{y}_t = h_t(x_t)$. Next, given $c(x_t)$ we compute $M(h, x_t)$ for each $h \in H$ and update the weights. Note that β, the adjustment rate parameter, can be considered to be the learning rate parameter in this context.

2.4.3 Performance

Note that

$$\begin{aligned} M(P_t, x_t) &= \sum_{h \in H} P_t(h) M(h, x_t) \\ &= \Pr_{h \sim P_t}[h(x_t) \neq c(x_t)] \\ &= \Pr[\hat{y}_t \neq c(x_t)] \end{aligned}$$

Combining this with Corollary 2, the expected number of mistakes made by the learner is

$$\sum_{t=1}^{T} M(P_t, x_t) \leq \min_{h \in H} \sum_{t=1}^{T} M(h, x_t) + O(\sqrt{T \ln |H|})$$

which means that the number of mistakes made by the learner cannot exceed the number of mistakes made by the best hypothesis in H by more than $O(\sqrt{T \ln |H|})$.

2.5 An Example

The following toy example will illustrate the MW algorithm. We assume we are in a on-line learning scenario trying to correctly classify emails into spam and non-spam categories. Our set of instances X contains the following elements, emails represented as (sender, subject, message) vectors:

- b_1 : (Friend, Cool Video, "Check out this YouTube video: *link*")
- b_2 : (College, College Newsletter, "College Announcements ...")
- b_3 : (Money Inc., EZ MONEY, "CALL 123-456-7890")
- b_4 : (Government.com, SSN Request, "Renew your SSN: *link*")
- b_5 : (News.com, Breaking News!, "Watch this video!: *link*")

Our target concept function c will yield 1 on spam messages and 0 on non-spam messages, which gives the following:

- $c(b_1) = 0$
- $c(b_2) = 0$
- $c(b_3) = 1$
- $c(b_4) = 1$
- $c(b_5) = 1$

Our hypothesis space H contains the following elements, functions that attempt to yield 1 on spam messages and 0 on non-spam messages:

- $a_1(x) = 1$ if x's sender not in contacts list, 0 otherwise
- $a_2(x) = 1$ if x''s subject in all caps, 0 otherwise
- $a_3(x) = 1$ if x contains links, 0 otherwise
- $a_4(x) = 1$ if $a_1(x) = a_3(x) = 1$, 0 otherwise

The above results give the following mistake matrix M, where 1 entries indicate that the hypothesis of the row disagrees with the target concept in classifying the email of the column:

$$M = \begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{array} \begin{array}{c} \begin{array}{ccccc} b_1 & b_2 & b_3 & b_4 & b_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

At the beginning of the first round $t = 1$ we have a weight of 1 on each of our hypotheses: $w_1(a_1) = w_1(a_2) = w_1(a_3) = w_1(a_4) = 1$. This means that our initial strategy will be to use each hypothesis with equal probability

$$P_1 = \left(\frac{w_1(a_1)}{\sum_i w_1(i)}, \frac{w_1(a_2)}{\sum_i w_1(i)}, \frac{w_1(a_3)}{\sum_i w_1(i)}, \frac{w_1(a_4)}{\sum_i w_1(i)} \right) = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right)$$

Now let's assume that $Q_t = (0, 0, 0, 1, 0)$ and b_4 is presented as the first example. So $x_1 = b_4$. Say that using P_1 we randomly select a_1 as our hypothesis for this round. So $h_1 = a_1$. We will predict that x_1 is spam because its sender is not in our contacts list: $\hat{y}_1 = h_1(x_1) = 1$. It is revealed to actually be spam ($c(x_1) = 1 = \hat{y}_1$) so we suffer no mistake this round.

Now we observe how each of the other hypotheses would have done on this example. Think of this as us considering the counterfactual situations in which we used a different hypothesis on this example. We see that $M(a_2, x_1) = 1$, which means that a_2 would have made a mistake on this example and may not be a good hypothesis.

At the conclusion of the round we update the weights for each of our hypotheses according to the following formula.

$$w_2(i) = w_1(i) * \beta^{M(i,Q_t)} \text{ for some } \beta \in [0, 1)$$

$M(a_1, Q_1) = M(a_3, Q_1) = M(a_4, Q_1) = 0$, so we can see that

$$w_2(a_1) = w_2(a_3) = w_2(a_4) = 1 * \beta^0 = 1$$

a_1, a_2 and a_4 retain their weight of 1 for the next round. This makes sense, we will never lower the weight on a hypothesis until it errs on an example. $M(a_2, Q_1) = 1$ so we have

$$w_2(a_2) = 1 * \beta^1 = \beta$$

Because $\beta \in [0, 1)$, the weight on a_2 will be decreased. The size of β parameter determines how quickly we adjust our strategy. Values of β close to 1 are conservative and change our strategy slowly, while values of β close to 0 make aggressive changes. Assume for this example we let $\beta = \frac{1}{2}$, so that $w_2(a_2) = \frac{1}{2}$.

At the start of round $t = 2$ we will have a new strategy $P_2 = (\frac{2}{7}, \frac{2}{7}, \frac{2}{7}, \frac{1}{7})$. We see that we are less likely to use hypotheses which would have performed poorly in the past and are more likely to use hypotheses which have not made mistakes. If we continue to use the MW algorithm as we observe more examples we will eventually converge to the optimal strategy $P^* = (0, 0, 0, 1)$, which is to always use hypothesis a_4. The discussion above tells us that relative to always using a_4 from the start, the MW algorithm will make only a reasonably small number of additional mistakes. It is an effective algorithm for on-line learning.

2.6 Relation to Bayesian Inference

Although the MW algorithm is a concrete example of a useful game theory and machine learning connection, its relation to more fundamental mathematical connections between games and learning may be unclear. As it turns out, under the log loss function the MW algorithm with β set to $\frac{1}{e}$ is equivalent to the Bayes prediction rule, where the generated distributions over the rows are equal to the Bayesian posterior distributions [2]. The widespread appearance of Bayesian inference in both machine learning and game theory makes this a powerful connection, as we will see in the next section.

3 Evolutionary Game Theory and Inference

3.1 Information Theory

Before discussing Bayesian inference and evolutionary game theory some key concepts in information theory must be defined. First is **entropy** [7], denoted $H(P)$, a measure of the average uncertainty in a random variable P. It can be interpreted as the average number of bits needed to encode a message drawn i.i.d from P. For example, a random variable that has a uniform distribution over 32 outcomes would need a label that takes on 32 different values so we have that the entropy of this random variable is

$$H(X) = -\sum_{i=1}^{32} p(i) \log_2 p(i) = -\sum_{i=1}^{32} \tfrac{1}{32} \log_2 \tfrac{1}{32} = \log_2 32 = 5 \text{ bits}$$

The other concept from information theory that will be critical for this paper is the **Kullback-Leibler divergence** [6] (sometimes called **information gain**, **information divergence**, or **relative entropy**), a measure of information gain from one state to another. It is an average measure of the additional bits needed to store y given a code optimized to store x. It is defined as

$$\begin{aligned} D_{KL}(x||y) &= \sum_i x_i \log \frac{x_i}{y_i} \\ &= \sum_i x_i \log x_i - \sum_i x_i \log y_i \\ &= H(x) - H(x,y) \end{aligned}$$

Where $H(x,y)$ is the **cross entropy** of x and y, the average number of bits needed to identify an event from a set of possibilities, if coding scheme is used based on y rather than x.

Observe that minimizing D_{KL} is equivalent to minimizing the cross entropy. Intuitively, minimizing D_{KL} with respect to y is trying to find the best distribution to approximate the 'true' distribution x [3].

3.2 Bayesian Inference

Bayesian inference is an inference system often used in machine learning. It features prominently in the scientific context, as the scientific method is essentially an application of Bayesian inference. As the name implies, Bayesian inference relies on the well-known Bayes' Theorem [4]:

$$P(H_i|E) = \frac{P(E|H_i)P(H_i)}{P(E)} \text{ for } i = 1, 2, \ldots, n$$

where:

- The events $H_1, H_2, \ldots, H_n$ constitute the entire state space:

  $$\sum_{i=1}^{n} P(H_i) = 1$$

  In the context of the scientific method, the events $H_1, H_2, \ldots, H_n$ are hypotheses which cover all possibilities.

- $P(H_i)$ is called the prior probability of H_i.

  In the context of the scientific method, $P(H_i)$ represents the likelihood of the hypotheses under available background knowledge.

- E represents the event of encountering new evidence and

  $$P(E) = \sum_{i=1}^{n} P(E|H_i)P(H_i)$$

  is the marginal probability of E. $P(E)$ appears in the denominator of Bayes' rule to normalize the equation.

  In the context of the scientific method, E represents the result of an experiment and $P(E)$ serves to normalize the results.

- $P(H_i|E)$ is called the posterior probability of H_i given the evidence E.

  In the scientific context, $P(H_i|E)$, the likelihood of our hypotheses given the results from our experiments, is the what experimenters desire to determine. Bayes' Theorem presents how results from experiments can be used to find this.

Given a sequence of observed evidence $E_1, E_2, \ldots$ the probabilities of the events $H_1, H_2, \ldots, H_n$ can be updated in a dynamic process. At each step the Kullback-Liebler divergence $D_{KL}(P(H|E)||P(H))$ measures the information gained from moving from the prior distribution to the posterior distribution. If we consider H^* to be the true distribution for H, then after each iteration of the process $D_{KL}(P(H^*)||P(H|E))$ represents the **potential information** of the system. The potential information of the system will decrease on each iteration until the prior distribution and the posterior distribution match, at which time the potential information is zero.

In the context of the scientific method, repeated experiments form the dynamic process and the KL divergence measures how much was learned about the hypotheses from the results of each experiment. Potential information represents how much still stands to be learned about the phenomenon under study and naturally decreases as theories more accurately reflect reality.

3.3 Evolutionary Game Theory

Evolutionary game theory applies concepts from game theory to model biological evolution. Key to these models is the **replicator dynamic**, a dynamic for describing how the entities in a system, say phenotypes such as fur color, propagate themselves over time. The **discrete replicator dynamic** looks at the change in a population over discrete generations and is defined as

$$x_i' = \frac{x_i f_i(x)}{\bar{f}(x)} \text{ for } i = 1, 2, \ldots, n$$

where:

- x_i is the proportion of the population of the ith type and $x = (x_1, \ldots, x_n)$ is the population distribution. All possible states of the population can be described by the types i.e. $\sum_i x_i = 1$.
- $f = (f_1, \ldots, f_n)$ is the fitness landscape and each function $f_i(x)$ is the fitness of type i dependent on population distribution x.
- $\bar{f}(x) = \sum_{i=1}^{n} x_i f_i(x)$ is the average fitness.
- x_i' is the frequency of type i in the next generation of the population.

Upon each generation the types are adjusted in proportion to their fitness relative to the average population fitness. Over time the population will adjust

to fitness landscape, increasing the proportion of some types at the expense of others. The stable fixed points of the replicator dynamic are in fact Nash equilibria, which generalizes von Neumann's minimax theorem [10]. A specific subset of these fixed points, evolutionary stable states, are of particular interest.

An **evolutionary stable state (ESS)** of the replicator dynamic is a population distribution that is robust to invasion by mutant types. Formally, a distribution $\hat{x}$ is an ESS of the replicator dynamic if $\hat{x} \cdot f(x) > x \cdot f(x)$ in some neighbohood of $\hat{x}$. This means that evolutionary stable states are asymptotically stable points of the replicator dynamic, better replies than all neighboring strategies. Note that an ESS may not always exist, such as in an evolutionary version of the standard rock, paper, scissors game.

Theorem 2. *Suppose that the fitness landscape is strictly positive, that is $f_i(x) > 0$ for all i and x. If the population distribution unfolds according to the discrete replicator dynamic then $\hat{x}$ is an interior ESS if and only if the potential information is decreasing along iterations of the dynamic.*

Proof. Reproduced from [4]

First note that the ESS condition can be stated as

$$\frac{\hat{x} \cdot f(x)}{x \cdot f(x)} > 1$$

for all x in a neighborhood of $\hat{x}$ (since we've assumed that the fitness landscape is strictly positive).

Consider the difference in potential information of two successive states:

$$P = D_{KL}(\hat{x}||x') - D_{KL}(\hat{x}||x)$$

Assume that x is in the ESS neighborhood of $\hat{x}$. Then,

$$\begin{aligned}
P &= \sum_i \hat{x}_i \log \hat{x}_i - \sum_i \hat{x}_i \log x'_i - \left(\sum_i \hat{x}_i \log \hat{x}_i - \sum_i \hat{x}_i \log x_i \right) \\
&= \sum_i \hat{x}_i \log x_i - \sum_i \hat{x}_i \log x'_i \\
&= \sum_i \hat{x}_i \log x_i - \sum_i \hat{x}_i \log \left(x_i \frac{f_i(x)}{\bar{f}(x)} \right) \\
&= -\sum_i \hat{x}_i \log \left(\frac{f_i(x)}{\bar{f}(x)} \right) \\
&\leq -\log \left(\sum_i \hat{x}_i \frac{f_i(x)}{\bar{f}(x)} \right) = -\log \left(\frac{\hat{x} \cdot f(x)}{x \cdot f(x)} \right) < 0
\end{aligned}$$

Where the log was moved outside the sum using Jensen's inequality. □

3.4 Relation of the Replicator Dynamic to Bayesian Inference

We see that both Bayesian inference and the replicator dynamic are dynamical systems governed by decreasing potential information. This is no coincidence. Mathematically, Bayesian inference is a special case of the discrete replicator dynamic; the posterior probability of any given hypothesis will not depend explicitly on the probability of other hypotheses. The table below gives the correspondence between the two concepts.

Formal Analogies

Bayesian Inference	**Discrete Replicator**
Prior Distribution $(P(H_1), \ldots, P(H_n))$	Population state $x = (x_1, \ldots, x_n)$
New Evidence $P(E\|H_i)$	Fitness landscape $f_i(x)$
Normalization $P(E)$	Mean fitness $\bar{f}(x)$
Posterior distribution $P(H_1\|E), \ldots, P(H_n\|E)$	Population state $x' = (x'_1, \ldots, x'_n)$
True Distribution H^*	Evolutionary Stable State $\hat{x}$

3.5 Kullback-Liebler Divergence as a Lyapunov Function for the Replicator Dynamic

The importance of potential information extends beyond discrete systems. It also underlies the dynamics of a continuous version of the replicator dynamic:

$$\dot{x}_i = x_i(f_i(x) - \bar{f}(x))$$

Recall that a **Lyapunov function** V is a continuous function such that $V(0) = 0$ and $V(x) > 0 \; \forall x \in U \setminus \{0\}$ for some neighborhood region U around $x = 0$.

Given some $x^* = 0$ as the equilibrium of the autonomous system $\dot{x} = f(x)$:

- If $\dot{V}(x) \leq 0 \; \forall x \in U \setminus \{0\}$ then the equilibrium is **stable**.
- If $\dot{V}(x) < 0 \; \forall x \in U \setminus \{0\}$ then the equilibrium is **locally asymptotically stable**.
- If $\dot{V}(x) < 0 \; \forall x \in \mathbb{R}^n \setminus \{0\}$ and $||x|| \to \infty \Rightarrow V(x) \to \infty$ the equilibrium is **globally asymptotically stable**.

We can think of a Lyapunov function as describing the "potential energy" of a system. It takes positive values everywhere except the points of equilibrium and is non-increasing along every path that the system can take. It allows us to analyze the stability of points without knowing the exact solution to the system.

An important fact about the Kullback-Leibler divergence related to the concept above is **Gibbs' inequality** which states:

$$D_{KL}(x||y) \geq 0 \text{ with equality if and only if } x = y$$

It follows that $D_{KL}(x^*||y)$ is always a local Lyapunov function for x^*. We use this fact in the following theorem.

Theorem 3. *The state $\hat{x}$ is an interior ESS for the replicator dynamic if and only if $D_{KL}(\hat{x}||x)$ is a local Lyapunov function.*

Proof. Reproduced from [4]

Let $V(x) = D_{KL}(\hat{x}||x) = \sum_i \hat{x}_i \log \hat{x}_i - \sum_i \hat{x}_i \log x_i$

By the discussion above it is a Lyapunov function for $\hat{x}$.

$$\begin{aligned}
\dot{V}(x) &= -\sum_i \hat{x}_i \frac{\dot{x}_i}{x_i} = -\sum_i \hat{x}_i (f_i(x) - \bar{f}(x)) \\
&= -\sum_i \hat{x}_i f_i(x) + \sum_i \hat{x}_i \bar{f}(x) = -\sum_i \hat{x}_i f_i(x) + \left(\sum_i \hat{x}_i\right) \bar{f}(x) \\
&= -\sum_i \hat{x}_i f_i(x) + \bar{f}(x) = -(\hat{x} \cdot f(x) - x \cdot f(x)) < 0
\end{aligned}$$

The last inequality holds if and only if $\hat{x}$ is an ESS. □

$D_{KL}(\hat{x}||x)$ is the potential information of the system and as proved above it is continually decreasing and becomes minimized as the system converges. In essence, the population 'learns' how best to survive in its environment, stabilizing once ecological niches are filled.

Similarly, one can consider scientific hypotheses in a 'Darwinian competition' to match our observations of the world, so that only the 'fittest' ideas survive. The learning-evolution connection has seen practical application in the evolutionary algorithms of artificial intelligence, which typically mimic evolutionary dynamics in an attempt to determine optimal parameters for complex models.

3.6 Other Fields

Information geometry is a branch of mathematics that combines differential geometry and probability theory. In light of the above discussion, it is unsurprising that the replicator equation is known as the natural gradient, and is closely linked to the fundamental information metrics of information geometry [4].

Statistical physics has also been shown to have mathematical connections with game theory. Gibbs entropy from statistical thermodynamics is equivalent to the information-theoretic entropy used in this paper multiplied by Boltzman's constant, so this result was not entirely unexpected. Interestingly, some of the same mathematical issues game theorists have faced in incorporating bounded rationality into their models have also confronted physicists [11].

3.7 Conclusion

We have shown here how game theory and machine learning, though ostensibly different, share the same fundamental mathematical structure. The dynamics of systems in both fields can be cast in terms of potential information defined

by the abstract field of information theory. The MW algorithm provides a concrete example of how a game playing strategy can be adapted to form an effective learning process, and proofs regarding the dynamics of Bayesian inference and the replicator dynamic demonstrate that generalized learning processes share the structure of evolutionary games which play out over time. Information geometry further generalizes the connections seen here. Statistical physics among other fields also shares fundamental dynamics based on information. Both mathematical insight and practical application benefit from recognizing the deep connections between fields such as game theory and machine learning.

References

[1] Y. Freund and R.E. Schapire, *Game theory, on-line prediction and boosting*, Proceedings of the ninth annual conference on Computational learning theory, ACM, 1996, pp. 325–332.

[2] ———, *Adaptive game playing using multiplicative weights*, Games and Economic Behavior **29** (1999), no. 1, 79–103.

[3] D.E.A. Fryer, *The kullback-liebler divergence as a lyapunov function for incentive based game dynamics*, arXiv preprint arXiv:1207.0036 (2012).

[4] M. Harper, *The replicator equation as an inference dynamic*, arXiv preprint arXiv:0911.1763 (2009).

[5] ———, *Escort evolutionary game theory*, Physica D: Nonlinear Phenomena **240** (2011), no. 18, 1411–1415.

[6] Solomon Kullback and Richard A Leibler, *On information and sufficiency*, The Annals of Mathematical Statistics **22** (1951), no. 1, 79–86.

[7] C.E. Shannon, *A mathematical theory of communication*, ACM SIGMOBILE Mobile Computing and Communications Review **5** (2001), no. 1, 3–55.

[8] C.E. Shannon and W. Weaver, *The mathematical theory of communication (urbana, il)*, University of Illinois Press **184** (1949), 1032–54.

[9] John Von Neumann and Oskar Morgenstern, *Theory of games and economic behavior (commemorative edition)*, Princeton university press, 2007.

[10] Jörgen W Weibull, *Evolutionary game theory*, The MIT press, 1995.

[11] David Wolpert, *Information theory the bridge connecting bounded rational game theory and statistical physics*, Complex Engineered Systems (2006), 262–290.

A Proof of Theorem 1

Reproduced from [1].

For $t = 1, \ldots, T$, we have that

$$\begin{aligned}
\sum_{i=1}^{n} w_{t+1}(i) &= \sum_{i=1}^{n} w_t(i) \cdot \beta^{M(i,Q_t)} && \text{(by definition of } w_{t+1}(i)) \\
&\leq \sum_{i=1}^{n} w_t(i) \cdot (1 - (1-\beta) M(i, Q_t)) && (\beta > 0,\ x \in [0,1]) \\
&= \left(\sum_{i=1}^{n} w_t(i) \right) \cdot (1 - (1-\beta) M(P_t, Q_t)) && \text{(by definition of } P_t)
\end{aligned}$$

Unwrapping the recurrence and recalling that $w_1(i) = 1$ we obtain

$$\sum_{i=1}^{n} w_{T+1}(i) \leq n \cdot \prod_{t=1}^{T} (1 - (1-\beta) M(P_t, Q_t))$$

Next, note that, for any j,

$$\sum_{i=1}^{n} w_{T+1}(i) \geq w_{T+1}(j) = \beta^{\sum_{t=1}^{T} M(j,Q_t)}$$

Combining with the preceding equation and taking logs gives

$$\begin{aligned}
(\ln \beta) \sum_{t=1}^{T} M(j, Q_t) &\leq \ln n + \sum_{t=1}^{T} \ln(1 - (1-\beta) M(P_t, Q_t)) \\
&\leq \ln n - (1-\beta) \sum_{t=1}^{T} M(P_t, Q_t) \quad (\forall x < 1,\ \ln(1-x) \leq -x)
\end{aligned}$$

Rearranging terms, and noting that this expression hold for any j gives

$$\sum_{t=1}^{T} M(P_t, Q_t) \leq a_\beta \min_j \sum_{t=1}^{T} M(j, Q_t) + c_\beta \ln n$$

Since the minimum (over mixed strategies P) in the bound of the theorem must be achieved by a pure strategy j, this implies the theorem.