UPPSALA
UNIVERSITET

U.U.D.M. Project Report 2013:25

Evolutionary Language Games

Linnéa Gyllingberg

Examensarbete i matematik, 15 hp
Handledare och examinator: David Sumpter
Augusti 2013

Department of Mathematics
Uppsala University

Abstract

In the late 1940s game theory was invented. Originally the mathematical theory was used to describe economical behaviour, but in the 1970s game theory was applied to study evolutionary biology. During the last two decades a mathematical framework describing the evolution of language has been developed. We investigate how the mathematical theories from game theory and evolutionary dynamics can be used to describe the evolution of vocabulary.

Contents

1 Introduction

For billions of years information has been transferred genetically from one individual to another, from one generation to the next. Evolutionary biology is the study of how this information transfer changes over time and gives rise to new species [7, pp. 249-252]. For a long time, the primary information transfer was genetic. About one million years ago human language started to evolve [7, p. 250]. Information could now be transferred by language and this gave rise to a new type of evolution: cultural evolution. How did humans evolve the ability to speak? Why did humans start to talk? What gave rise to the complex features of human language?

To answer these questions several aspects of the evolution of language must be studied. One way to try to answer these questions is by mathematical models. Theories to explain the evolution of language should explore three different aspects [9]. Firstly they should try to explain the evolution of the simplest communication system. Secondly they should show how natural selection leads to the transition from animal communication to human language by explaining the evolution of the simplest properties that distinguish human language from animal communication. And thirdly they should explain how natural selection leads to the complex features of modern human language. This thesis deals with the first aspect.

2 Evolutionary Game Theory

2.1 What is Evolution?

Ever since Charles Darwin published *On the Origin of Species* in 1859, the theory of evolution has been one of the most fundamental theories in science. Darwin gave a simple scientific explanation for all the diversity in nature.

Some parts have of course been modified due to new discoveries, but the basic ideas of Darwin still form the foundation of the theory of evolution today [1, p. 15]. The theory of evolution spans over all fields of biology. All biological systems can be interpreted in an evolutionary context.

All evolution requires three basic properties: reproduction, selection and mutation [7, p. 18]. Evolution needs populations of individuals that reproduce. Reproduction is not perfect and mistakes occur during the reproduction. These mistakes are called mutations. Mutations give rise to different types of individuals; variations arise. Some types will reproduce faster than others and selection will lead to the biological diversity we see in nature.

2.2 What is Game Theory?

Game theory is a field of mathematics invented in the late 1940s by the mathematician John von Neumann and the economist Oskar Morgenstein [3, pp. xiii-xv]. It is the study of strategic decision making. The basic question to answer is: Which strategy should I use to maximize my payoff? In game theory, a game is usually represented by a matrix which shows the possible strategies for the players and the payoff they get by playing the strategies. A simple, but yet very interesting, example of a game is the two-person two-strategies game The Prisoner's Dilemma. The Prisoner's Dilemma is studied in various fields: economics, political sciences and philosophy are just some examples.

The prisoner's dilemma can be formulated in the following way: Two criminals are suspected of committed a joint crime. They are both arrested and imprisoned and there is no possibility for them to talk to each other. The police does not have enough evidence to convince a jury. The prisoners are left with two choices: either they can remain silent or they can confess the crime and testify against their partner. If both confess, then both of them are sentenced to one year in prison. If one confesses and testifies against the other, while the other remains silent, the confessing prisoner is set free and the other one is sentenced to ten years in prison. If they both remain silent, they serve seven years in prison.

This can be represented with the payoff matrix in Figure 1.

Payoff Matrix

		Prisoner B	
		Remain silent	Confess
Prisoner A	Remain silent	-1 -1	-10 0
	Confess	0 -10	-7 -7

Figure 1: Payoff matrix for The Prisoner's Dilemma

2.2.1 Nash Equilibrium

An important idea in game theory is the concept of Nash equilibrium that was developed by the American mathematician John Nash in the early 1950s [7, pp. 51-53]. A Nash equilibrium can be described by the following example: Consider a two player game. If both players play a strategy that is a Nash equilibrium, then neither of the players can receive a higher payoff by changing only his own strategy.

Now consider The Prisoner's Dilemma and the payoff matrix in Figure 1. What is the Nash equilibrium? If both remain silent, each player can improve their own payoff by confessing. If both confess, then none of them can increase their payoff by switching strategy. Hence, both players confessing is a Nash equilibrium. There are various types of games in game theory, zero-sum and non-zero-sum games, combinatorial games and differential games are just some examples. Depending on the type of game, there are different formal definitions for a Nash equilibrium.

2.3 Evolutionary Game Theory

Originally game theory was used to describe economic behaviour, but in the 1970s game theory was applied on biology by John Maynard Smith and George Price [7, p. 46] to describe evolutionary biology. When applied to economics, business and political sciences, game theory is the study of mathematical models that describe strategies by intelligent rational decision-makers. Evolutionary biology does not require rational game players, instead it considers a population of individuals with fixed strategies interacting in a game [7, pp. 46-70]. The strategies can denote different types of features or qualities.

The simplest way to describe an evolutionary strategy is by a game with two strategies, A and B, in a population. Let x_A denote the frequency of individuals playing strategy A, and x_B denote the frequency of individuals playing strategy B. Then $\vec{x} = (x_A, x_B)$ defines the composition of the population. By denoting the fitness of A and B $f_A(\vec{x})$ and $f_B(\vec{x})$ respectively, the dynamics of the population can be described as

$$\begin{aligned} \dot{x}_A &= x_A(f_A(\vec{x}) - (x_A f_A(\vec{x}) + x_B f_B(\vec{x}))) \\ \dot{x}_B &= x_B(f_B(\vec{x}) - (x_A f_A(\vec{x}) + x_B f_B(\vec{x}))) \end{aligned} \tag{1}$$

Since x_A and x_B denote the frequency of the individuals using strategy A and B respectively, $x_A + x_B = 1$. Introducing the new variable x with $x_A = x$ and $x_B = 1 - x$, the system (1) can be rewritten as to

$$\dot{x} = x(1-x)(f_A(\vec{x}) - f_B(\vec{x})) \tag{2}$$

A game with two strategies is usually described by a payoff matrix:

$$\begin{array}{cc} & \begin{array}{cc} A & B \end{array} \\ \begin{array}{c} A \\ B \end{array} & \begin{pmatrix} a & b \\ c & d \end{pmatrix} \end{array} \tag{3}$$

The payoff matrix is not read in the same way as the payoff matrix in Figure 1. The meaning of the payoff matrix is that an A player gets payoff a when playing with another A player, and payoff b playing against a B player. A B player gets payoff c and d, playing against an A player and a B player, respectively. From this payoff matrix we get that the fitness for the different types of individuals is given by

$$\begin{aligned} f_A &= ax_A + bx_b \\ f_B &= cx_A + dx_B \end{aligned} \tag{4}$$

By using the variable change $x_A = x$ and $x_B = 1 - x$ we get

$$\begin{aligned} f_A &= ax + b(1-x) = (a-b)x + b \\ f_B &= cx + d(1-x) = (c-d)x - d \end{aligned} \tag{5}$$

Inserting the fitness functions into equation (2) we get the dynamics to be

$$\begin{aligned} \dot{x} &= x(1-x)(f_A - f_B) = x(1-x)(((a-b)x + b) - ((c-d)x - d) \\ &= x(1-x)((a-b-c+d)x + b - d) \end{aligned} \tag{6}$$

The equilibrium points of this differential equation is $x^* = 0$, $x^* = 1$ and $x^* = (d-b)/(a-b-c+d)$.

Since x denote the frequency of A players in the population, $x \in [0, 1]$. For some values of a, b, c and d the third equilibrium point, $x^* = (d-b)/(a-b-c+d)$, will not be in the interior of $[0, 1]$. For the inequality $0 < \frac{d-b}{a-b-c+d} < 1$ to be satisfied, either $a > c$ and $b < d$, or $a < c$ and $b > d$.

To determine the stability of the steady states, we start with evaluating the Jacobian of equation (6) for the different steady states:

$$J = \frac{d\dot{x}}{dx} = \frac{d}{dx}(x(1-x)((a-b-c+d)x + b - d) \tag{7}$$

$$= 2x(a-b-c+d) + (b-d) - 3x^2(a-b-c+d) - 2x(b-d) \tag{8}$$

An equilibrium point, x^* is stable if $J|_{x=x^*} < 0$ and unstable if $J|_{x=x^*} > 0$. For $x^* = 0$ we have:

$$J|_{x=0} = b - d \tag{9}$$

So $x^* = 0$ is stable when $b > d$ and unstable when $b < d$. The Jacobian of the fixed point $x^* = 1$ is:

$$J|_{x=1} = 2(a - b - c + d) + (b - d) - 3(a - b - c + d) - 2(b - d) = -a + c \quad (10)$$

Hence $x^* = 1$ is stable when $a > c$ and unstable when $a < c$. For $x^* = (d - b)/(a - b - c + d)$ we have:

$$J|_{x=\frac{d-b}{a-b-c+d}} = \frac{(a-c)(d-b)}{a-b-c+d} \quad (11)$$

When $a < c$ and $b > d$, $x^* = \frac{d-b}{a-b-c+d}$ is stable and when $a > c$ and $b < d$, it is unstable. If $a = c$ and $b = d$ then the differential equation (6) is $\dot{x} = 0$, and hence the frequency of A players and the frequency of B players is always constant.

Depending on the parameters a, b, c and d there are five possible cases for the steady states of equation (6):

(i) If $a > c$ and $b > d$ there are only two equilibrium points $x^* = 1$ and $x^* = 0$. $x^* = 1$ is the only stable point and hence all the B players of the population will become extinct.

(ii) If $a < c$ and $b < d$ there are also just two equilibrium points, $x^* = 1$ and $x^* = 0$, as in (1), but $x^* = 0$ is always stable. No matter how small frequency of B players there is in the beginning, A players will always become extinct.

(iii) If $a > c$ and $b < d$, then $x^* = (d - b)/(a - b - c + d) \in [0, 1]$ and will be unstable. Both $x^* = 1$ and $x^* = 0$ will be stable. The outcome depends on the initial condition.

(iv) If $a < c$ and $b > d$, then $x^* = 0$ and $x^* = 1$ will be unstable. $x^* = (d - b)/(a - b - c + d)$ will be in the interior of $[0, 1]$ and will be stable. Hence A players and B players will coexist.

(v) If $a = c$ and $b = c$ the frequency of A players and B players is always constant.

Now consider the concept of Nash equilibrium. For the strategies of the evolutionary game described by matrix (3) we have:

(i) A is a strict Nash equilibrium if $a > c$

(ii) A is a Nash equilibrium if $a \geq c$

(iii) B is a strict Nash equilibrium if $d > c$

(iv) B is a Nash equilibrium if $d \geq b$

2.3.1 Evolutionarily Stable Strategies

Unaware of the Nash equilibrium in game theory, the British biologist John Maynard Smith invented the concept of evolutionarily stable strategy (ESS) [7, pp. 53-55]. Consider a large population of A players. A single B player is introduced in the population. The game between A and B is again given by the payoff matrix (3). The question which the concept of ESS answers is the following: What are the conditions to oppose an invasion of B players into the population of A players? The answer is given by the conditions that make the equilibrium point $x^* = 1$ from equation (6) stable. In this particular evolutionary game, the concept of ESS can also be derived in the following way:

Assume that there is an infinitesimally small fraction of B players in the population. The frequency of B will be ϵ and the frequency of A will be $1 - \epsilon$. The fitnesses for A and B is given by equation (6). If A will resist an invasion of B, the fitness of A must be greater then the fitness of B.

$$a(1-\epsilon) + b\epsilon > c(1-\epsilon) + d\epsilon \tag{12}$$

After cancelling the ϵ terms, the inequality leads to

$$a > c \tag{13}$$

But if $a = c$ the inequality (12) leads to

$$b > d \tag{14}$$

So for the strategy A to be ESS either $a > c$ or $a = c$ and $b > d$.

For a general evolutionary game a strategy is ESS if it is a strict Nash equilibrium and it is a Nash equilibrium if it is ESS [7, p. 55]:

$$\text{Strict Nash} \Rightarrow \text{ESS} \Rightarrow \text{Nash}$$

3 The Language Game

In the article "The Evolutionary Language Game", Martin Nowak, Joshua Plotkin and David Krakauer investigate how evolutionary game dynamics must be modified to be able to describe evolution of language [6]. They study the evolution of the simplest language system by exploring how signals can evolve to become associated with different objects and compare different learning strategies. I have reproduced part of their work and studied the language dynamics for some special cases.

3.1 Model

We consider a group of individuals. Information can be transferred about n objects. An object can be an animal, an event, a person or anything else that can be referred to. There are m possible signals for describing these n objects. For each individual in the population there is an $n \times m$ probability matrix P and an $m \times n$ probability matrix Q. P, also called the active matrix, has the entries p_{ij}, which represents the probability that a speaker associates object i with sound j. For the active matrix, rows represent objects and columns represent sounds. An individual will always produce a signal when seeing an object which means that all rows of P will sum up to one.

The passive matrix Q has entries q_{ji} that denotes the probability of a listener to think of object i when hearing sound j. An individual will always think of an object when hearing a signal, leading to the rows of Q, just like P, sums to 1. Hence we have

$$\sum_{j=1}^{m} p_{ij} = 1 \qquad \text{and} \qquad \sum_{i=1}^{n} q_{ji} = 1$$

3.1.1 Communication Payoff

Each individual I_i has a language L_i given by P_i and Q_i. For two individuals I_1 and I_2 with languages L_1 and L_2, $p_{ij}^{(1)}$ denotes the probability for I_1 to make sound i when seeing an object j whereas $q_{ji}^{(1)}$ denotes the probability for I_1 to infer object i when hearing sound j. For individual I_2 these probabilities are given by $p_{ij}^{(2)}$ and $q_{ji}^{(2)}$. The probability for individual I_1 to communicate object i to individual I_2 is given by $\sum_j p_{ij}^{(1)} q_{ji}^{(2)}$. The total communication payoff between the two individuals sums these probabilities for all objects and then takes the average of the reversed situation. This leads to a communication payoff described by

$$F(L_1, L_2) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}^{(1)} q_{ji}^{(2)} + p_{ij}^{(2)} q_{ji}^{(1)} \tag{15}$$

3.1.2 Maximum Payoff

Since the equation evaluates both ability of I_1 to transfer information to I_2 and vice versa, there is a symmetry of the language game: $F(L_1, L_2) = F(L_2, L_1)$. The maximum payoff is obtained when two individuals speak the same language. For languages where $n = m$, the maximum payoff is $F_{\max}(L, L) = n$.

The maximum payoff is obtained when P is a permutation matrix and $Q = P^T$. A permutation matrix is a binary matrix with exactly one 1 in each row and in each column.

3.1.3 Learning a Language

As in real life, this model assumes that individuals learn their language by listening to other individuals and then imitating them. Every individual undergoes a learning phase. During the learning phase it constructs an association matrix, A. A is an $n \times m$ matrix with entries a_{ij} which determines how often an individual has heard other individuals referring to object i with the sound j. Assume that an individual, I_2, is learning from another individual, I_1. The association matrix is entirely constructed from the active matrix of I_1; P_1. How I_1 understands a language, Q_1, is not considered. In real life, language learning is a complex process that continues through life. In this model the learning phase is considered to last k times, meaning that I_2 samples the responses of I_1 to each object k times before developing her own language. From the association matrix, the active matrix and the passive is then constructed by normalizing rows and columns respectively:

$$p_{ij} = \frac{a_{ij}}{\sum_{l=1}^{m} a_{il}} \tag{16}$$

$$q_{ji} = \frac{a_{ij}}{\sum_{l=1}^{n} a_{lj}} \tag{17}$$

If I_1 never uses signal j for any word, the j:th column of the association matrix of I_2 will sum up to 0. Hence some elements of the passive matrix of I_2 will be undefined, if Q is constructed as described in equation (16). The teacher not using a signal j should lead to the learner associating all objects with the same probability when hearing signal j:

$$q_{j1} = q_{j2} = ... = q_{jn} = \frac{1}{n} \quad \text{when} \quad \sum_{l=1}^{n} a_{lj} = 0 \tag{18}$$

3.1.4 Population Dynamics

Now consider a group of N individuals with languages L_1 to L_N. Each individual talks to every other individual. The total payoff for each individual is obtained by summarizing the communication payoff and dividing it with the number of individuals they talk to:

$$F_I = \frac{\sum_J F(L_I, L_J)}{N - 1} \tag{19}$$

where $J \in \{1, ..., N\} \setminus \{I\}$.

There are different learning strategies for newborn individuals. They can learn from their parents, they can learn from role models in the population or they can learn from random individuals in the previous generation. Depending on the learning strategies of the population, the payoff is used in different ways. Regardless of the learning strategy, there are discrete generations - children always learn their language from the previous generation and there is always the same number of individuals in the next generation as in the previous.

When considering parental learning, communication payoff is interpreted as biological fitness. Individuals with high payoff will produce more offspring than individuals with lower payoff. The probability that a particular individual from the next generation has individual I as their parent is given by

$$\frac{F_I}{\sum_J F_J}$$

where $J \in \{1, ..., N\}$.

3.2 Implementation

To implement this model I used Matlab. The code is presented in Appendix A.

Initialization: Create an initial population, $G(0)$ with N individuals. Each individual i has a $n \times n$ $P(i)$ matrix and a $n \times n$ $Q(i)$ matrix. The P and Q matrices are either randomly generated or they can be stated in the beginning.

Iteration:

1. Evaluate the communication payoff for each individual for population $G(i)$ as described by equation (19). Store the fitness.

2. Create a new generation, $G(i+1)$ with N individuals. Create an association matrix A, an active matrix P and a passive matrix Q for each individual. All are zero matrices.

3. Each individual of generation $G(i+1)$ chooses a parent from generation $G(i)$ selected by their communication payoff. The probability of individual I in generation $G(i)$ being selected as parent for an individual in generation $G(i+1)$ is $\frac{F_I}{\sum_J F_J}$.

4. Each individual from $G(i+1)$ learns from her parent. The learner listens to her parent k times and constructs their association matrix A as described in section 3.1.3.

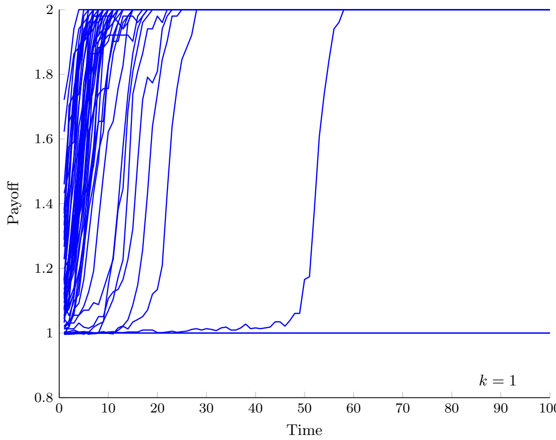
5. Construct the active and passive matrices for each individual from their association matrices according to equations (16). If any row of A sums up to 0, then Q is constructed according to (18).

6. Replace population $G(i)$ with population $G(i+1)$. Return to 1.

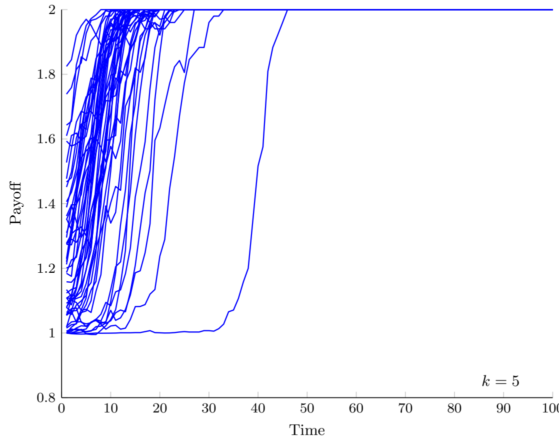
The average payoff for each generation is stored and then the communication payoff is plotted against generations.

3.3 Simulations and Analysis

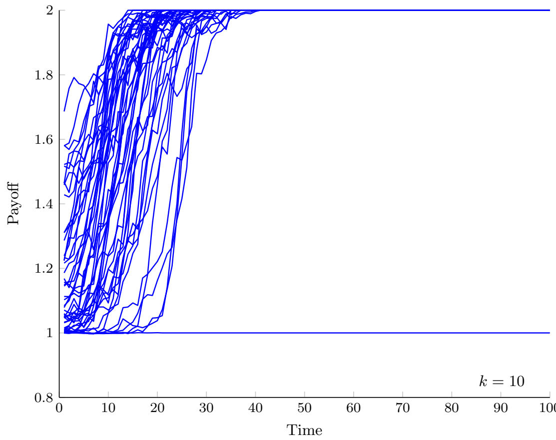
3.3.1 Two Objects and Two Signals



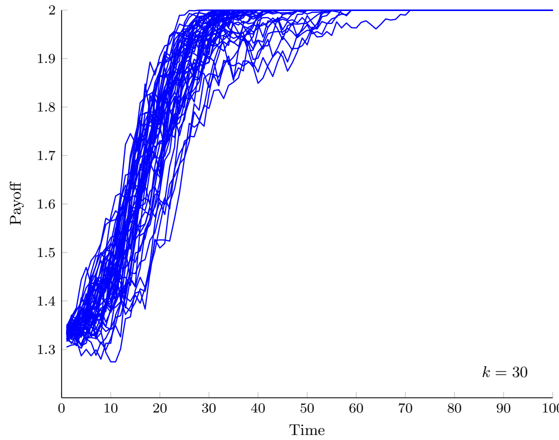
(a) $k = 1$



(b) $k = 5$



(c) $k = 10$



(d) $k = 30$

Figure 2: Simulation of parental learning for a 2-object 2-signal game. There are $N = 100$ individuals in population and 50 simulations.

I started the simulations with looking at a two object two signal game. Both P and Q are 2×2 matrices. In some simulations they were randomly generated in the beginning, in other simulations the first P and Q matrices were given. In Figure 2 parental learning for four different values of k is simulated 50 times for each value of k. The average payoff for each generation is plotted against time. The maximum payoff for the two object two signal game is 2, as described in section (3.1.2). When all the individuals of a population speak the same language with the maximum payoff, there is no ambiguity in the language. Every object can be described by exactly one word and every word describe exactly one object. In Figure 2a the maximal payoff was in average reached only after 10 generations. With $k = 5$, the convergence to a common language is slower; it takes about 20 generation in average before the total communication payoff is 2. The larger k is, the slower the convergence is to a common language. Since the individuals with the high payoff are more likely to have children than individuals with low payoff, selection will lead to a population with a better language. But in some of the simulations, the payoff stays at 1. This is the case when $k = 1$ and $k = 10$, but not when $k = 5$ and $k = 30$.

All simulations in this thesis are stochastic processes with absorbing states. An absorbing state is a state, once being entered, it cannot be left. For this model, all absorbing states will be languages where P is a binary matrix. This is because of the learning process:

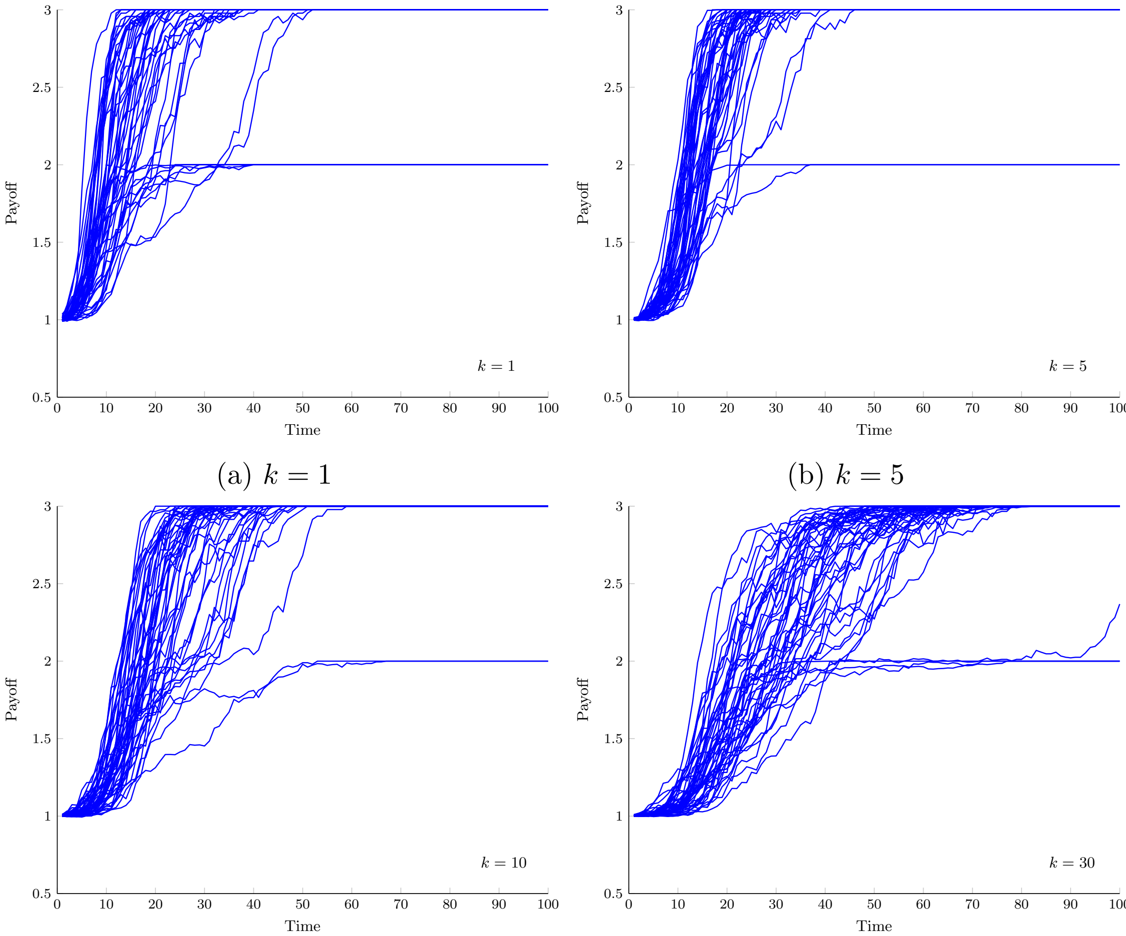
When an individual I_1 has a binary matrix P_1, the association matrix for an individual I_2 learning from I_1, will always lead to a binary active matrix. For example, consider a two object two signal game where I_1 has an active matrix $P_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ The association matrix for an individual learning from I_1 by listening k times will look like $A_2 = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix}$ and hence lead to the active matrix $P_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

All individuals in a population speaking the same language with payoff 1, implies that they all have an active matrix with two ones in one column: $P = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ or $P = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$. These matrices are binary and hence they are also absorbing states.

When all individuals in a population speak the same language with payoff 1, selection will not result in a better language. Why is then the convergence to a common language slower when k increases?

When $k = 1$ the association matrix of the learning individual I_2 will always be binary. A binary association matrix A_2 will lead to a active matrix $P_2 = A_2$. As k increases, the probability for P_2 being binary decreases. When $k \to \infty$, P_1 and P_2 will be exactly the same matrices. As all absorbing states are languages with binary matrices, as stated before, the smaller k is, the faster a common language will evolve. Does the value of k affect the number of simulations ending up in a suboptimal absorbing state? This is hard to determine from the simulations in Figure 2. An answer to this question can be seen in the simulations of a three object and three signal game for different values of k.

3.3.2 Three Objects and Three Signals



(a) $k = 1$ (b) $k = 5$

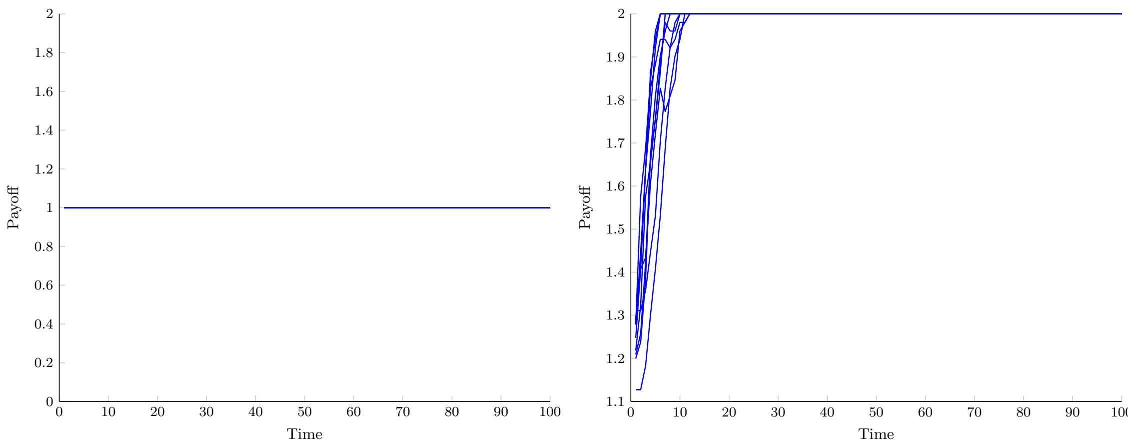
(c) $k = 10$ (d) $k = 30$

Figure 3: Simulation of parental learning for a 3-object 3-signal game. There are $N = 100$ individuals in population and 50 simulations.

In the three object three signal language game $n = m = 3$, so the active and passive matrices are 3×3 matrices. Figure 3 shows simulations for this game with different values of k. As in the previous simulations for the two object two signal game, for each value of k there are 50 runs. The maximum payoff in the three object three signal game is of course 3. The absorbing states are languages with payoff 1, 2 or 3. All runs either end up with fitness 2 or with fitness 3. A language with fitness 3 is an unambiguous language: P and Q are permutation matrices. As for the two object two signal game, the lower value of k, the faster the language tends to develop into an absorbing state. What also can be noticed in Figure 3 is that there is a small difference in the number of runs that end up in a suboptimal absorbing state depending on the value of k. For a small value of k, more runs seem to end up in a suboptimal absorbing state. To explain why, we have to consider the learning process again. When $k = 1$, the association matrix of a learning individual will always be binary, as explained earlier. Even though the active matrix of a teaching individual might be asymmetric and favour an unambiguous language, the probability for the learning individual to have an active matrix giving a suboptimal language is high when $k = 1$. For example, consider an individual I_2 in the two object two signal game learning from individual I_1. I_1 has an active matrix $P_1 = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}$. The probability for I_2 active matrix to look like $P_2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ is quite low, but it is much higher when $k = 1$ than when $k = 30$.

3.3.3 Homonymy and Synonymy

Synonymy is when a word has the same meaning as another word. Homonymy is when a word has several meanings. In linguistics, it is generally understood that synonyms are rare [6]. Homonyms are on the other hand frequent. It is easy to find examples of homonyms, but hard to find examples of synonyms. Can the evolutionary language game provide an explanation to this? In the context of the language game, synonymy is when a object can be referred to by several signals and homonymy is when a signal can be used to describe several objects.



(a) Homonymy: $P = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$

(b) Synonymy: $P = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$

Figure 4: Simulation of parental learning for a two object two signal game with initial conditions for homonymy and synonymy.

Homonomy is represented by a binary matrix with more than one 1 in a column. Figure 4 shows simulations of homonymy and synonymy in the two object and two signal game. In Figure 4a all individuals in the population started with an active matrix $P = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ and a passive matrix $Q = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$. In all simulations $k = 5$. There were ten runs and all runs ended with all individuals in the last generation having exactly the same active and passive matrices as the individuals in the first generation. This is not surprising at all. Homonymy is represented by a binary matrix, and as explained previously a language with an active matrix being binary is an absorbing state.

Synonymy refers to the case where there is more than one non-zero entry in a column. In Figure 4b all individuals had a language containing two synonyms in the beginning: $P = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$ and $Q = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$. Both the objects could be referred to by both the words. After just 10 generations, the synonyms were gone and the population spoke an unambiguous language with either $P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ or $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Synonymy seems to be unstable. To understand why, consider the following active matrix in the one object two signal game: $P = (0.5 \quad 0.5)$. Here one object can be referred to by two sounds. This situation is not stable. When $k = 1$, the next generation will have a binary active matrix. For $k > 1$, the next generation will not necessarily have binary matrices, but binomial sampling will lead to the new active matrices being more asymmetric than the former. Which eventually, will lead to binary matrices.

3.3.4 Evolutionarily Stable Languages

We have seen that all languages which are absorbing states have binary active matrices. But are some binary matrices better than other? Novak and Trapa investigates this question by introducing the concepts Nash equilibrium and evolutionarily stable strategies into the language game [9]. By following the definitions of game theory they define Nash equilibrium for the language game:

A language L is a strict Nash equilibrium if

$$F(L, L) > F(L, L') \quad \forall \quad L \neq L'$$

and a Nash equilibrium if

$$F(L, L) \geq F(L, L') \quad \forall \quad L \neq L'$$

In section 2.3.1 we derived the concept of evolutionarily stable strategies. Novak and Trapa define an evolutionarily stable language.

A language L is an evolutionarily stable strategy if

$$F(L, L) > F(L, L') \quad \forall \quad L \neq L'$$

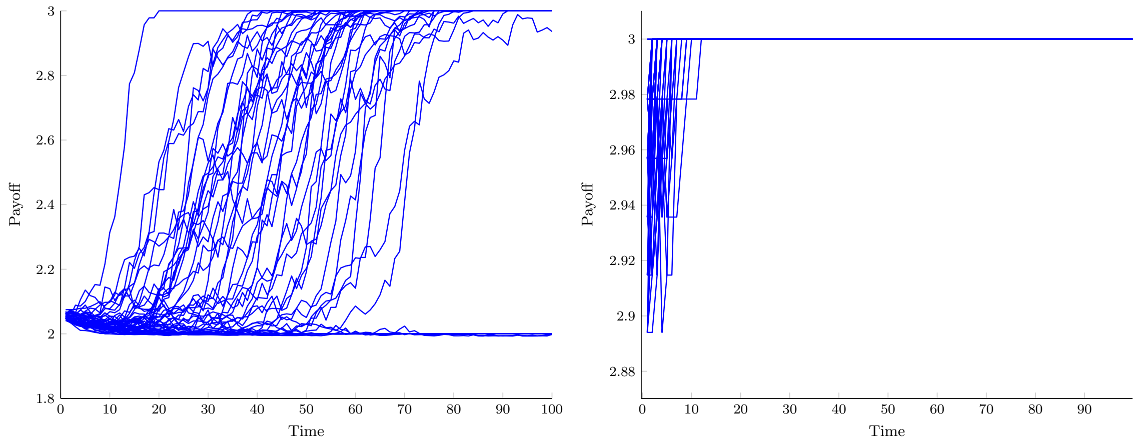
or if

$$F(L, L) = F(L, L') \quad \text{then} \quad F(L, L') > F(L', L')$$

Strict Nash equilibria and evolutionarily stable strategies are always fixed points of evolutionary dynamics. This means that all Nash equilibria and ESS must be absorbing states, but exactly which languages are Nash equilibria and which are ESS? They show that the conditions for a language to be a strict Nash equilibrium are rather restrictive. Only unambiguous languages, where each signal refers to exactly one object and one object can be referred to by exactly one signal are Nash equilibria. This means that L is a Nash equilibrium if and only if $n = m$ and P is a permutation matrix and $Q = P^T$.

In evolutionary game theory strict Nash implies ESS. But in this particular game a language is ESS if and only if it is strict Nash [9]. This means that the only evolutionarily stable languages are languages where P and Q are permutation matrices and $Q = P^T$.

Simulations for exploring these properties are shown in the figure below.



(a) (b)

Figure 5: Simulation of three object three signal game with initial conditions.

In Figure 5a, 98 of the 100 individuals speak a language L_1 with one homonym:

$$P_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \qquad \text{and} \qquad Q_i = \begin{pmatrix} 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}$$

for $i = 1, .., N - 2$.

Two of the individuals speak an unambiguous language L_2:

$$P_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \qquad Q_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

for $i = N - 1, N - 2$. After 50 runs we can see that for most of the runs, all individuals speak the unambiguous language after a number of generations. This is what to expect: the language L_1 is not evolutionarily stable, but the unambiguous language L_2 is. The evolutionary process is stochastic, so some of the runs still end up with all individuals speaking language L_1 after a number of generations.

In Figure 5b the initial conditions are the opposite: 98 of the individuals speak language L_2 and 2 of the individuals speak language L_2. In all 50 runs after a few generations, all individuals speak language L_2, as expected.

4 General Language Game

4.1 The Evolution of Vocabulary

4.1.1 Different Learning Strategies

In the article "The Evolutionary Language Game", Novak, Plotkin and Krakauer studied how different learning strategies affected the population dynamics [6]. The individuals talked about $n = 5$ objects and used $m = 5$ sounds in the simulations. One benefit with using $n = m = 5$ is that there are more absorbing states. With $n = m = 2$ there is only one other absorbing state except for the maximum fitness, and with $n = m = 3$ there are only two other absorbing states. Using $n = m = 5$ leads to more interesting dynamics. Their simulations of parental learning leads to similar results as the simulation in section 3.3: the smaller value of k the faster the population evolves a common language, but the likelier the process is to end up in an suboptimal absorbing state.

After investigating the population dynamics for parental learning, Novak et al. study the dynamics for role model learning and random learning. Role models are individuals with high fitness. Each individual imitates K individuals from the previous generation with high fitness. When $K = 1$, the implementation leads to an identical model as for parental learning. For $K > 1$ the convergence to a common language takes longer than for $K = 1$, but reaches in average a higher payoff.

The last learning model investigated is random learning, meaning that individuals imitate random individuals from the previous generation, regardless of their payoff. This means that there is no reward for higher payoff and therefore no selection for better languages. But still random learning lead to sufficient communication systems. This is because the absorbing states of this stochastic process is binary matrices. The binary matrices have on average higher payoff than matrices containing entries which numbers are between 0 and 1.

4.1.2 Taking Advantage of Mistakes

In daily life, language mistakes are made both while learning a language and when talking to other persons. Novak et al. introduce errors in the language acquisition, and hence study the effects of mistakes in the learning process. Assume that with probability ρ, a learning individual misunderstands the response of another individual. Instead the learning individual registers another randomly chosen response. Thus the probability for the learning individual to make the correct entry into her association matrix is $1 - \rho$.

Novak et al. simulates the population dynamics for four different error levels, $\rho = 0.0001$, 0.001, 0.01 and 0.1. When the error level is low, $\rho = 0.0001$ and $\rho = 0.001$, the dynamics resembles the dynamics without any noise. When $\rho = 0.01$ the only absorbing state for parental learning is languages with maximum payoff. With this rate of mistake, the errors prevent the language from getting caught in a suboptimal absorbing state. For $\rho = 0.1$ the probability of making mistakes is too high to improve the evolution of a common language, instead the all simulations lead to a fluctuated fitness which always was rather low. This implicates that there is an optimum error rate for the evolution of a common language.

4.1.3 Population Dynamics with Different Strategies

In evolutionary game theory the population dynamics for populations with different strategies is studied. Offspring inherit the strategy from their parents and their fitness is determined by their strategy, as described in section 2.3. In the evolutionary language game, the strategy children inherit is not the language, but the language acquisition. So what will happen with a population using mixed strategies?

Novak et al. consider a population where 20% learn their language from their parent and 80% acquire the language from random individuals. To begin with, the language of all individuals is just random matrices. As generations pass the fraction of random learners declines and a common language evolves. Their simulations show that parental learning is a much more efficient learning strategy than random learning when it comes to rapidly evolve a language with high communication payoff. But as soon as a common language has evolved, there is no selection against random learning individuals. This can be interpreted as an example of neutral drift.

The outcome of the competition between random learning and parental learning was not unexpected. But which strategy is the better when comparing parental learning to role model learning? As explained in section 4.1.1 there is no difference in the algorithm for the language acquisition between parental learning and role model learning when $K = 1$, where K is the number of role models. To study the dynamics for the strategies competing, they simulated the competition between the strategies that learn from their parent and M other individuals with high fitness. They did 100 runs with M from 0 to 4, all starting with equal proportions. $M = 0$ means pure parental learning. 36 runs of the simulation converged to a homogeneous population with either $M = 1$, 2, 3 or 4, 55 runs converged to a homogeneous population with individuals just learning form their parents and nine runs converged to a population with mixed learning strategies. This implicates that there is not a big difference between learning your language from your parent or from several successful individuals.

4.2 Other Studies

Several other studies have used the same model to investigate different aspects of language evolution. One approach to the evolutionary language game is made by Kenny Smith [8]. He investigates how different learning biases affect the language dynamics. Learning biases are assumed to be genetically transferred. There are three types of biases: learning rules that are biased in favour of acquiring homonyms, learning rules biased against acquiring homonyms and learning rules neutral to homonymy. As described in section 3.3.3, homonymy is a many-to-one mapping between meanings and signals. We have showed that synonymy is rare, but homonymy is plentiful in language. Smith shows with simulations that, after a few generations, populations with a learning bias favouring one-to-one mapping have better optimized languages than populations with a learning bias favouring homonymy. These results hold both for individuals learning their language from random individuals and for individuals learning from role models. Finally Smith investigates the evolution of learning biases. He finds that the only evolutionarily stable learning bias is the one in favour of unambiguous language. This outcome is similar to results of Nowak's and Trapa's work about Nash equilibrium of the evolutionary language game [9]: as described earlier they show that the only evolutionarily stable languages are unambiguous.

The model used for all the studies previously discussed is only relevant for studying the evolution of vocabulary. Vocabulary is only one of many interesting parts of the human language. Syntax, grammar and speech are some examples of what need to be studied to understand the complex process of language evolution.

The evolution of syntax is studied by Nowak, Plotkin and Jansen [5]. While other animals use signals to refer to whole situations, human language is syntactic. To understand what caused the transition from non-syntactic to syntactic language they use a model to describe how words spread in a population using non-syntactic communication. The model is similar to the SIR-model used to describe disease spreading in a population [2, pp. 83-115]. They then use the same model to describe word spreading in a population with syntactic communication and investigate when a syntactic language leads to higher payoff than a non-syntactic language. The results suggest that the essential step that lead to the transition non-syntactic to syntactic communication was the increase of relevant events that could be referred to.

Another language feature also studied by Nowak, together with Komarova and Niyogi, is the evolution of a universal grammar [4]. Universal grammar is a theory by the American linguist Noam Chomsky [7, pp. 262-263] who claims that the ability to learn grammar is native. By using formal language theory and replicator dynamics from evolutionary game theory they try to find an explanation for the evolution of an universal grammar.

In conclusion, this thesis has described a mathematical model used to explain the evolution of vocabulary. The mathematics of the model is based on game theory generally and evolutionary game theory specifically. By evolutionary game theoretical notions such as Nash equilibrium and evolutionarily stable strategies, linguistic phenomena such as homonymy and synonymy were analysed. As stated in the beginning, to understand the evolution of human language, different aspects has to be considered. A mathematical framework for the evolution of a simple vocabulary is just one of them. There are several studies that developed mathematical models to understand other aspects of the evolution of language. And hopefully, there will be even more research in this field where mathematics, biology and linguistics meet, resulting in a more profound picture of the evolution of human language.

References

[1] Mats Björklund. *Evolutionsbiologi.* Studentlitteratur, Lund, 2005.

[2] Nicholas F. Britton. *Essential Mathematical Biology.* Springer, 3rd edition, 2005.

[3] Morton D. Davis. *Game Theory - A Nontechnical Introduction.* Dover Publications, Mineaola (NY), 1997.

[4] Martin A. Nowak Natalia L. Komarova and Partha Niyog. Computational and evolutionary aspects of language. *Nature*, 417:611–617.

[5] Joshua B. Plotkin Martin A. Nowak and Vincent A. A. Jansen. The evolution of syntactic communication. *Nature*, 404:495–498, 2000.

[6] Joshua B. Plotkin Martin A. Nowak and David C. Krakauer. The evolutionary language game. *J. Theor. Biol.*, 200:147–162, 1999.

[7] Martin A. Nowak. *Evolutionary Dynamics: Exploring the Equations of Life.* The Belknap Press of Harvard University Press, Cambrige, Massachusettes and London, England, 2006.

[8] Peter Smith. The evolution of vocabulary. *J. Theor. Biol.*, 228:127–142, 2004.

[9] Peter E. Trapa and Martin A. Novak. Nash equilibrium for an evolutionary language game. *J. Math. Biol.*, 41:172–178, 2000.

A Matlab code

A.1 Two Object Two Signal Game

```matlab
% Population dynamics for parental learning

clear all

figure(1)
hold on

for t=1:50 % Number of simulations

    N = 100; % Number of individuals in the population

    P1 = [];

    a=rand(1,N); % Random entries for active matrix

    b=rand(1,N); % Random entries for active matrix

    for i=1:N %Create N active matrices
        P1{i} = [a(i) 1-a(i); b(i) 1-b(i)];

    end

    c=rand(1,N); %Random entries for passive matrix

    d=rand(1,N); %Random entries for passive matrix

    for i=1:N %Create N passive matrices
        Q1{i} = [c(i) 1-c(i); d(i) 1-d(i)];


    end


    F1 = zeros(1,N); % Create a fitness matrix with 1 row and N ...
       columns.
    for i=1:N
        %Goes through the languages of each individual
        for  m=setdiff([1:N],i)
            % Calculate the fitness between each indiviual
            for k=1:2
                % Calculate the fitness
                for j=1:2
                    F1(i)=F1(i)+0.5*(P1{i}(k,j)*Q1{m}(j,k)+P1{m}(k,j)*Q1{i}(j,k));
                end
            end
        end
        F1 = F1./(N-1);
    end

    F1 = F1./sum(F1); %
```

```
f=cumsum(F1);


G = 100; %Number of generations

% Create the next generation

F = zeros(G,N);

for l=1:G % Go through the process for a number of generations
    l

    for i=1:N % Create N active matrices
        P2{i} = zeros(2,2);
    end

    for i=1:N % Create N passive matrices
        Q2{i} = zeros(2,2);
    end

    k = 1; %Number of times a child listen to its parent

    %Create association matrices for all new individuals
    for i=1:N
        A2{i}=zeros(2,2);
    end

    for i=1:N
        parent=sum(rand>f)+1; %Give the new generation parents
        for c=1:k
            %Take a random object
            %i=round(rand*2);

            %Go through all objects

            for j=1:2
                r=rand; %Choose a random word
                if r<P1{parent}(j,1)
                    %Say 'a'
                    A2{i}(j,1)=A2{i}(j,1)+1;
                    %Mistake
                    %                   if rand<0.01
                    %                       A2{i}(j,2)=A2{i}(j,2)+1;
                    %                   else
                    %                       A2{i}(j,1)=A2{i}(j,1)+1;
                    %                   end
                else
                    %say 'b'
                    A2{i}(j,2)=A2{i}(j,2)+1;
                end
            end
        end

        for j=1:2  %Caluculate the active matrix for each ...
           individual
```

```
            P2{i}(j,1)=A2{i}(j,1)./(A2{i}(j,1)+A2{i}(j,2));
            P2{i}(j,2)=A2{i}(j,2)./(A2{i}(j,1)+A2{i}(j,2));
        end


        for j=1:2   %Caluculate the active matrix for each ...
           individual

            if   A2{i}(1,j)+A2{i}(2,j) == 0 %Check if the ...
               parents haven't talked about one word

                Q2{i}(j,1)=0.5;
                Q2{i}(j,2)=0.5;

            else %Otherwise calculatw the passive matrix in the ...
               usual way

                Q2{i}(j,1)=A2{i}(1,j)./(A2{i}(1,j)+A2{i}(2,j));
                Q2{i}(j,2)=A2{i}(2,j)./(A2{i}(1,j)+A2{i}(2,j));
            end
        end
    end

    %Calculate the fitness for the next generation

    %Children become adults.
    P1=P2;
    Q1=Q2;

    %Recalculate the fitness
    F2 = zeros(1,N); % Create a fitness matrix with 1 row and N ...
       columns.
    for i=1:N
        % Go through the languages of each individual
        for  m=setdiff([1:N],i)
            % Calculate the fitness between each indiviual
            for k=1:2
                % Calculate the fitness
                for j=1:2
                    F2(i)=F2(i)+0.5*(P2{i}(k,j)*Q2{m}(j,k)+P2{m}(k,j)*Q2{i}(j,k));
                end
            end
        end
    end
    F2 = F2./(N-1);


    F(l,:)=F2; %Store fitness for each generation


    F2 = F2./sum(F2);


    f = cumsum(F2);

```

```
    end
    plot(mean(F')) %Plot the average fitness for each generation

end
hold off
```

A.2 Three Object Three Signal Game

```
% Population dynamics for a 3-object 3-signal language game

clear all

figure(1)
hold on

for t=1:50 % Number of simulations
    t

    N = 100; % Number of individuals in the population

    P1 = [];

    a=rand(9,N); % Create an 9xN matrix with random entries used to ...
        the active matrix

    for i=1:N % Create N active matrices
        P1{i} = [[a(1,i) a(2,i) a(3,i)]./(a(1,i)+a(2,i)+a(3,i)); ...
            [a(4,i) a(5,i) a(6,i)]./(a(4,i)+a(5,i)+a(6,i)); [a(7,i) ...
                a(8,i) a(9,i)]./(a(7,i)+a(8,i)+a(9,i))];
    end

   b=rand(9,N); % Create an 9xN matrix with random entries used to ...
       the passive matrix

    for i=1:N % Create N passive matrices
        Q1{i} = [[b(1,i) b(2,i) b(3,i)]./(b(1,i)+b(2,i)+b(3,i)); ...
            [b(4,i) b(5,i) b(6,i)]./(b(4,i)+b(5,i)+b(6,i)); [b(7,i) ...
                b(8,i) b(9,i)]./(b(7,i)+b(8,i)+b(9,i))];
    end


    F1 = zeros(1,N); % Create a fitness matrix with 1 row and N ...
        columns.
    for i=1:N
        % Go through the languages of each individual
        for m=setdiff([1:N],i)
            % Calculate the fitness between each indiviual
            for k=1:3
                % Calculate the fitness
                for j=1:3
                    F1(i)=F1(i)+0.5*(P1{i}(k,j)*Q1{m}(j,k)+P1{m}(k,j)*Q1{i}(j,k));
                end
            end
        end
```

```
    end

    F1 = F1./(N-1);

    F1 = F1./sum(F1);

    f=cumsum(F1);


    G = 100; % Number of generations

    % Create the next generation

    F = zeros(G,N);

    for l=1:G % The process for a number of generations

        for i=1:N % Create N active matrices
            P2{i} = zeros(3,3);
        end

        for i=1:N % Create N passive matrices
            Q2{i} = zeros(3,3);
        end

        k = 5; % Number of times a child listen to its parent

        % Create association matrices for all new individuals
        for i=1:N
            A2{i}=zeros(3,3);
        end

        for i=1:N
            parent=sum(rand>f)+1; % Give the new generation parents
            for c=1:k
                % Take a random object
                % Go through all objects

                for j=1:3
                    r=rand; % Choose a random word
                    if r<P1{parent}(j,1)
                        % Say 'a'
                        A2{i}(j,1)=A2{i}(j,1)+1;
                    elseif r<P1{parent}(j,1)+P1{parent}(j,2)
                        % Say 'b'
                        A2{i}(j,2)=A2{i}(j,2)+1;
                    else
                        % Say 'c'
                        A2{i}(j,3)=A2{i}(j,3)+1;
                    end
                end
            end

            for j=1:3

                % Caluculate the active and passive matrix for each ...
```

```
                        individual

                    P2{i}(j,1)=A2{i}(j,1)./(A2{i}(j,1)+A2{i}(j,2)+A2{i}(j,3));
                    P2{i}(j,2)=A2{i}(j,2)./(A2{i}(j,1)+A2{i}(j,2)+A2{i}(j,3));
                    P2{i}(j,3)=A2{i}(j,3)./(A2{i}(j,1)+A2{i}(j,2)+A2{i}(j,3));

                end
                for j=1:3
                    if     A2{i}(1,j)+A2{i}(2,j)+A2{i}(3,j) == 0  % ...
                        Check if the parents haven't talked about one word

                        Q2{i}(j,1)=1/3;
                        Q2{i}(j,2)=1/3;
                        Q2{i}(j,3)=1/3;

                    else % Otherwise calculate the passive matrix in ...
                        the usual way

                        Q2{i}(j,1)=A2{i}(1,j)./(A2{i}(1,j)+A2{i}(2,j)+A2{i}(3,j));
                        Q2{i}(j,2)=A2{i}(2,j)./(A2{i}(1,j)+A2{i}(2,j)+A2{i}(3,j));
                        Q2{i}(j,3)=A2{i}(3,j)./(A2{i}(1,j)+A2{i}(2,j)+A2{i}(3,j));
                    end
                end
        end

        % Children become adults.
        P1=P2;
        Q1=Q2;

        % Calculate the fitness for the next generation
        F1 = zeros(1,N); % Create a fitness matrix with 1 row and N ...
            columns.
        for i=1:N
            % Go through the languages of each individual
            for m=setdiff([1:N],i)
                % Calculate the fitness between each indiviual
                for k=1:3
                    % Calculate the fitness
                    for j=1:3
                        F1(i)=F1(i)+0.5*(P1{i}(k,j)*Q1{m}(j,k)+P1{m}(k,j)*Q1{i}(j,k));
                    end
                end
            end
        end


        F(l,:)=F1/(N-1); % Store fitness for each generation


        F1 = F1./sum(F1);


        f = cumsum(F1);

    end

```

```
plot(mean(F')) % Plot the average fitness for each generation

end

hold off
```