DEPARTMENT OF COMPUTER SCIENCE

COMSM2202

Research Review

Inferring population dynamics of house-hunting ants

Student:

Christina Pantopoli

Supervisor:

Dr. Oliver Ray

Contents

1. Executive Summary

The behaviour of ant-species *Temnothorax albipennis* has been extensively studied by the Bristol Ant Lab, which led to the discovery of many mathematical models that examine and analyse the process of an ant colony emigration from one nest site to another. These models describe this occurrence by a set of differential equations which detail the development of the different categories of the ants in a colony. Although these mathematical models were developed by the use of different equation discovery models, they do not take into account the relationship between the equations as each equation was created individually.

The aim of the project is to create a differential equation discovery model through the use of genetic algorithms while taking into account the relationship between the equations and then integrate that into a software. The software will be developed on either C or Mat Lab as they offer libraries on genetic algorithms and other algorithms that are used in machine learning to solve equations. Within this project we will also be looking into how credit assignment works on equation discovery systems. The final step we aim to do is to utilise the genetic algorithms produced, to generate ODEs (ordinary differential equation) that represents the emigration process of an ant colony and compare them with existing ODEs that were discovered using other models. This project is 80% part II, 10% part I and 10% part III.

Therefore, the key deliverables of this project are:

- The production of a genetic algorithm that outputs the expected equation based on the input and also generate equations that might depend on the development of another one.
- A prototype software that will the genetic algorithm to find equations.
- An evaluation of the algorithm by running experiments with different input data and compare the result with the expected equation by applying methods of statistical analysis like margin of error.
- An assessment of the credit assignment of the genetic algorithm. Essentially discover if there are errors, where it occurred and how important it is to the accuracy of the outcome.
- An analysis of the ODEs created by this software and previous software by determining their accuracy.

Upon completion, the project will offer development in many areas. The main novelty is that it will provide an equation discovery model that can find sets of equations whereas all the existing ones do not offer that. Moreover, it will help deliver more clear results for the population dynamics of ants during the process of ant emigration since after we run data collected from experiments that observe the ants during emigration, the software will deliver more accurate equations that outline the behaviour of ants. This project will also assist in the improvement of the tracking software for the ants that is used as we could compare the data that the tracking software provides with the data that the ODEs supply and use a statistical analysis to find if the data are precise. If not then the project could aid to enhance the tracking software so it will take into account more variables or other defining conditions.

2. Introduction and Scope

2.1. Introduction

Finding solutions to problems using algorithms and optimization methods has been long considered a major development in the field of evolutionary computation. Within this project we aim to use these methods to develop a new equation discovery system that will help us learn equations from data in a more accurate way and develop set of multiple equations instead of a single equation each time. Through the development of this system we seek to solve equations that describe the process of an ant colony emigrations from one nest to another and compare them with equations that have been discovered using other equation discovery systems.

2.2. Motivation

This project was derived from pursuing the creation of equations that can explicitly describe the behaviour of *Temnothorax albipennis* during the emigration process. This species has been extensively been study in the University of Bristol as many useful information can be obtained by observing the behaviours of ants and the decisions they make in different environments.

Fig. 1 Temnothorax albipennis

The Bristol Ant Lab has conducted many experiments on this species of ants during the emigration process from one nest to another and each time a parameter changes so as have a better understanding of this process. For example, one experiment includes obstacles in the route from one nest to the other, another has the initial nest have a see-through roof and many other variables change for each experiment. These experiments are recorded and then the videos are processed through a tracking software which offer data in a format that can be used in equations discovery models.

Finally, after the results from these experiments are processed through the equation discovery systems they offer mathematical models that describe the movement of the ants during the emigration.

Different equation discovery system were used by other postgraduates and PhD students of the University of Bristol during the period of their thesis and sets of equations were discovered accordingly. The models that have been used so far are Pratt, Planqué and Nutonian Eureqa.

Out of the models used, the most capable model is Eureqa as it can discover equations for a variety of data even if the data are extremely complex. However, Eureqa as all the models mention above have the ability to only execute on single equations which set back the discovery of sets of equations.

The outcome of this project will provide the ability to use a system so as to solve multiple equations in parallel and each equation is relevant to each other so their accuracy should also improve. Moreover, it will enable biologists to derive more precise results that describe the behaviour of ants that could also contribute to the improvement of the tracking software that is used to interpret the movements of ants.

2.3. Areas of Research

To achieve the objectives set out, the necessary knowledge of various scientific areas are in need and they will be covered later on inside this research review. These scientific areas can be identified into three parts: machine learning, genetic algorithms and equation discovery.

Indeed Machine learning is an area that will be necessary to implement the project. Within this segment we will describe why machine learning is so important nowadays and the various tools that are used as Machine learning techniques. We will also explain the reason as to why we are going to use these tools

Genetic Algorithms are paramount to the realization of this project. Within this section we will describe why genetic algorithms are the preferred method that will be used for this project and the detailed representation of a normal process it follows. Additionally, we will cover the advantages and disadvantages of using them.

Genetic Programming:

We cannot forget to mention equation discovery systems as the projects aim essentially creating one. This part will briefly cover some of the equation discovery systems that were already mention. Nutonian Eureqa will be examined in more detail as it is the foundation that will be used to create our own equation discovery system.

3. Machine Learning

3.1. Broad Machine Learning Background

Machine Learning is a section of computer science that was developed from trying to improve the performance of computers so as to give them the ability to learn. Machine leaning consists of the study of algorithms and models that increase their performance through experience.

A more formal definition of machine learning is : "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E" [1].

There are three types of problems and tasks in machine learning supervised learning, unsupervised learning and reinforcement learning. For this project we will be using supervised learning as we know what the expected results should be. Machine learning can also be divided in other categories based on the output of the task such as classification problems, regression and other. In our case the category our learning task belongs is a regression problem as the outputs will be continuous.

Machine learning algorithms are of great use when trying to solve problems that involve big data sets and trying to discover patterns or irregularities in the data or when trying to make a program adapt to conditions that keep on altering.

The modelling of a machine learning problem includes selecting the appropriate training experience, choosing the type of the target function and how it is represented and deciding on the algorithm for developing the target function from training examples.

3.2. Genetic Algorithms Background

General Information

Humankind has always wanted to develop new techniques and tools so that it would enhance the performance of tasks, as a result scientists believed that evolution could be used in systems as a tool to optimize solutions for technical problems. The idea behind this was the development of a population with all the possible solutions for a given problem and apply techniques inspired from genetic evolution and natural selection. In the mid 70's genetic algorithms made the their first appearance when J. Holland created a model that could solve optimisation problems with the use of GAs.

GAs are mostly used to find the best hypotheses for a problem, so the search for it starts with a population of initial hypotheses and the hypotheses could be considered as the genes in a chromosome. With the use of natural selection and genetic operators such as mutation and crossover, a new generation of hypotheses (genes) is born by selecting the hypotheses that are evaluated to be most fit. The fitness is calculated with the use of a probabilistic model which calculates the fitness of one hypothesis to the proportional fitness of the other current hypotheses. Natural selection guarantees that the hypotheses with the best fitness will be used to generate the future populations. By using the crossover operator during the process, the GA combines genes from two hypotheses and combines them to create two new offspring which will be better fit than their parents. Genes could be mean bits of strings or a node that contains a part of a program of each hypothesis. Another genetic operator used is mutation which is often executed after crossover, it performs a random change in the new generation of hypotheses which allows for a single child to be produced from a single parent. An impressive achievement is that by combining the crossover and the mutation operator, the population can move from the local optimum it might get deadlocked in, this way the fitness of the new generation gets keep improving until the GA will provide the best hypotheses. GAs have many advantages and disadvantages and some of them will be stated below.

There are many reasons why we are going to use GAs:

- They can seek the space of hypotheses that have complex pieces and the effect of each piece on the hypothesis may be demanding to describe with other methods.
- GAs do not require any information on the gradient descend.
- Their behaviour when solving big scale optimization problems is ideal and they offer a solution more accurately and faster.
- They can find a hypothesis which is a global solution and that has high fitness.
- GA can handle a parallel search for hypotheses.

However, GAs are not always suitable as they have some disadvantages:

- GAs do not guarantee to find the local optimum, since if they reach a point where the fitness of the population satisfies a condition, they will stop their evolution.
- The best hypotheses is found by comparing it with the rest hypotheses that belong in the population as a result if the initial is not diverse enough. Even if the GAs run multiple iterations, they will not discover the best hypothesis but just the "best" hypothesis based on the original parents.
- GAs are unable to find solutions when the fitness function returns only true or false as the search will not converge on the answer.

Typical structure of Genetic Algorithms

GA($Fitness$, $Fitness\_threshold$, p, r, m)

Fitness: A function that assigns an evaluation score, given a hypothesis.
$Fitness\_threshold$: A threshold specifying the termination criterion.
p: The number of hypotheses to be included in the population.
r: The fraction of the population to be replaced by Crossover at each step.
m: The mutation rate.

- *Initialize population:* $P \leftarrow$ Generate p hypotheses at random
- *Evaluate:* For each h in P, compute $Fitness(h)$
- While $[\max_h Fitness(h)] < Fitness\_threshold$ do

Create a new generation, P_S:

1. *Select:* Probabilistically select $(1-r)p$ members of P to add to P_S. The probability $\Pr(h_i)$ of selecting hypothesis h_i from P is given by

$$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$

2. *Crossover:* Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P, according to $\Pr(h_i)$ given above. For each pair, $\langle h_1, h_2 \rangle$, produce two offspring by applying the Crossover operator. Add all offspring to P_s.
3. *Mutate:* Choose m percent of the members of P_s with uniform probability. For each, invert one randomly selected bit in its representation.
4. *Update:* $P \leftarrow P_s$.
5. *Evaluate:* for each h in P, compute $Fitness(h)$

- Return the hypothesis from P that has the highest fitness.

Fig.2 Basic steps of a GA [1]

The core body of a GA consists of five steps:

1) Initialization: The initial population is created either by picking random hypotheses or by disrupting an input hypotheses. The initialization of the population is not considered critical since the population will evolve so as to become have more disparate hypotheses based on the optimization parameters.

2) Evaluation: In the second step the fitness of each hypotheses is calculated with the use of the fitness function which is the only criteria of evaluation of the hypotheses. This assessment is used either as a terminate condition or the process of probabilistic choosing which of the current hypotheses are going to be used to create the next generation.

   The fitness function receives a hypothesis and returns a value that represents the degree of fitness for that hypothesis, usually the output is in the margin from 0 to 1 where 1 indicates that the hypothesis meets all the conditions of the fitness function and it is an acceptable solution. The representation of the fitness function depends on the problem and can be from simple to very complex. Even though the ideal fitness function is continuous and monotonous it rarely is, so the next thing we seek is for it to have many local maximum or an isolated global maximum.

   The most common approach to select which hypotheses are going to be selected is based on appraising their quality is probabilistically. The decisive point is that the fitness function that is chosen must be accurate and it should provide a computational cost that gives the best results.

3) Natural selection: In this step the hypotheses with the best fitness score are put together in a subset in no particular order. However, the hypotheses which are not fit enough are discarded from the population. The number we select and dispose is not random, we use one of the variables passed as input in the GA to compute the proper amount in each case. In the end the best fit hypotheses are added to the population of the new generation.

   Hypotheses can be selected or replaced depending on many different methods. The method mention in the figure Fig.2 is called fitness proportionate selection and it computes the ratio of its fitness to the sum of the fitness of all the hypotheses, which shows the probability that a hypothesis will be selected. Another method is the tournament selection where the population is divided into pairs and then the most fit of the two remains becomes a parent for the next generation whereas the other one gets changed. This method offers a more diverse population. An additional method is rank selection where the hypotheses are ranked according to their fitness and there are chosen based on their rank.

   When selecting the hypotheses that will be used for the new population, one should be in caution to avoid crowding which happens when a hypothesis is greatly better than the others so the population instinctively carries characteristics only from that hypothesis. This leads to the population becoming less diverse and it slows down the progress of the GA.

4) Crossover: In the fourth step a genetic operation is performed on our population. A certain amount of hypotheses are picked based on their fitness and placed into pairs and they are combined so as to produce two children by taking information from each parent. The children are then placed into the population of the generation. It is considered that the children are better than their parents since they carry characteristics from both of them.

   There are different methods to apply crossover based on the crossover mask that will be chosen to use. The crossover operation continues being applied to the new population.

5) Mutation: In the final step another genetic operator is executed which is the mutation operator. This operator produces one child by on parent by changing random characteristics of the parent. This step offers the possibility to generate solutions that do not exist in the population.

At this point, the population contains hypotheses that have better quality than the previous one and the steps from 1 to 5 are repeated as long as the GA is terminated. The GA can be terminated depending on many conditions:

- There is a fixed number of iterations that the GA is supposed to perform.
- The degree of suitability of the best solution is greater than a given threshold.
- Finding a solution that .meets the minimum standard.
- If better results are not generated after a number of iterations that means it has reached a local optimum and the operation should come to an end.

6) Convergence and Update: The method selection of parents that will be used in the crossover operation significantly affects performance of the GA. Two problems that often occur during the selection process are premature convergence and slow convergence. Convergence is the prevalence of a solution or small variations of that solution in a large percentage of the population, if the GA is efficient then the population will converge after enough iterations to the global optimum. In premature closure, the population rapidly converges around a solution that is a local optimum. This phenomenon occurs in cases where the fitness function shows very sharp peaks and intense local maxima. To address this problem one solution is to remap the fitness function so as to become smoother. In the case of slow convergence the exact opposite transpires after a huge number of iterations the population still does not converges and the only way to work this out is by changing the fitness function to give a more intense fluctuations. Previously, I mention that some of the original solutions are placed in the new population along with the new developed children this method is called partial-renewal and it allows the children to compete with their parents so that the best one prevails.

Problem Domains

It is not specific where it is appropriate to use GAs to an optimization problem. Some scientists claim that there is no better method for optimization other than GAs even though there similar methods that exist such as gradient-descent, Monte-Carlo, Simplex – Simplex optimization and many others that can perform the same task. The key is to use the knowledge of the system which is going to be optimized in order to choose the best technique that finds the best solution faster.

When there is not enough knowledge on the calculation of the gradient descent then GAs are preferred. Furthermore, another reason why GAs are preferred is their versatility. GAs can use the knowledge of system so as to customize itself for example if the program is caught in a local optimum then it can change the mutation operator. So while there is no guarantee that GAs can behave better than other methods for an application, they do not fail to achieve a satisfactory solution to the optimization problem. Another feature GAs have is that they do not directly optimize their variables but they optimize their performance.

GAs are not the optional preference when the problem requires the calculation of gradient descent, when the global optimum is a requirement and when the population of the GA is too excessive.

3.3 Genetic Programming Background

If the crossover operator is applied on genetic programming algorithm then a random subtree of each parent

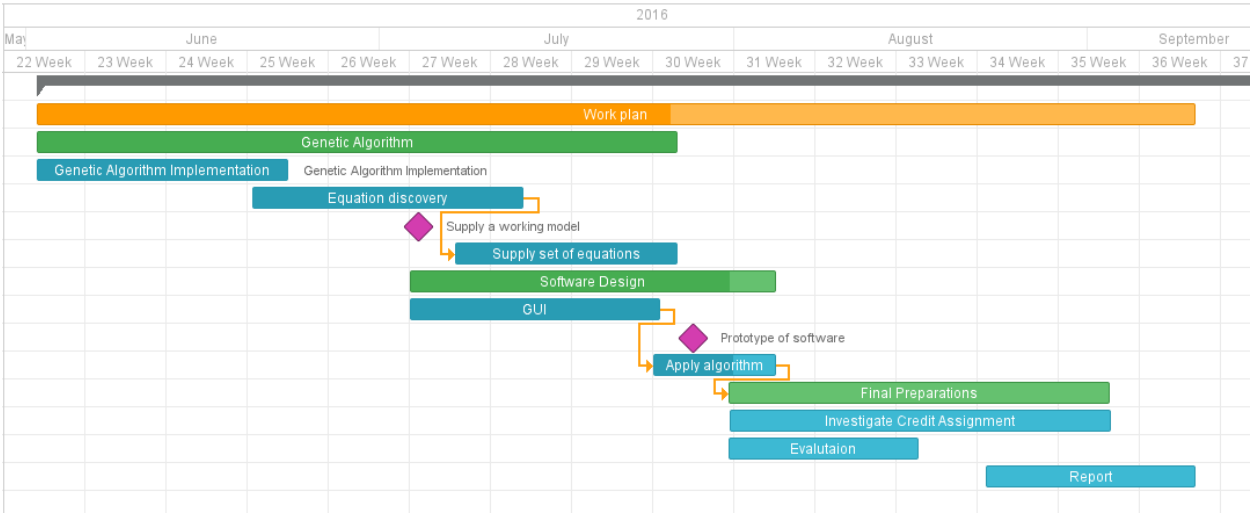
Eureqa employs an accuracy-parsimony Pareto front to narrow down the number of solutions it presents. However, the Eureqa system includes a failing in that it has no consideration for initial conditions of differential equations.

. Expected Timeline

Expected Timeline

The section for the expected timeline has been divided into two parts. The first part is a Gantt chart which describes the expected timeline and the time periods of each task for the project. The second expands upon the stages and milestones contained within the Gantt chart.

	Task name	Start date	End date
	Project Develpomemt	01/06/16	20/01/17
1	Work plan	01/06/16	09/09/16
1.1	Genetic Algorithm	01/06/16	09/09/16
1.1.1	Genetic Algorithm Implementation	01/06/16	23/06/16
1.1.2	Equation discovery	20/06/16	13/07/16
1.1.3	Supply a working model	04/07/16	04/07/16
1.1.4	Supply set of equations	07/07/16	27/07/16
1.2	Software Design	04/07/16	04/08/16
1.2.1	GUI	04/07/16	25/07/16
1.2.2	Prototype of software	28/07/16	28/07/16
1.2.3	Apply algorithm	25/07/16	04/08/16
1.3	Final Preparations	31/07/16	02/09/16
1.3.1	Investigate Credit Assignment	31/07/16	02/09/16
1.3.2	Evalutaion	31/07/16	16/08/16
1.4	Report	22/08/16	09/09/16



. Expected Timeline

1.1. Genetic Algorithm Implementation

We will examine many different genetic algorithms so as to try figure out which will the most beneficial for our project. This stage will involve finding which fitness function to use and all the parameters that are needed to create a genetic algorithm.

1.2. Equation discovery System

After the decision of the GA is finalized, we then try to create one that can solve equations based on the data given. Firstly, we will try to solve equations that are very simple such as linear equations and then move on to more difficult equations like differential equations which is our goal.

Supply a working Model - Milestone: At this point we should have an algorithm that if given certain data will be able to solve a number of equations. If we reach this

1.3. Supply set of equations

The next step will be to develop a system based on the algorithm, in order to enable it to solve more than one equation each time and the solution of each equation to be depended on the development of the other.

1.4. Software design

At this point, we will have to decide which programming language to use that has all the capacities needed. It should offer genetic algorithms libraries and other options that will come in mind later on. The features that this software demands are straightforward as the working of it is mostly based on just input and output.

1.5. Apply algorithm

This stage concerns the design the implementation of the algorithm created so far. The program should be competent that it takes data from the user and the options needed for the algorithm to run and then supply the equations formed on the data.

9. Risk Analysis and Contingency Plan

In this section of the research review we will list the risks that might occur in the development of the project and affect the completion of the project. Included in this list is an analysis of their likelihood, their severity and the preventing measures that will be taken to either stop them from occurring or to resolve them if they do happen.

9.1. Limited Time / Operational Feasibility

As I have not background in machine learning, it may take me longer time to be able to implement the algorithms needed to create the equation discovery model. I have no previous knowledge in genetic algorithms or genetic programming so that might bring the development of the expected equation discovery model to a halt.

Contingency plan: Use the non-allocated time to catch up to any obstacles that might occur and consult with an expert if possible to receive advice.

Severity: Medium

Likelihood: 50%

9.2. Illness

I might get sick during the development of the project or for some unforeseen reasons I might be unable to work for a period of time and that will affect my progress.

Contingency plan: I have left small periods of time in the work plan as empty just in case I get sick and just to be more flexible with the work plan. Anything more serious that will take a more extensive period of time would be deem by the university as an exception.

Severity: Low

Likelihood: 20%

9.3. Impossibility in creating a discovery equation model for a set of equations

This is the risks with the highest impact on the project. Without starting the process of creating the genetic algorithm that will be the base of the system it is difficult to say the likelihood of this occurring.

Contingency plan: The focus of the project will alter in performing credit assignment in the equation discovery model that will be developed at that point.

Severity: High

Likelihood: 40%

References

[1] - T.M. Mitchell. Machine Learning. New York ; London.1997

[2] - J.R. Koza. Genetic programming: on the programming of computers by means of natural selection. Cambridge. MIT Press. 1992

[3] - M.Schmidt and H.Lipson. Distilling free-form natural laws from experimental data. Science. Vol.324, no. 5923, pp.81-85, 2009.

[4] - J.J. Grefenstette. Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms. Machine Learning, Vol.3 (2), pp.225-245. Washington. 1988

[5] - http://www.discoverlife.org/mp/20q?search=Temnothorax+albipennis