ECE 4271: Evolutionary Processes, Evolutionary Algorithms, Evolutionary Games

Eric Johnson edj36

May 16, 2018

Abstract

ECE 4271: Evolutionary Processes, Evolutionary Algorithms, Evolutionary Games was taught in Spring 2018 at Cornell University by Professor Delchamps. What follows are my notes that I used to summarize his lectures, handouts, and other assigned readings throughout the semester.

Contents

1 Genetic Algorithms: An Overview

This section draws on material from Chapter 1 of *An Introduction to Genetic Algorithms* by Melanie Mitchell.

Computer scientists have studied evolutionary systems with the idea that evolution could be used as an optimization tool for engineering problems. Genetic algorithms were invented by John Holland in the 1960s to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaption might be imported into computer systems.

Why is evolution appealing? Many computational problems require a search through a huge number of possibilities for a solution and require a computer program to be adaptive. Evolution is effectively searching among an enormous number of possibilities for solutions and is an adaptive method for designing innovative solutions to complex problems. In this light we see how the mechanisms of evolution inspire computational methods.

1.1 Search Spaces and Fitness Landscapes

Searching in a "search space" is when you search among a collection of candidate solutions for a desired solution. This is very common in computer science. There is usually a notion of "distance" between candidate solutions in a search space. An example of this is Hammning distance - the number of locations at which corresponding bits differ in genotypes (bit strings of length l).

1.2 Elements of Genetic Algorithms

There is no rigorous definition of a *genetic algorithm.* However most genetic algorithms have at least the following elements:

- populations of chromosomes
- selection according to fitness - selects chromosomes in the population for reproduction, fitter chromosomes selected with higher probability
- crossover to produce random offspring - randomly choose locus and exchange subsequences after that locus between two chromosomes to create two offspring, e.g. if we have 1000100 and 1111111 and locus is 3, we get 1001111 and 1110100
- random mutation of new offspring - randomly flips bits in chromosome

Chromosomes are typically in the form of bit strings. Each chromosome can be thought of as a point in the search space of candidate solutions. GA processes populations of chromosomes, successively replacing one such population after another. GA most often requires a fitness function that assigns a score (fitness) to each chromosome in the current population. Fitness of chromosomes depends on how well chromosomes solve the problem at hand.

1.3 Simple Genetic Algorithm

Given a clearly defined problem to solve and bit string representation of candidate solutions, simple genetic algorithm is as follows

1. Start with randomly generated population of n l-bit chromosomes (candidate solutions)
2. Calculate fitness $f(x)$ of each chromosome x in population
3. Repeat until n offspring are created

(a) Select pair of chromosomes from current population, select with probability based on fitness. Select with replacement.

(b) With some crossover probability p_c, cross over the pair at random chosen point. If no crossover, form two offspring that are exact copies of parents.

(c) Mutate two offspring at each locus with probability p_m and place resulting chromosomes into the new population. If n is odd, one new population memeber can be discarded at random.

4. Replace current population with the new population
5. Go to step 2

Each iteration is called a generation. Entire set of generations is called a run.

Common selection is **fitness-proportionate selection**: the number of times an individual is expected to reproduce is equal to its fitness divided by the average of fitnesses in the population.

1.4 Genetic Algorithms and Traditional Search Methods

Some meanings of "search". First, **search for stored data** is to try and efficiently retrieve information stored in computer memory (binary search). **Search for paths to goals** is to efficiently find a set of actions that will move from a given initial state to a given goal (depth first search). **Search for solutions** is to efficiently find a solution to a problem in a large space of candidate solutions (this is very general). These are the kind of problems where genetic algorithms are used. We can see how the first is different than the next two. When we utilize a GA we are never searching for a explicitly defined piece of information. For the second problem, a tree style structure is typically used in the search. The general search for solutions can be considered a superset of searching for paths to goals. Genetic algorithms are one of many methods for searching for solutions problems. Other general methods include hill climbing, simulated annealing, and tabu search.

All of the search for solutions general methods

1. generate a set of candidate solutions
2. evaluate candidate solutions according to some fitness criteria
3. decide on the basis of this evaluation which candidates will be kept and which will be discarded
4. produce further variants by using some kind of operators on surviving candidates

1.5 Some Applications of Genetic Algorithms

Genetic algorithms have been used in a wide variety of optimization tasks. These can include numerical optimization and combinatorial optimization problems.

Machine learning problems including classification and prediction, and also using genetic algorithms to evolve parts ot machine learning systems, i.e. weights in a neural network.

Genetic algorithms are used to model innovation, bidding strategies, and markets in economics.

In addition to the three above, others include automatic programming, immune systems, ecology, population genetics, evolution and learning, and social systems.

An example from economics is the prisoner's dilemma and game theory. A strategy for prisoner's dilemma games is TIT FOR TAT.

> TIT FOR TAT: cooperate in the first game, do whatever the other player did it its move in the previous game as your move in the next game.

Genetic algorithms were used to develop strategies that could score better than the human developed TIT FOR TAT. This doesn't make it worse, because TIT FOR TAT works well for *every* game played, while the genetic algorithms typically developed strategies that exploited weaknesses in the strategies it experienced in its given environment. Conclusion to this was that GAs are as good at doing what evolution often does: develop highly specialized adaptations to specific characteristics of an environment (Axelrod). In more general/changing environments, research (Axelrod) found that GAs discovered strategies that were essentially variants of TIT FOR TAT.

1.6 How Do Genetic Algorithms Work?

Genetic algorithms are simple to describe and program, but their behavior can be complicated. Many open questions exist about how much they work and what types of problems they are best suited for.

Traditional theory of GAs is that GAs work by discovering, emphasizing, and recombining good "building blocks" of solutions in a highly parallel fashion (building block hypothesis). Holland introduced the notion of schemas to formalize these informal building blocks. A **schema** is a set of bit strings that can be described by a template made up of ones, zeros, and asterisks (wild cards), e.g. $H = 1 * * * 1$. H stands for hyperplane in the l dimensional space of length l bit strings. String that fit the schema template H are said to be instances of H. The number of defined bits in a schema is the number of non-asterisk bits. The defined lenght of a schema is the destance between its outermost defined bits. Not every possible subset of the set of length l bit strings can be described as a schema. There are 2^l possible bit strings of length l, so 2^{2^l} possible subsets of strings, but there are only 3^l possible schemas. Any given bit string of length l is an instance of 2^l different schemas. Any given population of n strings contains instances of between 2^l and $n \cdot 2^l$ different schemas. GAs don't explicitly represent or calculate estimates of schema average fitness, but GAs behavior (in terms of increase and decrease in numbers of instances of given schemas in the population) can be described as though it actually were calculating and storing the average fitnesses.

Define $m(H,t)$ to be the number of instances of schema H at time t. $d(H)$ is the defining length of H, l is the length of bit strings in search space. p_c is probability that single point crossover will be applied to a string. $\hat{u}(H,t)$ is the observed average fitness of H at time t. $f(x)$ is the fitness of x and $\bar{f}(t)$ is the average fitness of the population at time t. Assume selection is carried out as described above: expected number of offspring of string x is equal to $f(x)/\bar{f}(t)$. The expected number of instances of H at time $t+1$ (ignoring crossover and mutation) is

$$E(m(H,t+1)) = \sum_{x \in H} \frac{f(x)}{\bar{f}(t)} = \frac{\hat{u}(H,t)}{\bar{f}(t)} m(H,t)$$

Schema H is said to "survive" under single point crossover if one of the offspring is also an instance of schema H. A lower bound on the probability $S_c(H)$ that H will survive single-point crossover is

$$S_c(H) \geq 1 - p_c\Big(\frac{d(H)}{l-1}\Big)$$

The probability of survival under crossover is higher for shorter schemas. Let p_m be probability of any bit being mutated. $S_m(H)$, the probability that schema H will survive under mutation of an instance of H is

$$S_m(H) = (1-p_m)^{o(H)}$$

where $o(H)$ is the order of H (number of defined bits in H). The probability of survival under mutation is higher for lower order schemas. Adding these effects to the expectation above, we arrive at the **Schema Theorem** (Holland)

$$E(m(H,t+1)) \geq \frac{\hat{u}(H,t)}{\bar{f}(t)} m(H,t)\Big(1 - p_c\frac{d(H)}{l-1}\Big)\Big((1-p)^{o(H)}\Big)$$

Common interpretation is that the theorem implies that short, lower order schemas whose average fitness remains above the mean will receive exponentially increasing numbers of samples over time. Also, to make more formal, the **building block hypothesis** that goes along with this is that a genetic algorithm seeks optimal performance through the juxtaposition of short, low-order, high-performance schemas, called the building blocks. Here cross-over is believed to be major source of GAs power. Later we will see that this is not the case (from Delchamps lectures).

In evaluating population of n strings, GA is implicitly estimating the average fitness of all schemas present in the population and increasing/decreasing representation according to the Schema Theorem. This simultaneous implicit evaluation of large numbers of schemas in population of n strings is known as *implicit parallelism*. Selection biases sampling procedure to schemas who's fitness is estimated to be above average. Schema Theorem and Building Block Hypothesis deal primarily with selection and crossover. Holland says mutation serves to insure against loss of diversity. (Delchamps will show that there is more to it than this).

2 Markov Chains

2.1 What is a Markov Chain?

Building blocks of a Markov chain are

- State space S, generally will be set of positive integers or finite set $\{1, 2, \ldots, M\}$ for $M > 0$.
- For each $i \in S$, a set of one-step transition probabilities $P(i, j)$ for $j \in S$. $P(i, j)$ denotes the probability that the Markov chain will be "in" state j at the next time step given that it is "in" state i now. $P(i, j) \geq 0$ for all i and j, and $\sum_{j \in S} P(i, j) = 1$ for every i.
- Initial distribution $\pi(0)$ over states. $\pi_i(0)$ represents the probability that the Markov chain is in state i at time 0, so $\pi_i(0) \geq 0$ for all $i \in S$ and $\sum_{i \in S} \pi_i(0) = 1$.

Markov Property For every $n \in \mathbb{N}$ and every $i_0, i_1, \ldots, i_{n-1}$, i, and $j \in S$, the probability that the Markov chain will be in state j at time $n + 1$ given that it is in state i at time n and state i_m at time m for every $0 \leq m < n$ is $P(i, j)$.

Key point is the irrelevance of past history to computing the probability of making an $i \to j$ transition at time n given that you know the chain is in state i at time n.

For any sequence of states $i_0, i_1, \ldots, i_n$, the probability of seeing the Markov chain pass through these states in order starting at time 0 is

$$\pi_{i_0}(0)P(i_0, i_1)P(i_1, i_2) \cdots P(i_{n-1}, i_n)$$

Define a *run* of a Markov chain as the particular infinite sequence of states it visits starting at time 0 as a result of the various random events that determine the starting state and subsequent state transitions.

For any $m > 0$, define the m-step transition probabilities as follows: $P^{(m)}(i, j)$ is the probability that the Markov chain visits state j at time m given that it started in state i at time 0. This is the same as the probability that, m time steps from now, the chain will be in state j given that it is in state i now (no matter what now is). We can inductively define recursive formula for m-step transition probabilities via

$$P^{(m+1)}(i, j) = \sum_{k \in S} P^{(m)}(i, k)P(k, j) \;\; \forall\, i, j \in S \text{ and } m > 0$$

If M (number of states) is finite, form an $M \times M$ matrix P from transition probabilities via

$$[P]_{ij} = P(i, j), \;\; 1 \leq i, j \leq M$$

Raising P to powers gives matrices of higher order transition probabilities.

$$P^{(m)}(i, j) = [P^m]_{ij}, \;\; 0 \leq i, j \leq M, \;\; m > 0$$

2.2 Random and non-random quantities; transient and recurrent states

Most fundamental random quantity associated with a Markov chain is X_n, the state the Markov chain is in at time n.

$$\text{Prob}\{X_{n_1} = i_1\} = \begin{cases} \sum_{j \in S} \pi_j(0) P^{(n_1)}(j, i_1) & n_1 > 0 \\ \pi_{i_1}(0) & n_1 = 0 \end{cases}$$

The initial distribution and one-step transition probabilities determine the Markov chain completely.

For any two states i and j in S and any $k > 0$, define $f_{ij}^{(k)}$ as the probability that the first time after time 0 that the chain visits state j is time k, given that the chain starts in state i at time 0 (f = first). If we take $i = j$, we get $f_{jj}^{(k)}$, the probability that the first time the chain returns to state j starting from state j is k time steps in the future.

For any states i and j set

$$r_{ij} = \sum_{k=1}^{\infty} f_{ij}^{(k)}$$

r_{ij} is the probability that the chain will reach state j in finite time given that it starts in state i (r = return). Setting $i = j$ gives r_{jj}, the probability that, starting from state j, the chain will return to state j in finite time. $f_{ij}^{(k)}$ and r_{ij} are non-random quantities.

An important random quantity is the first hitting time T_j, defined for any $j \in S$, T_j is the first positive time that the chain visits state j. If chain follows path that never hits state j, $T_j = \infty$.

Given any state i, any random quantity Z associated with Markov chain, and any possible value z for Z, define

$$\text{Prob}_i\{Z = z\}$$

as the probability that $Z = z$ given that the chain started in state i at time 0.

Define

$$E_i(Z)$$

as the expected value of Z given that the chain started in state i at time 0.

$$\text{Prob}_i\{T_j = k\} = f_{ij}^{(k)}$$

Observe:

- if $r_{ij} = 0$ then $\text{Prob}_i\{T_j = \infty\} = 1$
- if $r_{ij} = 1$ then $\text{Prob}_i\{T_j < \infty\} = 1$
- if the chain starting from state i has positive probability r_{ij} of hitting state j in finite time but $r_{ij} < 1$, then $\text{Prob}_i\{T_j = \infty\} = 1 - r_{ij}$ and $\text{Prob}_i\{T_j < \infty\} = r_{ij}$

The T_j are called first hitting times for obvious reasons. If we start the chain off in state j, then we call T_j the first return time for state j.

Useful fact: Standard formula for sum of geometric series gives

$$\sum_{m=1}^{\infty} m(1-p)^{m-1} = -\frac{d}{dp}\Big(\frac{1}{1-(1-p)}\Big) = \frac{1}{p^2}$$

Another random quantity of interest is $N_j(n)$, for any $n > 0$ and $j \in S$ is the number of visits the Markov chain makes to state j over time interval $1 \le m \le n$. $N_j(n)$ is a random quantity for every j and n since its value depends on the path the Markov chain follows. Use N_j to denote total number of times the Markov chain hits j after time 0,

$$N_j = \lim_{n \to \infty} N_j(n) \;\; \forall\, j \in S$$

In particular, we don't count being in state j at time 0 as contributing to $N_j(n)$ or N_j.

$$E_i(N_j(n)) = \sum_{m=1}^{n} P^{(m)}(i,j) \;\; \forall\, i,j \in S$$

Set $i = j$ to get

$$E_j(N_j(n)) = \sum_{m=1}^{n} P^{(m)}(j,j) \;\; \forall\, j \in S$$

Take limit $n \to \infty$ to get

$$E_i(N_j) = \sum_{m=1}^{\infty} P^{(m)}(i,j) \;\; \forall\, i,j \in S$$

Recurrent a state is recurrent iff $r_{jj} = 1$. I.e. j is a recurrent state if and only the Markov chain starting in state j will return to j in finite time with probability 1.

Transient a state is transient iff $r_{jj} < 1$. I.e. state j is transient if and only there's positive probability that the chain, having started in j, will never return to j.

Let S_T be the set of transient states and S_R be the set of recurrent states. A first decomposition of the state space S of the Markov chain is

$$S = S_T \cup S_R$$

Theorem(s) State j is recurrent iff

$$\text{Prob}_j\{N_j = \infty\} = 1$$

If j is recurrent, then for any state i

$$E_i(N_j) = \begin{cases} 0 & \text{if } r_{ij} = 0 \\ \infty & \text{if } r_{ij} > 0 \end{cases}$$

State j is transient if and only if

$$\text{Prob}_j\{N_j < \infty\} = 1$$

If j is transient, then for any state i

$$E_i(N_j) = \frac{r_{ij}}{1 - r_{jj}}$$

Fact If i is recurrent and $i \to j$, then j is also recurrent and $j \to i$. Furthermore, $r_{ij} = r_{ji} = 1$. $i \to j$ means that, with positive probability, the chain starting in state i will hit state j in finite time.

2.3 Recurrence classes and limits of expected time averages

We've learned so far that regardless of the initial distribution, the Markov chain will never visit a transient state infinitely often and that if the chain ever visits a recurrent state j, it will revisit j infinitely often.

Irreducible A set of states C is *closed* if and only if for every $i \in C$, $r_{ik} = 0$ when $k \notin C$. A closed set C of states is said to be *irreducible* if and only if $i \to j$ for every i and j in C. If the entire state space S is irreducible, we say that the Markov chain is irreducible.

The chain can't escape from a closed set C of states once it is in C. Think of any closed set of states C as constituting its own private little Markov chain with state space C and transition probabilities $\{P(i,j) : i,j \in C\}$.

Recurrence class For any recurrent state i, define the *recurrence class* of i as the set of all $j \in S$ such that $i \to j$. Two different recurrence classes cannot intersect.

Refining our decomposition of Markov chains above, we have the set S_R of recurrent states splits into the disjoint union of one or more recurrence classes. Each of these is a closed, irreducible set of states.

Mean first return time for state j is $m_j = E_j(T_j)$. Since

$$\mathrm{Prob}_j\{T_j = k\} = f_{jj}^{(k)} \ \ \forall\, k > 0$$

have

$$E_j(T_j) = \sum_{k=1}^{\infty} k f_{jj}^{(k)} \ \ \forall\, j \in S$$

If j is a transient state, then by definition $m_j = \infty$, it is also still possible that $m_j = \infty$ for a recurrent state...

Positively recurrent A recurrent state j is *positively recurrent* if and only if $m_j < \infty$.

Null recurrent A recurrent state j is *null recurrent* if and only if $m_j = \infty$. How? $m_j = \infty$ occurs, roughly speaking, when $f_{jj}(k)$ doesn't go to zero fast enough as $k \to \infty$. If with significant probability you have to wait a very long time to return to state j, then it's possible that, on average, you'll wait essentially forever.

Fact For any recurrence class C, either every $j \in C$ is positively recurrent or every $j \in C$ is null-recurrent.

Theorem With notation as we have above,

- If j is a transient state, then

$$\lim_{n\to\infty} \frac{E_j(N_j(n))}{n} = \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} P^{(m)}(j,j) = 0$$

- If j is a recurrent state, then

$$\lim_{n\to\infty} \frac{E_j(N_j(n))}{n} = \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} P^{(m)}(j,j) = \frac{1}{m_j}$$

If j is null-recurrent, then the long-term expected fraction of the time the chain spends in state j is zero, just as it would have been if j were transient.

Theorem With notation as we have above,

- If j is a transient state, then for any state i

$$\lim_{n\to\infty} \frac{E_i(N_j(n))}{n} = \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} P^{(m)}(i,j) = 0$$

- If j is a recurrent state, then for any state i

$$\lim_{n\to\infty} \frac{E_i(N_j(n))}{n} = \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} P^{(m)}(i,j) = \frac{r_{ij}}{m_j}$$

Corollary (to theorem above) If the number of states in a Markov chain is finite, then the chain has at least one recurrent state and no null-recurrent states.

2.4 Stationary distributions

For each m, $\pi(m)$ describes the distribution over states of the Markov chain, in the sense that $\pi_i(m)$, for each $i \in S$ (S is set of states), is the probability that we are in state i at time m (i.e., after m steps on the chain). The distributions follow a simple recursion in m. To find $\pi_j(m+1)$, we can use transition probabilities $P(i,j)$ as follows

$$\pi_j(m+1) = \sum_{i \in S} \pi_i(m) P(i,j)$$

for every $j \in S$ and $m > 0$. It follows by induction that

$$\pi_j(m) = \sum_{i \in S} \pi_i(0) P^{(m)}(i,j)$$

for every $j \in S$ and $m > 0$.

At any time $m \geq 0$, say X_m is the state the Markov chain is in at time m. For each $i \in S$ and $m \geq 0$, the probability $X_m = i$ is $\pi_i(m)$.

Stationary Distribution A vector $\bar{\pi} = (\bar{\pi}_i)_{i \in S}$ is a *stationary distribution* for the Markov chain with transition probabilities $P(i,j)$ iff $\bar{\pi}_i \geq 0 \ \forall \ i \in S$, $\sum_{i \in S} \bar{\pi}_i = 1$, and

$$\bar{\pi}_j = \sum_{i \in S} \bar{\pi}_i P(i,j) \quad \forall \ j \in S$$

If $\bar{\pi}$ is a stationary distribution for a Markov chain in the sense of the definition directly above, and we set $\pi(0) = \bar{\pi}$, then clearly $\pi(m) = \bar{\pi}$ for every $m > 0$ because of the recursion above. Thus again by induction, for every $m > 0$ and every stationary distribution $\bar{\pi}$ we have

$$\bar{\pi}_j = \sum_{i \in S} \bar{\pi}_i P^{(m)}(i,j) \quad \forall \ j \in S$$

A Markov chain can have many stationary distributions, or exactly one, or none at all.

Theorem If $\bar{\pi}$ is a stationary distribution, then $\bar{\pi}_j = 0$ if j is a transient or null-recurrent state.

This asserts that any stationary distribution $\bar{\pi}$ for a Markov chain must be concentrated on the set of positively recurrent states in the sense that $\bar{\pi}_j > 0$ only if j is positively recurrent. A Markov chain lacking positively recurrent states has no stationary distributions. Also, any Markov chain with at least one positively recurrent state has at least one stationary distribution.

Positively Recurrent Class Consider a Markov chain that has at least one positively recurrent state and therefore, at least one recurrence class C consisting solely of positively recurrent states, such a recurrence class C is a *positively recurrent class.*

Lemma Suppose a Markov chain has a positively recurrent class C. Define a distribution $\bar{\pi}$ for the Markov chain by setting

$$\bar{\pi}_j = \begin{cases} \frac{1}{m_j} = q_j & \text{if } j \in C \\ 0 & \text{if } j \notin C \end{cases}$$

Then $\bar{\pi}$ is a stationary distribution for the Markov chain. Furthermore, $\bar{\pi}$ is concentrated on the recurrence class C since $\bar{\pi}_j = 0$ for $j \notin C$.

Theorem Suppose a Markov chain has at least one positively recurrent state. For each positively recurrent class C, the chain has a unique stationary distribution $\bar{\pi}^C$ concentrated on C. The distribution $\bar{\pi}^C$ is given by

$$\bar{\pi}^C = \begin{cases} \frac{1}{m_j} & \text{if } j \in C \\ 0 & \text{if } j \notin C \end{cases}$$

where $m_j = E_j(T_j)$ is the expected return time to state j.

Corollary If $\bar{\pi}$ is a stationary distribution for a Markov chain, then for any positively recurrent class C, either $\bar{\pi} = 0$ for every $j \in C$ or

$$\frac{\bar{\pi}_j}{\sum_{i \in C} \bar{\pi}_i} = \bar{\pi}_j^C \quad \forall\, j \in C$$

Consider a chain that has at least one positively recurrent class. Let Π be the set of all positively recurrent classes. We can use the Corollary above as a recipe for generating all the stationary distributions for a Markov chain. Corollary states that every stationary distribution $\bar{\pi}$ for the chain takes the form

$$\bar{\pi} = \sum_{C \in \Pi} \lambda_C \bar{\pi}^C$$

where $\lambda_C \geq 0\ \forall\ C$ and $\sum_{C \in \Pi} \lambda_C = 1$. For each C

$$\lambda_C = \sum_{j \in C} \bar{\pi}_j$$

Every stationary distribution for the chain is a convex combination of the stationary distributions concentrated on the various recurrence classes.

2.5 Convergence of time averages with probability 1

Suppose during a run of a Markov chain you collect a certain amount of money every time you hit a state. Say you collect $f(i)$ whenever you are in state i. The average amount of money you collect per unit time on a given run up through time n is

$$S_n(f) = \frac{1}{n} \sum_{m=1}^{n} f(X_m)$$

where X_m is the state of the Markov chain at time m. If we define an indicator $\chi_{\{j\}}$ for any state j as follows

$$\chi_{\{j\}}(i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

then

$$S_n(\chi_{\{j\}}) = \frac{1}{n} N_j(n)$$

is the fraction of the time the Markov chain spends in state j during the time interval $1 \leq m \leq n$.

Strong Law of Large Numbers (SLLN) Let $\{Z_m : m \geq 1\}$ be a sequence of independent real-valued random variables with common probability distribution ρ. Let $E_\rho(Z)$ be the common expected value of all the Z_m, and assume $E_\rho(Z)$ is finite. Then, with probability 1,

$$\lim_{n \to \infty} \frac{1}{n} \sum_{m=1}^{n} Z_m = E_\rho(Z)$$

Theorem For an arbitrary homogeneous Markov chain with countable state space, and with notation as above, Markov chain,

- if j is a transient state, then for any state i,

$$\lim_{n \to \infty} \frac{N_j(n)}{n} = 0$$

for every path emanating from i with the possible exception of a set of paths of probability zero.

- if j is a recurrent state, the for any state i,

  $$\lim_{n\to\infty} \frac{N_j(n)}{n} = \frac{1}{m_j}$$

  for any path emanating from i that hits j in finite time, and

  $$\lim_{n\to\infty} \frac{N_j(n)}{n} = 0$$

  for any path emanating from i that doesn't hit j, both assertions holding with the possible exception of a set of paths of probability zero.

Theorem Given a Markov chain with finite state space $\mathcal{S}$, suppose the chain has only one recurrence class. Then for any $f : \mathcal{S} \to \mathbb{R}$ and for any initial distribution $\pi(0)$,

$$S_n(f) = \frac{1}{n}\sum_{m=1}^{n} f(X_m)$$

converges with probability 1 as $n \to \infty$ to

$$E_{\bar{\pi}^*}(f) = \sum_{j\in S} f(j)\bar{\pi}^*_j$$

where $X(m)$ is the state of the Markov chain at time m and $\bar{\pi}^*$ is the unique stationary distribution for the Markov chain.

If a finite-state Markov chain has multiple recurrence classes, the limiting behavior of $S_n(f)$ depends on the initial distribution for the Markov chain.

Theorem Suppose a Markov chain with state space $\mathcal{S}$ is irreducible and has only positively recurrent states. Let π^* be the unique stationary distribution guaranteed by the Theorem above. Let $f : \mathcal{S} \to \mathbb{R}$ be any function for which $E_{\pi^*}(f)$ is finite. Then for any initial distribution $\pi(0)$,

$$S_n(f) = \frac{1}{n}\sum_{m=1}^{n} f(X_m)$$

converges with probability 1 as $n \to \infty$ to $E_{\pi^*}(f)$, where $X(m)$ is the state of the Markov chain at time m.

2.6 Convergence of distributions

Periodic Let $d > 1$ be an integer. A state j of a Markov chain is *periodic of period* d iff d is the greatest common divisor of the set of integers $m > 0$ for which $P^{(m)}(j,j) > 0$.

Aperiodic State j is *aperiodic* iff the greatest common divisor of the set of integers $m > 0$ for which $P^{(m)}(j,j) > 0$ is 1.

Idea: Given a state $j \in \mathcal{S}$, list all $m > 0$ for which it's possible to make a j-to-j transition in exactly m steps. For these and only these m-values, $P^{(m)}(j,j) > 0$. If all are multiples of some integer $d > 1$, then state j is periodic. If no such d exists, j is aperiodic. Periodic states can be either transient or recurrent and recurrent periodic states can be either positively recurrent or null-recurrent.

Fact Let i be a recurrent state with period $d > 1$. If $i \to j$, then j is also periodic with period d.

This Fact implies that if i is an aperiodic recurrent state, then every j in the recurrence class of i is also aperiodic. Furthermore, every state in a periodic recurrence class has the same period, so we can talk about things like "d-periodic recurrence classes".

Fact If j is an aperiodic state and i is any state for which $i \to j$, there $\exists\ M > 0$ such that $P^{(m)}(i,j) > 0\ \forall\ m \geq M$.

Theorem Suppose a Markov chain with state space $\mathcal{S}$ is irreducible and has only aperiodic positively recurrent states. Let π^* be the unique stationary distribution (guaranteed by Theorem above). Then

$$\lim_{n\to\infty} P^{(n)}(i,j) = \frac{1}{m_j} = \pi_j^* \quad \forall\ i,j \in \mathcal{S}$$

It follows that for any initial distribution $\pi(0)$, $\pi(n)$ converges to π^* as $n \to \infty$ in the sense that

$$\lim_{n\to\infty} \pi_j(n) = \pi_j^*$$

for every $j \in \mathcal{S}$.

Theorem Suppose a Markov chain with state space $\mathcal{S}$ is irreducible and has only aperiodic positively recurrent states. Let π^* be the unique stationary distribution (guaranteed by Theorem above). Then, for any initial distribution $\pi(0)$, $\pi(n)$ converges to π^* as $n \to \infty$ in the sense that

$$\lim_{n\to\infty} \left(\sup_{A\subset\mathcal{S}} \left| \sum_{j\in A} (\pi_j(n) - \pi_j^*) \right| \right) = 0$$

Theorem Suppose every state of a Markov chain is positively recurrent and aperiodic. Then

$$\lim_{n\to\infty} P^{(n)}(i,j) = \begin{cases} \frac{1}{m_j} & \text{when } i \text{ and } j \text{ are in the same recurrence class} \\ 0 & \text{else} \end{cases}$$

Furthermore, for any initial distibution $\pi(0)$, $\pi(n)$ converges as $n \to \infty$ in the sense that

$$\lim_{n\to\infty} \pi_j(n) = \sum_{C\in\Pi} \lambda_C \bar{\pi}_j^C \quad \forall\ j \in S$$

Where Π is the set of all recurrence classes and

$$\lambda_C = \sum_{i\in C} \pi_i(0) \quad \forall\ C \in \Pi$$

For any recurrence class C, λ_C is the total probability that the initial distribution $\pi(0)$ allots to C. The limiting distribution allots the same total probability to C, but in general it's spread differently among the states in C. If there are transient states in the Markov chain, λ_C will have different values.

Absorption probability For each transient state i and recurrence class C, we can define r_i^C as the common value of r_{ij} for $j \in C$. r_i^C is the *absorption probability* of the transient state i into the recurrence class C.

Theorem Suppose every state of a Markov chain is either transient or positively recurrent and aperiodic. Let S_T be the set of transient states and S_R be the set of recurrent states. Then for any $i \in S_T$

$$\lim_{n\to\infty} P^{(n)}(i,j) = \begin{cases} 0 & \text{when } j \in S_T \\ \frac{r_{ij}}{m_j} & \text{when } j \in S_R \end{cases}$$

Furthermore, for any initial distribution $\pi(0)$, $\pi(n)$ converges as $n \to \infty$ in the sense that

$$\lim_{n\to\infty} \pi_j(n) = \sum_{C\in\Pi} \lambda_C \bar{\pi}_j^C \quad \forall\ j \in S$$

Where Π is the set of all recurrence classes and

$$\lambda_C = \sum_{i\in C} \pi_i(0) + \sum_{i\in S_T} \pi_i(0) r_i^C \quad \forall\ C \in \Pi$$

In particular, $\lim_{n\to\infty} \pi_j(n) = 0$ when $j \in S_T$.

Fact If i and j are states with period d that lie in the same recurrence class, then $\exists$ an integer ρ_{ij}, with $0 \leq \rho_{ij} < d$, such that when n is sufficiently large, $P^{(n)}(i,j) > 0$ iff $n = qd + \rho_{ij}$ for some positive integer q.

Theorem Suppose a Markov chain with state space $\mathcal{S}$ is irreducible and has only positively recurrent states with period $d > 1$. Let π^* be the unique stationary distribution (guaranteed by Theorem above). Then, for every i and j in $\mathcal{S}$, $\exists$ an integer ρ_{ij}, with $0 \leq \rho_{ij} < d$, such that $P^{(n)}(i,j) > 0$ when n is sufficiently large iff $n = qd + \rho_{ij}$ for some integer $q > 0$. Futhermore for every i and j,

$$\lim_{n\to\infty} P^{(qd+\rho_{ij})}(i,j) = d\pi_j^*$$

3 Metropolis

3.1 Metropolis-Hastings Algorithm

You have a large finite set $S = \{1, 2, 3, \dots, M\}$. Let π be probability distribution on S, assume $\pi_j > 0$ for all $j \in S$. Goal is to compute expected values of the form

$$E_\pi(f) = \sum_{\in S} f(j)\pi_j$$

for various functions $f : S \to \mathbb{R}$. You can only know π up to an arbitrary multiplicative constant - you might know only $p_j = c_0\pi_j$ for every $j \in S$ for some unknown $c_0 > 0$. Note

$$c_0 = \sum_{j\in S} p_j$$

because $\sum_{j\in S} \pi_j = 1$. However, S might be so large and the formulas for the p_j so complicated that computing c_0 is intractable. In order to find $E_\pi(f)$, it suffices to draw an independent sequence of samples $\{Z_m : m > 0\}$ from S distributed according to π. The Strong Law of Large Numbers implies that with probability 1,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} f(Z_m) = E_\pi(f)$$

Suppose we could construct an irreducible Markov chain with state space S whose unique stationary distribution π^* was π. If X_m is the state of the Markov chain at time $m > 0$ starting from any initial state in S, the various limit theorems for Markov chains guarantee that

$$\lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} f(X_m) = E_{\pi^*}(f) = E_\pi(f)$$

with probability 1. The multi-set of states you obtain by recording the various states visited by a typical Markov-chain run $\{X_n : n > 0\}$ will be distributed over S according to π. Coming up with such a Markov chain would not only generate a population of points from S distributed according to π, but would also provide a recursive way of approximating $E_\pi(f)$ for any f: simply run the Markov chain and apply the equation directly above. The Metropolis- Hastings algorithm provides an ingenious technique for constructing irreducible such chains that have π as their (necessarily unique) stationary distribution.

First some preliminaries. If $\bar{\pi}$ is any stationary distribution for the Markov chain, then

$$\sum_{i\in S} \bar{\pi}_i P(i,j) = \bar{\pi}_j \quad \forall\, j \in S$$

Balance conditions Every stationary distribution satisfies the following conditions (the *balance conditions* on $\bar{\pi}$.

$$\sum_{i\in S} \bar{\pi}_i P(i,j) = \left(\sum_{i\in S} P(j,i)\right)\bar{\pi}_j = \sum_{i\in S} P(j,i)\bar{\pi}_j$$

If π is any distribution that satisfies the balance conditions, then π is a stationary distribution because

$$\sum_{i\in S} \pi_i P(i,j) = \left(\sum_{i\in S} P(j,i)\right)\pi_j = \pi_j \quad \forall\, j \in S$$

Detailed balance conditions One way, but not the only way, for a distribution π to satisfy the balance conditions is for the two infinite sums in the balance conditions to be equal term-by-term, i.e.

$$\pi_i P(i,j) = P(j,i)\pi_j \quad \forall\, i,j \in S$$

these conditions are the *detailed balance conditions*. It follows that if π satisfies the detailed balance conditions, then π is a stationary distribution for the Markov chain.

Back to our original problem. We have π and S and we have access to $p_j = c_o\pi_j$ for some $c_o > 0$ like above. Let $Q(i,j)$ be any set of transition probabilities of a Markov chain that satisfies

- $Q(i,j) = Q(j,i)\ \forall\, i,j \in S$
- The Markov chain with transition probabilities $Q(i,j)$ is irreducible.

Many Q's are possible, simplest is random walk on S. Now define $P(i,j)$ for every i and j in S as follows

$$P(i,j) = \begin{cases} Q(i,j) & \text{if } i \neq j \text{ and } p_j \geq p_i \\ Q(i,j)\frac{p_j}{p_i} & \text{if } i \neq j \text{ and } p_j < p_i \\ Q(i,i) + \sum_{j\in S_i} Q(i,j)\left(1 - \frac{p_j}{p_i}\right) & \text{if } i = j \end{cases}$$

where $S_i = \{j \in S : p_j < p_i\}$. It turns out that this makes an irreducible Markov chain on S with $P(i,j)$ as the transition probabilities, and with π as the stationary distribution. By running the chain starting from an arbitrary initial state we can use the methodology describe above to approximate $E_\pi(f)$ for functions $f : S \to \mathbb{R}$. We can also produce a set $\mathcal{P}$ of samples from S distributed according to π. To construct $\mathcal{P}$ use this algorithm:

- **Initialization:** Set $\theta_0 = i$, where $i \in S$ is an arbitrary state. Set $\mathcal{P} = \{\theta_0\}$. Proceed to Proposal Step.
- **Proposal Step:** Given θ_m, choose $\psi_{m+1} \in S$ according to the transition probabilities $Q(i,j)$; that is, set $\psi_{m+1} = j$ with probability $Q(i,j)$ when $\theta_m = i$. Proceed to the Accept-Reject Step.
- **Accept-Reject Step:** When $\theta_m = i$ and $\psi_{m+1} = j$,
  - If $p_j \geq p_i$, set $\theta_{m+1} = \psi_{m+1}$.
  - If $p_j < p_i$, set $\theta_{m+1} = \psi_{m+1}$ with probability $\frac{p_j}{p_i}$ and set $\theta_{m+1} = \theta_m$ with probability $1 - \frac{p_j}{p_i}$.

  Add θ_{m+1} to $\mathcal{P}$ and return to Proposal Step.

4 Schema Theorem

Everything that follows pertains to a standard genetic algorithm with population size n (an even number), string length L, mutation probability p_m, and one-point crossover probability p_c. The population at time t is $P(t)$, the selection is fitness-proportional, and the (strictly positive) fitness function is f.

Fitness-proportional selection gives

$$p_{sel}(x) = \frac{f(x)}{\sum_{y\in P(t)} f(y)}$$

as the probability that x is selected on any given parental draw for any individual $x \in P(t)$.

Subset Selection Lemma If Q is any subset of $P(t)$, then the probability that any given parental draw yields a member of Q is

$$p_{sel}(Q) = \frac{\sum_{x \in Q} f(x)}{\sum_{y \in P(t)} f(y)}$$

Consequently, the expected number of members of Q selected as parents for $P(t+1)$ is

$$np_{sel}(Q) = n\left(\frac{\sum_{x \in Q} f(x)}{\sum_{y \in P(t)} f(y)}\right) = |Q|\left(\frac{\frac{1}{|Q|}\sum_{x \in Q} f(x)}{\frac{1}{n}\sum_{y \in P(t)} f(y)}\right)$$

where $|Q|$ is the number of individuals in Q.

Order The *order of a schema* H, written $o(H)$, is the number of defined bits (i.e. non-stars) in the defining symbol for H.

Defining length The *defining length* of a schema H, written $d(H)$, is the "distance" between the leftmost and rightmost defined bits in the defining symbol for H.

4.1 New lower bound

For schema H, and $Q = H \cap P(t)$, (Q is just the subset of all H-representatives in $P(t)$), the lower bound on expected number of members of H selected as parents for $P(t+1)$ is as follows

$$E(|H \cap P(t+1)|) \geq np_{sel}(Q)\left(1 - p_c\frac{d(H)}{L-1}\right)(1-p_m)^{o(H)} + n(p_{sel}(Q))^2 p_c \frac{d(H)}{L-1}(1-p_m)^{o(H)}$$

The first term on the right-hand side of this is the same as the lower bound in Mitchell's version of the **Schema Theorem** (after accounting for differences in notation), which is mentioned in Section 1 above. The additional nonnegative term (second term of right-hand side of above) provides a "tighter" lower bound than the "official" Schema Theorem that was given in the book.

5 Game Theory

n-player finite strategic-form game An *n-player finite strategic-form game* consists of the following:

- A set of n agents that we'll call Player 1, Player 2, ..., Player n
- For each $i \in \{1, 2, \ldots, n\}$, a finite set of possible actions A_i for Player i
- For each $i \in \{1, 2, \ldots, n\}$, a payoff function

$$u_i : A_1 \times A_2 \times \cdots \times A_n \to \mathbb{R}$$

where $u_i(a_1, a_2, \ldots, a_n)$ represents the payoff to Player i given that each Player j takes action $a_j \in A_j$, $1 \leq j \leq n$.

The possible outcomes of an n-player game are in one-to-one correspondence with n-tuples $(a_1, a_2, \ldots, a_n)$ of actions, where $a_i \in A_i$ for all i.

An important and surprisingly delicate assumption lurking behind the definition of an *n-player finite strategic-form game* is that the structure of the game is common knowledge to all the players, every player knows the actions available to all the players and knows every player's payoff function, but that every player knows that every player knows these things, and that every player knows that every player knows that every player knows these things, and so on ad infinitum.

5.1 Example Games

When a game is played, Player 1's action determines the row and Player 2's determines the column, and the corresponding matrix entry describes the game outcome that results. The ordered pair of numbers in the matrix entry represents

(payoff to Player 1, payoff to Player 2)

arising from the outcome.

Prisoners' Dilemma Perhaps the most famous game of all. Payoffs are

	C	D
C	(3,3)	(0,5)
D	(5,0)	(1,1)

Even though both would fare better if both took action C, the constraint that they come up with their strategies independently and in isolation forces the unfortunate (D, D) outcome.

Battle of the Sexes/Brothers Payoffs are

	C	B
C	(3,2)	(1,1)
B	(1,1)	(2,3)

Battle of the Brothers has no particular outcome that we expect to see when rational players get around to playing the game.

Matching Pennies The idea is that both players turn a hidden penny either heads-up (action H) or tails-up (action T) when it comes time to play the game. Then they unveil their hidden pennies, and Player 1 collects both pennies if the faces match while Player 2 collects both pennies if the faces disagree. Payoffs are

	H	T
H	(1,-1)	(-1,1)
T	(-1,1)	(1,-1)

You should play this game with the mixed strategy (defined below) $.5H + .5T$. The expected payoff for the game is 0.

5.2 Actions, Strategies, Replies

Distinction between actions and strategies: strategies are plans of action. Strategies described for the Prisoners' Dilemma and Battle of the Brothers are plans to take a single specific action. Another kind of strategy could entail randomization. Randomized strategies play a central role in game theory.

Pure strategy In an n-player finite strategic-form game, a *pure strategy* s_i for Player i is a plan to take some specific action in A_i once the game is played. A *pure-strategy profile* $(s_1, s_2, \ldots, s_n)$ is a choice for each player j of a pure strategy s_j.

Mixed strategy A *mixed strategy* σ_i for Player i is a probability distribution over action set A_i that has the following strategic interpretation: once the game is played, Player i will choose his action randomly from A_i in accordance with the probability distribution σ_i. The support of such a mixed strategy σ_i, written $\text{supp}(\sigma_i)$, is the set of actions in A_i to which σ_i assigns positive probability. A *mixed-strategy profile* $(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is an independent choice for each player j of a mixed strategy σ_j.

Slight abuse of notation, say $u_i(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is the expected payoff to Player i given that the players play mixed-strategy profile $(\sigma_1, \sigma_2, \ldots, \sigma_n)$. Expected payoffs resulting from a mixed-strategy profile are easy to calculate given the independence of the players' mixed-strategy choices.

In an n-player game, use the notation σ_{-i} to denote a choice of mixed strategy for all players other than Player i and $u_i(a_i, \sigma_{-i})$ to denote the expected payoff to Player i when he takes action $a_i \in A_i$ and the other players play according to σ_{-i}. Also use $u_i(\sigma_i, \sigma_{-i})$ to denote the expected payoff to Player i when he plays mixed strategy σ_i and the other players play according to σ_{-i}.

Best-reply action A *best-reply action* for Player i against σ_{-i} is an action $\widehat{a}_i \in A_i$ for Player i that satisfies

$$u_i(\widehat{a}_i, \sigma_{-i}) \geq u_i(a_i, \sigma_{-i}) \quad \forall\, a_i \in A_i$$

Pure-strategy best reply A *pure-strategy best reply* for Player i against σ_{-i} is a pure strategy $\widehat{s_i}$ for Player i of the form "Play $\widehat{a_i}$," where $\widehat{a_i}$ is a best-reply action for Player i against σ_{-i}.

Mixed-strategy best reply A *mixed-strategy best reply* for Player i against σ_{-i} is a mixed strategy $\widehat{\sigma}_i$ for Player i that satisfies

$$u_i(\widehat{\sigma}_i, \sigma_{-i}) \geq u_i(\sigma_i, \sigma_{-i})$$

for every mixed strategy σ_i for Player i.

Take a look at best-reply actions and strategies in the example games we've considered so far.

Prisoners' Dilemma The pure strategy "Play D" is thus the unique pure-strategy best reply for each player to any mixed (or pure) strategy by the other player.

Battle of Brothers If you're Player 1 in the Battle of the Brothers and Player 2 is playing mixed strategy

$$\sigma_2 = qC + (1-q)B$$

- C is the unique best-reply action for you against σ_2 if $q > 1/3$. "Play C" is a pure-strategy best-reply for Player 1 if and only if $q \geq 1/3$.
- B is the unique best-reply action for you if $q < 1/3$. "Play B" is a pure-strategy best reply for Player 1 if and only if $q \leq 1/3$.
- Both C and B are best-reply actions when $q = 1/3$. When $q = 1/3$, the mixed strategy $pC + (1-p)B$ is a best-reply strategy for Player 1 $\forall\, p \in [0, 1]$.

For Player 2, if Player 1 is playing mixed strategy

$$\sigma_1 = pC + (1-p)B$$

- C is the unique best-reply action for you against σ_1 if $p < 2/3$. "Play C" is a pure-strategy best-reply for Player 2 if and only if $p \leq 2/3$.
- B is the unique best-reply action for you if $p > 2/3$. "Play B" is a pure-strategy best reply for Player 2 if and only if $p \geq 2/3$.
- Both C and B are best-reply actions when $p = 2/3$. When $p = 2/3$, the mixed strategy $qC + (1-q)B$ is a best-reply strategy for Player 2 for all $q \in [0, 1]$.

Matching Pennies If you are Player 1 and Player 2 is playing mixed strategy

$$\sigma_2 = qH + (1-q)T$$

- H is the unique best-reply action for you against σ_2 if $q > 1/2$. "Play H" is a pure-strategy best-reply for Player 1 if and only if $q \geq 1/2$.

- T is the unique best-reply action for you if $q < 1/2$. "Play T" is a pure-strategy best reply for Player 1 if and only if $q \leq 1/2$.
- Both H and T are best-reply actions when $q = 1/2$. When $q = 1/2$, the mixed strategy $pH + (1-p)T$ is a best-reply strategy for Player 1 for all $p \in [0, 1]$.

For Player 2, Player 1 is playing mixed strategy

$$\sigma_1 = pH + (1-p)T$$

- H is the unique best-reply action for you against σ_1 if $p < 1/2$. "Play H" is a pure-strategy best-reply for Player 2 if and only if $p \leq 1/2$.
- T is the unique best-reply action for you if $p > 1/2$. "Play T" is a pure-strategy best reply for Player 2 if and only if $p \geq 1/2$.
- Both H and T are best-reply actions when $p = 1/2$. When $p = 1/2$, the mixed strategy $qH + (1-q)T$ is a best-reply strategy for Player 2 for all $q \in [0, 1]$.

By convention, Player 1's mixed strategy depends on p, Player 2's mixed strategy depends on q.

5.3 Nash equilibria

Fact A mixed strategy $\widehat{\sigma}_i$ for Player i is a mixed-strategy best reply against σ_{-i} if and only if every action in $\text{supp}(\widehat{\sigma}_i)$ is a best-reply action for Player i against σ_{-i}.

Pure-strategy Nash equilibrium A pure-strategy profile $s^* = (s_1^*, s_2^*, \ldots, s_n^*)$ in an n-person game is a *pure-strategy Nash equilibrium* or PSNE when, for each i, s_i^* is a pure-strategy best-reply to s_{-i}^*.

Nash equilibrium A mixed-strategy profile $\sigma^* = (\sigma_1^*, \sigma_2^*, \ldots, \sigma_n^*)$ in an n-person game is a *Nash equilibrium* when, for each i, σ_i^* is a mixed-strategy best-reply to σ_{-i}^*.

One useful way to evaluate a pure- or mixed-strategy profile σ^* to see whether it's a Nash equilibrium is to ask yourself the following question(s): if I were Player i, and all the other players announced what strategies they were playing, would I have any incentive to change my strategy? If the answer for every i is no, then σ^* is a Nash equilibrium. If any player has reason to switch, it's back to the drawing board.

Theorem Every finite strategic-form game has at least one Nash equilibrium.

A good way to go about computing all the Nash equilibria is to look first for PSNE and then for mixed-strategy Nash equilibria. Results for example games are as follows.

Prisoners' Dilemma

- PSNE: (D,D)
- MSNE: none

Battle of the Brothers

- PSNE: (C, C) and (B, B)
- MSNE: (σ_1^*, σ_2^*) where $\sigma_1^* = (2/3)C + (1/3)B$, $\sigma_2^* = (1/3)C + (2/3)B$

Matching Pennies

- PSNE: none
- MSNE: (σ_1^*, σ_2^*) where $\sigma_1^* = \sigma_1^* = (1/2)H + (1/2)T$

Hawk Dove Game A new example game. Payoffs are

	H	D
H	(-25,-25)	(50,0)
D	(0,50)	(15,15)

- PSNE: (H,D) and (D,H)
- MSNE: (σ_1^*, σ_2^*), where $\sigma_1^* = \sigma_2^* = (7/12)H + (5/12)D$.

Scissors-Paper-Rock One final example game is this differently named version of Rock-Paper-Scissors. Payoffs are

	S	P	R
S	(1,1)	(2,0)	(0,2)
P	(0,2)	(1,1)	(2,0)
R	(2,0)	(0,2)	(1,1)

- PSNE: none, can check by trying/comparing all possibilities
- MSNE: $\sigma^* = (\sigma_1^*, \sigma_2^*) = ((1/3)S + (1/3)P + (1/3)R, (1/3)S + (1/3)P + (1/3)R)$

Chain Store Game (aka Market Entry), with payoffs as follows

	A	R
E	(2,2)	(-1,-1)
N	(0,4)	(0,4)

Player 2 owns a Home Depot in some market. Player 1 is trying to decide whether to start a Lowe's in the same market. If Player 1 takes action E, he enters (i.e., opens his store); if Player 1 takes action N, he doesn't enter. Player 2 can plan either to accommodate (action A), which would entail sharing the customers in the market with Player 1 if Player 1 enters, or to retaliate (action R) against an entering Player 1 by starting a price war that costs both players. Game has two PSNE (E, A) and (N, R). It also has a bunch of Nash equilibria of the form

$$\sigma^* = (\sigma_1^*, \sigma_2^*) = (N, qA + (1-q)R)$$

with $0 \leq q \leq 1/3$. The equilibria here are sketchy.

Trembling-hand perfect A Nash equilibrium

$$\sigma^* = (\sigma_1^*, \sigma_2^*, \ldots, \sigma_n^*)$$

in an n-player game is *trembling-hand perfect* when $\exists$ a sequence of strategy profiles $\{\sigma^k = (\sigma_1^k, \ldots, \sigma_n^k)\}$ that converges to σ^* as $k \to \infty$, satisfying

$$\text{supp}(\sigma_j^k) = A_j \quad \forall\, j, k$$

and for which σ_i^* is a best reply to σ_{-i}^k for all i.

This is due to Reinhard Selten. His trembling-hand terminology refers to "possible mistakes" by players when they implement their strategies. In a trembling-hand perfect equilibrium, Player i's strategy would also be a best reply for Player i against a mixed-strategy profile for the other players that calls for all of them to play all their available actions with at least small positive probability. It happens that every finite strategic-form game possesses at least one trembling- hand perfect Nash equilibrium.

The only trembling-hand perfect Nash equilibrium of the chain store game is (E,A).

5.4 Iterated Prisoners' Dilemma

In a repeated game, the same agents play the same strategic-form game - called the stage game - over and over again. The stage game in the Iterated Prisoners' Dilemma (IPD) is the Prisoners' Dilemma game.

A technique called backward induction enables humble agents like us to reason about how rational players might play the many-stage IPD. The only Nash-equilibrium outcome of the twice-repeated Prisoners' Dilemma is for both players to play D in both stages.

In a repeated game, we define *outcome* a bit differently than in strategic-form games, namely as a sequence of outcomes of the stage game.

In the IPD with K stages, where $K > 2$, we assume that both players observe and remember the outcomes of all the stages as they occur. A pure strategy for a player in the K-stage IPD consists of quite a complicated plan of action, namely

- a pure-strategy plan of action for the first stage, of which two are possible, and
- for each k, $1 \leq k < K$, a mapping from the set of possible outcomes of stages 1 through k to the set of pure-strategy plans of action for the $(k + 1)$th stage.

For $K = 3$, player has $2 \times 16 \times 2^{16} = 2^{21}$ pure strategies. As K gets larger, the number of pure strategies grows rapidly, to say the least, but backward induction can serve to make analyzing these situations possible. Via backward induction, as in the two-stage IPD, a run of (D,D) outcomes is the only Nash-equilibrium outcome of the K-stage IPD.

Some strategies for IPD.

Define a pure-strategy Nash equilibrium for this game as any strategy profile (s_1^*, s_2^*) that satisfies the following criteria: no strategy s_1 for user 1 gives him higher long-term average payoff than s_1^* when user 2 is playing s_2^*, and no strategy s_2 for user 2 gives him higher long-term average payoff than s_2^* when user 1 is playing s_1^*.

Relentless Punishment: Play C on Day 1 and keep playing C unless the other user plays D on some day, and if that happens play D on every day after that. If both users play the Relentless Punishment strategy, then neither user has an incentive to switch, thus the strategy profile wherein both players play Relentless Punishment is therefore a Nash equilibrium.

Forgiving Punishment:

- **Phase 1:** Play C on Day 1 and keep playing C unless the other user plays D on some day, and if that happens switch to Phase 2.
- **Phase 2:** Play D for two days and return to Phase 1.

Neither player has an incentive not to play the Forgiving Punishment strategy given that the other player is playing it, so the strategy profile wherein both players play Forgiving Punishment is a Nash-equilibrium profile. Like the Relentless Punishment, it results in all (C, C) outcomes when both players stick to it.

6 Evolutionarily Stable Strategy Concepts

John Maynard Smith, a British evolutionary biologist, came up with the ESS idea even before he learned about Nash equilibria.

Setup is a 2-player finite symmetric game. In such a game, each player has the same finite action space A, and we can describe the payoffs using a single function $u : A \times A \to \mathbb{R}$ as follows: for each a and a' in A,

$u(a, a')$ is the payoff to a player against a' player. Thus if $A = \{a_1, \ldots, a_m\}$ and

$$\sigma = p_1 a_1 + p_2 a_2 + \cdots + p_m a_m$$

is a mixed strategy for either player, the expected payoff to the other player when he plays action a against σ is

$$u(a, \sigma) = p_1 u(a, a_1) + p_2 u(a, a_2) + \cdots + p_m u(a, a_m)$$

If that other player instead plays a mixed strategy

$$\sigma' = q_1 a_1 + q_2 a_2 + \cdots + q_m a_m$$

against σ, he gets expected payoff

$$u(\sigma', \sigma) = \sum_{i=1}^{m} \sum_{j=1}^{m} q_i p_j u(a_i, a_j)$$

Maynard Smith's scenario is as follows. Every agent in large population of agents is playing the same pure or mixed strategy σ^* from a finite symmetric game. The agents meet at random in "pairwise matches" with each other to play the game. Call σ^* the *incumbent strategy.* Suppose that, somehow, a small fraction of players of some other strategy σ appears in population. The new population, the *post-entry population*, has a small fraction ϵ of σ-players and a large fraction $(1-\epsilon)$ of σ^*-players. The strategy σ is called the *invading strategy.* If a σ^*-player plays a random match with a member of the post entry population, its expected payoff will therefore be

$$(1-\epsilon)u(\sigma^*, \sigma^*) + \epsilon u(\sigma^*, \sigma)$$

This is the same as the payoff to a σ^*-player against a player mixing via $(1-\epsilon)\sigma^* + \epsilon\sigma$, i.e.

$$(1-\epsilon)u(\sigma^*, \sigma^*) + \epsilon u(\sigma^*, \sigma) = u(\sigma^*, (1-\epsilon)\sigma^* + \epsilon\sigma)$$

Similarly, if a σ-player plays such a random match against a randomly chosen opponent from the post-entry population, its expected payoff will be

$$(1-\epsilon)u(\sigma, \sigma^*) + \epsilon u(\sigma, \sigma) = u(\sigma, (1-\epsilon)\sigma^* + \epsilon\sigma)$$

Evolutionarily stable strategy In the context above, σ^* is said to be an *evolutionarily stable strategy* (ESS) of the original finite symmetric game when for every strategy $\sigma \neq \sigma^*$ $\exists$ some $\bar{\epsilon} > 0$ such that for every $\epsilon < \bar{\epsilon}$ we have

$$u(\sigma^*, (1-\epsilon)\sigma^* + \epsilon\sigma) > u(\sigma, (1-\epsilon)\sigma^* + \epsilon\sigma)$$

The point is that for an arbitrary invading strategy σ, an ESS "does strictly better" in the post-entry population than σ does provided the fraction of invaders is small enough. The ESS concept is static rather than dynamic. Maynard Smith didn't account for how invaders might appear and/or get wiped out.

Basic Fact σ^* is an ESS iff both of the following conditions hold:

- (σ^*, σ^*) is a Nash equilibrium of the original game - i.e. σ^* is a best reply to itself.
- If σ is an alternative best reply to σ^* in the original game, then

  $$u(\sigma^*, \sigma) > u(\sigma, \sigma)$$

  I.e. σ^* does better against σ than σ does against itself.

The SPR game (as specified above) has no ESS. In general, while a finite symmetric game has at least one Nash equilibrium (and in fact at least one symmetric Nash equilibrium of the form (σ^*, σ^*)), it need not have an ESS.

Another consequence of the Basic Fact is that if (σ^*, σ^*) is a strict Nash equilibrium, then σ^* is an ESS. That's because a strict Nash equilibrium has no alternative best replies, so we never need to check the second condition in the Basic Fact.

The pure strategies “play D” in the Prisoners’ Dilemma and “play C” and “play B” in the symmetric version of the Battle of the Sexes/Brothers are all ESS. Three-stage IPD has no ESS.

Although the general three-stage IPD has no ESS, we can focus on four specific pure strategies for the game and analyze the ESS outcomes. Four strategies are

- RP: play C on the first stage and continue playing C until your opponent plays D, after which play D for the rest of the stages. (RP stands for relentless punishment.)
- RPD: play as in RP except always play D on the last stage.
- CD^2: play C on the first stage and D on the last two under all circumstances.
- D^3: play D on every stage under all circumstances.

There are many ways to mix and match the strategies but we will just focus on a few. First symmetric game that arises is

	RP	D^3
RP	(9,9)	(2,7)
D^3	(7,2)	(3,3)

Both (RP, RP) and (D^3, D^3) are strict Nash equilibria of this game and hence both RP and D^3 are evolutionarily stable strategies.

What makes any kind of equilibrium cooperation impossible in the three-stage IPD is the fact that a player’s best move on the final stage is always to play D.

Next game looks like

	RP	RPD
RP	(9,9)	(6,11)
RPD	(11,6)	(7,7)

This has only one Nash equilibrium, namely (RPD,RPD), which is strict and RPD is therefore an ESS.

Next game looks like

	RP	RPD	D^3
RP	(9,9)	(6,11)	(2,7)
RPD	(11,6)	(7,7)	(2,7)
D^3	(7,2)	(7,2)	(3,3)

(RPD, RPD) is still a Nash equilibrium but RPD is no longer an ESS. To see why, note that D^3 is an alternative best reply to RPD and $2 = u(RPD, D^3) < u(D^3, D^3) = 3$. $(D^3, D^3$ is a strict Nash equilibrium, so D^3 is an ESS.

Last we have

	RP	RPD	CD^2	D^3
RP	(9,9)	(6,11)	(4,9)	(2,7)
RPD	(11,6)	(7,7)	(4,9)	(2,7)
CD^2	(9,4)	(9,4)	(5,5)	(2,7)
D^3	(7,2)	(7,2)	(7,2)	(3,3)

The only Nash equilibrium is (D^3, D^3), which is strict, making D^3 an ESS. (Your fitness depends on who else you are playing against).

7 Replicator Dynamics

Replicator Dynamics are one of several examples of deterministic dynamical approaches to evolutionary game theory (EGT). The approach entails modeling complicated inherently random processes with deterministic nonlinear dynamical systems. The well developed machinery of dynamical-systems theory makes it possible to analyze them rigorously, regardless of whether or not the resulting models are faithful or effective.

Setup Like above in ESS, the setup is a 2-player finite symmetric game. In such a game, each player has the same finite action space $A = \{a_1, a_2, \dots, a_m\}$, and we can describe the payoffs using a single function $u : A \times A \to \mathbb{R}$ as follows: the payoff to a player playing a_i against an opponent playing a_j is $u(a_i, a_j)$. If

$$\sigma = p_1 a_1 + p_2 a_2 + \cdots + p_m a_m$$

is a mixed strategy for either player, the expected payoff to the other player when he plays action a_i against σ is

$$u(a_i, \sigma) = p_1 u(a_i, a_1) + p_2 u(a_i, a_2) + \cdots + p_m u(a_i, a_m)$$

If that other player instead plays a mixed strategy

$$\sigma' = q_1 a_1 + q_2 a_2 + \cdots + q_m a_m$$

against σ, he gets expected payoff

$$u(\sigma', \sigma) = \sum_{i=1}^{m} \sum_{j=1}^{m} q_i p_j u(a_i, a_j)$$

We'll be considering large populations of agents that play the game with each other. Each agent is hard-wired to play one of the actions in A every time it plays the game against another agent. The agents are not rational. Can think of the agent populations as multi-sets of actions from A rather than as populations of agents.

First step is to add the same positive constant to every payoff in the game being analyzed to arrive at a game whose payoffs are strictly positive. This won't change Nash equilibria. For the Hawk-Dove game, we can transform the payoffs by adding 26 to each to get

	H	D
H	(1,1)	(76,26)
D	(26,76)	(41,41)

Assume (for now) the payoffs are integer-valued. The population at some integer time $t \geq 0$, contains a large number of agents each of which is programmed to play one of the actions in A. The population at time $t+1$ arises from the population at time t as follows:

(i) Each agent i in time t population plays the game with an opponent chosen at random from the time t population.

(ii) The winning agent from (i) clones a number of copies of itself proportional (with some positive integer proportionality constant K) to its payoff in the game from (i); these copies become members of the next population. The proportionality constant K is the same for all agents.

(iii) After every agent in the current population has completed (i) and (ii), the agents in the current population disappear and their clones from (ii) constitute the population at time $t+1$.

For each i, suppose the population at time t contains $N_i(t)$ agents programmed to play action a_i. Entire population consists of $N(t) = \sum_{i=1}^{n} N_i(t)$ agents. For each i, let $p_i(t) = \frac{N_i(t)}{N(t)}$ be the fraction of agents in the population playing a_i.

In step (i), every agent playing a_i picks a random opponent to play the game with. The probability that the chosen agent will be an a_j-player is $p_j(t)$. The expected payoff to the a_i-player is therefore

$$\sum_{i=1}^{n} p_j(t)u(a_i, a_j) = u(a_i, p(t))$$

where $p(t)$ is shorthand to represent the mixed strategy

$$p_1(t)a_1 + p_2(t)a_2 + \cdots + p_n(t)a_n$$

No one is actually playing this mixed strategy, some people call $p(t)$ the *population strategy* at time t.

Delchamp's standard form of the discrete-time replicator dynamics (using $u(p(t), p(t))$ to represent the payoff in a game to a player playing the population strategy $p(t)$ against an opponent also playing $p(t)$) is as follows

$$p_i(t+1) = \frac{u(a_i, p(t))}{u(p(t), p(t))} p_i(t), \quad 1 \le i \le n$$

Some comments on features of the equation

- Think of the discrete-time dynamical system governed by the equation as having the n-vector whose ith entry is $p_i(t)$ as its state at time t. In other words, the equation describes a discrete-time dynamical system whose state lives in $\mathbb{R}^n$.
- If a_i gains a higher-than-average payoff against the population strategy $p(t)$, then $p_i(t+1) > p_i(t)$ — that is, the population at time $t+1$ contains a larger fraction of a_i-players than the population at time t. Similarly, if a_i gains a lower-than-average payoff against $p(t)$, the population at time $t+1$ contains a smaller fraction of a_i-players than the population at time t.
- If for some i we have $p_i(0) = 0$,then we'll have $p_i(t) = 0$ for all $t > 0$. Thus if an action is absent from the population at time 0, it never appears.
- If (σ^*, σ^*) is a Nash equilibrium of the original game, where

  $$\sigma^* = p_1^* a_1 + p_2^* a_2 + \cdots + p_n^* a_n$$

  then the n-vector p^* whose ith entry is p_i^* is a fixed point of the discrete time dynamical system governed by the equation. This is another way of saying that if the population strategy at time 0 is σ^*, the population strategy at every time $t > 0$ is σ^*.
- The equation in general has fixed points other than the Nash equilibria of the original game. Any monomorphic population — that is, any population composed solely of a_i-players for some specific i — has $p_i = 1$ and $p_j = 0$ for all other j. If you start with such a population at time 0, you'll have such a population for every time $t > 0$.

One form of continuous-time replicator dynamics where D denotes derivative with respect to t,

$$Dp_i(t) = \frac{u(a_i, p(t)) - u(p(t), p(t))}{u(p(t), p(t))} p_i(t), \quad 1 \le i \le n$$

Standard form of continuous-time replicator dynamics

$$Dp_i(t) = (u(a_i, p(t)) - u(p(t), p(t)))p_i(t), \quad 1 \le i \le n$$

Both have the same fixed points and trajectories in p-space, they have the same *phase portrait.* But they do lead to different time-parameterizations of trajectories of the phase portrait.

If you're looking to analyze the continuous-time replicator dynamics for some game, you need not do any positive-constant-adding beforehand.