Information Theoretic Connections between Game Theory and Machine Learning

Grant Stafford, Advisor Dash Fryer

April 26, 2013

Motivation

"If one has really technically penetrated a subject, things that previously seemed in complete contrast, might be purely mathematical transformations of each other." -John von Neumann

Why study connections between game theory and machine learning?

Developing Applications

Mathematical Elegance

Game Theory

"If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is." -John von Neumann

Game Theory

Game theory studies mathematical models of conflict and cooperation between intelligent rational decision-makers.

Machine Learning

"Experience is the teacher of all things." -Julius Caesar

Machine Learning

Machine learning is concerned with giving computers the ability to "learn" (improve at a task with experience) without being explicitly programmed.

Information Theory

"Information is the resolution of uncertainty."-Claude Shannon

Information Theory

Information theory is a field of mathematics involved with the quantification of information.

Applications

- Digital communications
- Data compression
- Statistical analysis
- Modelling of probabilistic phenomena
- Many others

Matrix Games by Example

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Matrix game

A **matrix game** is a two-player zero-sum game in which every player has a constant and fixed number of moves, allowing the payoffs be represented in a single **payoff matrix**.
We let the entries of the matrix represent normalized losses (rather than payoffs) for the first player.

Matrix Games by Example

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Payoffs

If player 1 chooses move h_2 and player 2 chooses move x_4 then player 1 loses 1 and player 2 gains 1.

Matrix Games by Example

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$\mathbf{P = [0, 0, 0, 1]}$

$\mathbf{Q = [\frac{1}{4}, \frac{1}{2}, 0, \frac{1}{4}, 0]}$

Strategies

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.

Matrix Games by Example

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$\mathbf{P = [0, 0, 0, 1]}$

$Q = [\frac{1}{4}, \frac{1}{2}, 0, \frac{1}{4}, 0]$

Pure Strategy

A **pure strategy** is one in which a single entry is 1 and all others are 0.

Matrix Games by Example

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$P = [0, 0, 0, 1]$

$\mathbf{Q = [\frac{1}{4}, \frac{1}{2}, 0, \frac{1}{4}, 0]}$

Mixed Strategy

Strategies with more than one positive entry are called **mixed strategies**.

Matrix Games by Example

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$P = [0, 0, 0, 1]$

$Q = [\frac{1}{4}, \frac{1}{2}, 0, \frac{1}{4}, 0]$

$\mathbf{\frac{1}{4} * 0 + \frac{1}{2} * 0 + \frac{1}{4} * 0 = 0}$

Expected Payoff

The **expected payoff** for a game with P_1 strategy P and P_2 strategy Q is $\sum_{i,j} P(i)MQ(j) = P^T MQ$.

Matrix Games by Example

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

P = [0, 0, 0, 1] - minimax strategy

Q = [$\frac{1}{4}$, $\frac{1}{2}$, 0, $\frac{1}{4}$, 0] - maximin strategy

$\frac{1}{4} * 0 + \frac{1}{2} * 0 + \frac{1}{4} * 0 = 0$ - value of the game

Minimax Theorem (Von Neumann, 1928)

$$\underbrace{\min_P \max_Q M(P,Q)}_{minimax} = \underbrace{\max_Q \min_P M(P,Q)}_{maximin} = v = \text{value of the game}$$

On-line Learning: Applying the MW Algorithm

"A learning experience is one of those things that says, 'You know that thing you just did?' Don't do that.'" -Douglas Adams

$$\mathbf{M} = \begin{array}{c} \\ \mathbf{h_1} \\ \mathbf{h_2} \\ \mathbf{h_3} \\ \mathbf{h_4} \end{array} \begin{array}{c} \begin{array}{ccccc} \mathbf{x_1} & \mathbf{x_2} & \mathbf{x_3} & \mathbf{x_4} & \mathbf{x_5} \end{array} \\ \begin{pmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \end{array}$$

Applying the MW Algorithm

Define the payoff matrix M to be the $|H| \times |X|$ **mistake matrix** whose entries are 1 if and only if h disagrees with the target c on instance x.
The row player is the learner and the column player is the environment, choosing hypotheses and instances respectively.

On-line Learning: Example

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

1. $h_1 = 1$ if message sender not in contacts
2. $h_2 = 1$ if message subject in all caps
3. $h_3 = 1$ if message contains untrusted links
4. $h_4 = 1$ if $h_1 = h_3 = 1$

1. x_1 : <Friend, Cool Video, "Check out this YouTube video: *link*">
2. x_2 : <Chirps, Chirps 12/7, "Chirps for December 7th ..."">
3. x_3 : <Money Inc., EZ MONEY, "CALL 123-456-7890">
4. x_4 : <Government.com, SSN Request, "Renew your SSN: *link*">
5. x_5 : <News.com, Breaking News!, "Watch this video: *link*>

On-line Learning: Preliminaries

On-line Learning

An **on-line learning** scenario is an iterated scenario in which the learner is given examples one at a time, attempts to classify them, and is then presented with the true classification.
The goal is to minimize prediction errors relative to the best hypothesis available.

On-line Learning Scenario

On round $t = 1, \ldots, T$:

1. The learner observes an instance x_t from X, the set of instances.
2. The learner makes a randomized prediction $\hat{y}_t \in \{0, 1\}$ of the label associated with x_t, based on one of the hypotheses in H, the set of hypotheses.
3. The learner observes the correct label $c(x_t)$, given by the target concept.

MW Algorithm

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & \ldots? \end{array} \\ \begin{pmatrix} ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \end{pmatrix} \end{array}$$

$\mathbf{w_1 = [1, 1, 1, 1]}$

MW Algorithm

To update his strategies, *learner* 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.

MW Algorithm

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & \ldots ? \end{array} \\ \begin{pmatrix} ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \end{pmatrix} \end{array}$$

$w_1 = [1, 1, 1, 1]$

$\mathbf{P_1 = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]}$

MW Algorithm

On each round t, *learner* computes a new mixed strategy P_t by normalizing the weights:

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

MW Algorithm

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & \ldots? \end{array} \\ \begin{pmatrix} ? & ? & ? & \mathbf{0} & ? \\ ? & ? & ? & \mathbf{1} & ? \\ ? & ? & ? & \mathbf{0} & ? \\ ? & ? & ? & \mathbf{0} & ? \end{pmatrix} \end{array}$$

$P_1 = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$, $Q_1 = [0, 0, 0, 1, 0]$, $\hat{y}_1 = h_1(x_4) = 1$, $c(x_4) = 1$

$\mathbf{w_2 = [1, \frac{1}{2}, 1, 1]}$ if $\beta = \frac{1}{2}$

MW Algorithm

Learner chooses h_t according to P_t, predicts $h_t(x_t)$ and observes $c(x_t)$. Loss is suffered and for each i, the weights are updated 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 learning rate.

MW Algorithm

$$M = \begin{array}{c} \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & \ldots ? \end{array} \\ \begin{pmatrix} ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \end{pmatrix} \end{array}$$

$w_2 = [1, \frac{1}{2}, 1, 1]$ if $\beta = \frac{1}{2}$

$\mathbf{P_2 = [\frac{2}{7}, \frac{1}{7}, \frac{2}{7}, \frac{2}{7}]}$

MW Algorithm

The new weights are used in the next round and the algorithm repeats.

MW Algorithm: Performance

Corollary 2 (Freund & Schapire, 1996)

For any matrix M with n rows and entries in $[0, 1]$ and with β set to

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

the average per-trial error suffered by *learner* over T rounds 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 $\Delta_T = \sqrt{\frac{2\ln n}{T}} + \frac{\ln n}{T} = O(\sqrt{\frac{\ln n}{T}})$.

Information Theory

"You should call it entropy, ... no one really knows what entropy really is, so in a debate you will always have the advantage." -John von Neumann

Entropy (Shannon & Weaver, 1948)

Entropy, denoted $H(X)$, is a measure of the average uncertainty in a random variable X.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

It can be interpreted as the average number of bits needed to encode a message drawn i.i.d from X.
Entropy depends on the number of possible outcomes and the probability of those outcomes.

Information Theory

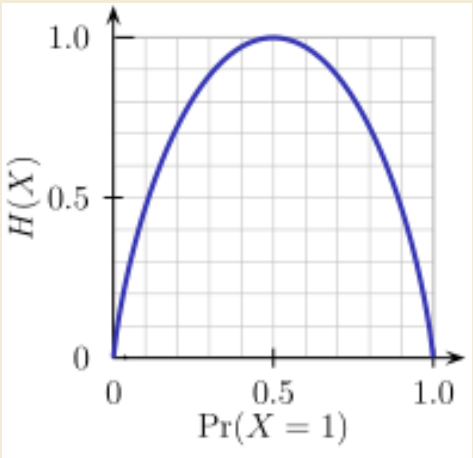
Entropy Example

Consider a coin that lands heads with probability p and tails with probability $1 - p$.

If $p = \frac{1}{2}$ then $H(X) = -(\frac{1}{2} \lg \frac{1}{2} + \frac{1}{2} \lg \frac{1}{2}) = 1$ bit

If $p = \frac{1}{4}$ then $H(X) = -(\frac{1}{4} \lg \frac{1}{4} + \frac{3}{4} \lg \frac{3}{4}) = 0.811\ldots$ bits

If $p = 0$ then $H(X) = -(1 \lg 1) = 0$ bits



Information Theory

Kullback-Liebler Divergence (Kullback & Liebler, 1951)

The **Kullback-Liebler divergence** (sometimes called **information gain**, **information divergence**, or **relative entropy**), is 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.

Bayesian Inference

"Inductive inference is the only process known to us by which essentially new knowledge comes into the world." -R. A. Fisher

Bayesian Inference

Bayesian inference is an inference system often used in machine learning.
As the name implies, Bayesian inference relies on the well-known Bayes' Theorem:

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

In Context of the Scientific Method

Bayesian inference is essentially the core component.

Bayesian Inference

Bayesian Inference

Bayesian inference is an inference system often used in machine learning.
As the name implies, Bayesian inference relies on the well-known Bayes' Theorem:

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

Definitions

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

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

In Context of the Scientific Method

The events $H_1, H_2, \ldots, H_n$ are hypotheses which cover all possibilities.

Bayesian Inference

Bayesian Inference

Bayesian inference is an inference system often used in machine learning.
As the name implies, Bayesian inference relies on the well-known Bayes' Theorem:

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

Definitions

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

In Context of the Scientific Method

$P(H_i)$ represents the likelihood of the hypotheses under available background knowledge.

Bayesian Inference

Bayesian Inference

Bayesian inference is an inference system often used in machine learning.
As the name implies, Bayesian inference relies on the well-known Bayes' Theorem:

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

Definitions

E represents the event of encountering new evidence and

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

In Context of the Scientific Method

E represents the result of an experiment and $P(E)$ normalizes the results.

Bayesian Inference

Bayesian Inference

Bayesian inference is an inference system often used in machine learning.
As the name implies, Bayesian inference relies on the well-known Bayes' Theorem:

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

Definitions

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

In Context of the Scientific Method

$P(H_i|E)$, 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.

Bayesian Inference as a Dynamic Process

Bayesian Inference as a Dynamic Process

Given a sequence of observed evidence $E_1, E_2, \ldots$
Update the probabilities of the events $H_1, H_2, \ldots, H_n$ using Bayesian inference.

At each step the KL divergence $D_{KL}(P(H|E)||P(H))$ measures the information gained from moving from the prior distribution to the posterior distribution.

If we let H^* 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.

In 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.

Evolutionary Game Theory

"The theory of evolution by cumulative natural selection is the only theory we know of that is in principle capable of explaining the existence of organized complexity." - Richard Dawkins

The Replicator Dynamic

The **replicator dynamic** describes how the entities in a system, say phenotypes such as fur color, propagate themselves over time.

The Discrete Replicator Dynamic

Discrete Replicator Dynamic

The **discrete replicator dynamic** looks at the change in a population over discrete generations and is defined as:

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

Definitions

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. $\sum_i x_i = 1$.

The Discrete Replicator Dynamic

Discrete Replicator Dynamic

The **discrete replicator dynamic** looks at the change in a population over discrete generations and is defined as:

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

Definitions

$f = (f_1, \ldots, f_n)$ is the **fitness landscape** and each function $f_i(x)$ is the fitness of type i dependent on population distribution x.

The Discrete Replicator Dynamic

Discrete Replicator Dynamic

The **discrete replicator dynamic** looks at the change in a population over discrete generations and is defined as:

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

Definitions

$\bar{f}(x) = \sum_{i=1}^{n} x_i f_i(x)$ is the **average fitness**.

The Discrete Replicator Dynamic

Discrete Replicator Dynamic

The **discrete replicator dynamic** looks at the change in a population over discrete generations and is defined as:

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

Definitions

x_i' is the frequency of type i in the next generation of the population.

Evolutionary Stable States (ESS)

Evolutionary Stable State (ESS)

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 $\hat{x}$.

Remark

An ESS may not always exist.

Theorem 3

Theorem 3 (Harper, 2010)

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 i.e. the difference in potential information of two successive states:

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

is always be negative until the ESS state is reached.

Making the Connection

Formal Analogy

Both Bayesian inference and the replicator dynamic are **dynamical systems governed by decreasing potential information**.

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.

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 $\hat{x}$

Relation to Bayesian Inference

Relation to Bayesian Inference

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.

Applying the Connection

Evolution as a Learning Process

In essence, a population 'learns' how best to survive in its environment, stabilizing once ecological niches are filled.

The Scientific Method as an Evolutionary Process

Similarly, one can consider scientific hypotheses in a 'Darwinian competition' to match our observations of the world, so that only the 'fittest' ideas survive.

In Applications

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.

Questions?

"... in mathematics you don't understand things. You just get used to them."-John von Neumann

Questions?

References

- 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.
- M. Harper, *The replicator equation as an inference dynamic*, arXiv preprint arXiv:0911.1763 (2009).
- John Von Neumann and Oskar Morgenstern, *Theory of games and economic behavior (commemorative edition)*, Princeton university press, 2007.
- C.E. Shannon and W. Weaver, *The mathematical theory of communication (urbana, il)*, University of Illinois Press **184** (1949), 1032-54.
- Solomon Kullback and Richard A Leibler, *On information and suffciency*, The Annals of Mathematical Statistics **22** (1951), no. 1, 79-86.