

MACHINE LEARNING APPLICATIONS IN
MICROBIAL POPULATION DYNAMICS ANALYSIS

by

Taotao Yu

A THESIS

Presented to the Faculty of
The Graduate College at the University of Nebraska
In Partial Fulfillment of Requirements
For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Stephen D. Scott

Lincoln, Nebraska

December, 2000

Machine Learning Applications in Microbial Population Dynamics Analysis

Taotao Yu, M.S.

University of Nebraska, 2000

Adviser: Stephen D. Scott

With the advent of new, efficient molecular biological techniques, modern biology is encountering an unprecedented information explosion stage. Astronomical amounts of data, especially DNA and protein sequence data, are imposing a great challenge for current computational methods. Computers are going to play a key role not only in accumulation and retrieval of data, but also in its interpretation. Machine learning algorithms are increasingly receiving researchers' attention and interest, because such approaches are ideally suited for domains characterized by the presence of large amounts of data, "noisy" patterns, and the absence of general theories.

In this thesis, we designed a framework based on hierarchical clustering algorithms to identify genetically similar groups in microbial populations from microbial DNA fingerprinting data. We also implemented a synthetic data generator based on parameterized mathematical models of microbial

population growth and saturation, which can be used to directly compare regressors and classifiers. Furthermore, we implemented various regression algorithms and evaluate their performance on the synthetic data. We also report some results on the minimum data requirements for these regression methods to accurately model the populations. Also, we propose a methodology for automatically identifying saturation points in population dynamics datasets. Finally, we present some suggestions of further research, such as new regressors that may converge faster and can still identify the changes in the growth model.

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor, Dr. Stephen Scott for his advice, encouragement and support throughout my study in UNL.

I would also like to take this opportunity to thank Dr. Andrew Benson for his invaluable inputs and great vision for this research. My sincere appreciation is also extended to my other committee members: Dr. Jitender S. Deugon and Dr. Tapabrata Maiti for their help in various ways.

I am also appreciative of the input on this thesis from Arumugam Manimozhayan.

Thanks to my family, for their love and encouragement over the years to help me reach this point. I am especially grateful to my lovely wife, Song, for her understanding, encouragement and wholehearted support.

Table of Contents

Chapter 1	Introduction.....	1
1.1	Motivation.....	1
1.2	Purpose of the Research.....	3
1.3	The Experiments.....	4
1.4	How this Thesis is Organized.....	7
Chapter 2	Background and Review.....	8
2.1	Integration of Computer and Biology.....	8
2.2	Why Machine Learning Approach.....	9
2.3	Clustering Algorithms and Pattern Recognition.....	11
2.4	Microbial Population Dynamics.....	16
2.4.1	General Microbial Population Model.....	17
2.4.2	Current Trends of Microbial Population Dynamics.....	22
Chapter 3	Data Analysis Approaches.....	24
3.1	Linear Regression Analysis.....	24
3.2	Gradient Descent For Finding Best Regressor.....	26

Chapter 4	Experimental Results and Discussion.....	31
4.1	Growth Model Implementation.....	31
4.2	Data Analysis Experimental Results.....	33
Chapter 5	Conclusions and Future Development	41
References	44
Appendix A:	Sample of E. coli DNA Fingerprinting Gel Pattern and Part of its Corresponding Bit File.....	48
Appendix B:	Example of Clustering Procedure.....	49
Appendix C:	Experimental Procedure for Artificial Data.....	50

Chapter 1 Introduction

1.1 Motivation

The idea of computing is as old as human history. People always want to expand their computing power by inventing some complex counting devices. In the late 1930s, the English mathematician Alan Turing described the principles, including all the capabilities and limitations, of a hypothetical universal computing machine known as the Turing machine. Based on such theoretical work, people started to build modern electronic computers. The first operational electronic, general-purpose computer, the ENIAC, was built in the 1940s. Later John von Neumann developed the "stored-program concept", which has greatly influenced the design of the modern computer. Since then, the impact of computers on extending human intelligence has been phenomenal and unprecedented. Computers have moved into every aspect of society, culture and science rapidly and completely. This has led to the information revolution in the history of the human civilization. There is now a new definition of scientific investigation, with the computational scientists joining theoretical and experimental scientist in every discipline, such as chemistry, physics, geoscience, molecular biology, etc.

The greatest impact has been on biology, whose main goal is to understand where we are from, how we live, and how we become distinct individuals. Biology, traditionally considered as an experimental science field that emphasizes on the experimental results to derive theory, now is entering into a new era because of modern computer technology. The ever-increasing power of the modern computer has given scientists new opportunities to tackle problems so complex that they could not have been investigated without computers.

Computers are extending scientists' reach by enabling them to analyze enormous amounts of data, simulate complex systems, and exploit hidden patterns. Computational analysis of biological sequences (linear descriptions of proteins, DNA and RNA molecules) has completely changed its character since the late 1980s [1]. The development of new, efficient experimental techniques, such as PCR, microarray technology and DNA sequencing, has led to an information explosion, especially with the Human Genome Project launched in the late 1980s. The Human Genome Project is one of the most exciting research efforts ever, which is to sequence and map all the genetic code of a human being. Such an effort would not be possible without the computer's aid. The computer is playing a key role in accumulation and interpretation of enormous volumes of data. A new interdisciplinary field called bioinformatics, which integrates computer science and molecular biology, has emerged and has been one of the fastest growing fields in recent years.

Microbial population dynamics analysis is an active research area in predictive microbiology and microbiological ecology, with the main goal to describe, model and predict the dynamic behavior of microbial populations by mathematical and statistical measurements. Furthermore, population dynamics research enables scientists to investigate the changes of microbial diversity, the behavior of some particularly interested microorganism at different levels. At the same time, because of the development of molecular tools such as PCR and DNA fingerprinting, scientists are able to directly investigate inherent genetic information. This is a vast improvement over the time-consuming culture processes previously used to identify distinct genotypes and strains. However, on the other hand, those modern techniques produce significantly more data than ever before, which can only be modeled and analyzed by the ever-increasing computing power. The inherent complexity and variability of microbial populations poses great

challenges to conventional algorithms and approaches. Hence, it is important to apply some robust and efficient algorithms that are capable of extracting useful information from complex and noisy systems. Machine learning deals with how to construct computer programs that automatically improve with experience. Imagine computers learning from medical records which treatment is most effective for specific diseases on specific people. Algorithms based on machine learning techniques have been proved to have the advantage of robustness, noise-resistance and capability to automatically learn the underlying theory from the training data, through classification, inference, decision-making and model fitting. Many of them have already been successfully applied to various practical problems, such as speech recognition [2], robot control [3], and image analysis [4].

1.2 Purpose of this Research

Microbial dynamic population analysis is one of the oldest research areas in microbiology. It would be difficult to overstate the significant importance of microorganisms in the whole ecosystem. Recent research developments in system ecology, organism microbiology, and molecular biology indicate that microbial diversity is far beyond our imagination. More and more new microbes are going to be discovered. The importance of microorganisms comes not only from their major role in geochemical cycle and energy production, but also from their presence in every aspect of human life. Better understanding the distribution and activity of certain microbial dynamic populations will greatly help us to define the genetic diversity in the population, and provide better control and management for microorganisms.

We are faced with a dynamic universe. Every object we perceive is undergoing change of one sort or another, rapidly or slowly. Dynamic shifts in microbial populations are likely to be associated with environmental or physiological changes. For example, the

infection of a human or animal's gastrointestinal tract by *E. coli* O157:H7 (an emerging cause of foodborne illness, toxin-producing *E. coli* strain) is likely to significantly affect the total microbial population of the gastrointestinal tract (perhaps due to source competition). On the other hand, the capability and activity of *E. coli* O157:H7 to infect the gastrointestinal tract is also influenced by other population components, as well as other internal and external factors. For example, if some antimicrobial agents are introduced into the sample populations, the experimental data sets corresponding to the shifts between the control group and treatment group can be fed into some machine learning algorithm to find how the variables and other population residents affect the capacity of a certain microbe to enter and flourish in the population. In our research, we use DNA fingerprinting data for the identification of natural *E. coli* microbial populations and study the pattern and distribution of homogenous microbial groups in the population. This research employs classical bacterial techniques coupled with more recent developments in molecular biology as well as modern computer science algorithms, such as clustering algorithms and learning algorithms, to process and analyze the data. Because of certain advantages of machine learning algorithms, algorithms based on machine learning theory are being considered in this research. To our knowledge, no similar approaches have been published. This thesis makes an attempt in that direction by applying certain machine learning techniques as a starting point for further investigation.

1.3 The Experiments

In this project, we apply computational approaches and develop certain modeling and analysis tools for microbial population dynamics based on DNA fingerprinting information. Because of a lacking of real data, we developed a framework for artificial data

generation from a real data set based on certain models. The real data set provided by Dr. Andrew Benson's Lab in Department of Food Science and Technology (University of Nebraska-Lincoln) contains sequences of binary data corresponding to the *E. coli* DNA fingerprinting gel pattern obtained through a series of molecular biological protocols. *E. coli* samples are collected from cow's intestinal tracts at different time points. Each cow's diet is strictly controlled to minimize the nutrients' effect on microbial population inside the intestinal tract. Then after several continuous steps of dilution to decrease the microbial concentration to some degree appropriate for culturing, the samples are placed onto an *E. coli*-specific medium, where only *E. coli* strains can survive and undergo a normal life cycle. Different *E. coli* strains develop into distinct colonies on the medium. Then *E. coli* DNAs are extracted and purified from twenty randomly-selected colonies (called *isolates*) according to certain molecular biology protocols. After that, all *E. coli* DNAs are analyzed using electrophoresis to be separated on the gel. This procedure is called *DNA fingerprinting*. The gel pattern with certain DNA bands on certain positions defines the difference of genetic information in each *E. coli* strain. Those band patterns can be converted to a binary format. If there exists a band at a certain position, bit 1 is obtained, otherwise, 0 is obtained. By such a conversion, we are able to link various *E. coli* strains directly to the binary data, which can be easily manipulated and analyzed. For details on the experimental procedures, see Kim et. al.[5].

The starting file contains *E. coli* strain genetic information from two diet-controlled cows at ten time points, with twenty isolates at each time. So our data sets actually have three parameters: the number of animal hosts (in this case, the number of cows), the number of time points, and the number of isolates at each time points. Thus a total of $2 \times 10 \times 20 = 400$ strings of binary data from the starting file serve as the template for artificial data generation.

Each artificial data file contains different sequences of binary data, or bit vectors. The artificial data generation procedure is discussed in detail in Section 4.1.

Then we applied a hierarchical clustering algorithm called the Matrix Updating Algorithmic Scheme (MUAS) on those generated artificial binary data sets to combine the DNA fingerprinting binary vectors into genetically homogenous clusters called *groups*. We use the *Hamming distance* to measure inter-vector distance for the MUSA algorithm. The Hamming distance between vectors v and w , denoted by $d(v,w)$, is defined as the number of places where v and w differ. For example, the Hamming distance between $v=(1\ 1\ 0\ 0\ 0\ 1\ 1)$ and $w=(0\ 0\ 1\ 1\ 1\ 0\ 0)$ is 7. We will provide further details of general clustering procedures and MUAS in Section 2.2.

In hierarchical clustering, the data can be viewed in a binary tree structure (also called a threshold dendrogram, or simply a dendrogram). Based on the dendrogram derived from the MUAS, we are able to select a complete clustering based on a distance threshold. We trace from the root of the tree until internal nodes with distance less or equal to the threshold are found. Each subtree rooted at such nodes is considered to be one group, i.e. the leaves of the subtree correspond to binary vectors with a low degree of dissimilarity and thus belong to the same group. Then the abundance of this particular group from one specific host and one specific time point is just the total number of binary vectors that belong to the group from that specific host and time. For an example of the clustering procedure, please refer to Appendix

After the abundance of each group is collected, we randomly assign different growth models to each individual group based on the artificial data generating procedure discussed in Section 4.1. Such an artificial growth model generator can be directly used in further theoretical research on microbial population dynamics.

In order to find a good linear regressor between abundance data and time points generated from the exponential growth model, we use the natural logarithm of abundance data (logarithm to the base e , or \ln). I.e. all response variables (abundance data) used in following experiments are actually the natural logarithm of original abundance data. We applied a standard linear regression algorithm and an on-line gradient descent (GD) algorithm on the transformed abundance data. Those algorithms are discussed in detail in Chapter 3. Preliminary results show that the standard linear regressor finds a good fit only if there is exponential growth phase all over the time points, i.e. there is no saturation phase in the growth model. The GD algorithm is capable of finding the different linear regressors that are best for different phases of the growth model, i.e. before the saturation point and after the saturation point. Also we evaluated GD's performance with different learning rates. Finally we propose some preliminary ideas on the minimum data requirements (number of sample points) for GD and standard linear regression algorithm based on our preliminary results.

1.4 How this Thesis is Organized

This thesis consists of five chapters. Chapter 2 will introduce some background knowledge about machine learning, clustering algorithms and microbial population dynamics. In Chapter 3, we will introduce the standard linear regression algorithm and an on-line gradient descent (GD) algorithm used in this research. In Chapter 4, we will describe the design and implementation of the artificial data generator. Then we will describe how the GD algorithm and standard linear regression algorithm perform on artificial data. Finally, in Chapter 5, we will conclude the thesis and present some suggestions on future work.

Chapter 2 Background and Review

We start this chapter by discussing the integration of computer science and biology. Then we will talk about some computational approaches, including machine learning and pattern recognition algorithms. After that, we will introduce basic concepts of microbial population dynamics and different growth models. At the end of this chapter, we will describe possible trends for microbial population dynamics research driven by recent development of computer and molecular biology techniques.

2.1 Integration of Computer Science and Biology

With the continuing collaboration between computer science and biology in recent years, a new area called Bioinformatics has emerged. Bioinformatics is about how to manipulate and interpret the enormous volumes of data generated from experiments in areas such as genomics, proteomics, structural biology, and dynamics of macromolecular assemblies, which will lead to a better understanding of life. Bioinformatics is defined as the computational systems used to collect, store, and analyze biological information. These include software systems used to sequence DNA, database systems used to store and organize the data, and application software used to align sequences and pose queries, find genes, predict protein structure, etc. The recent completion of sequencing the human genome is one of the biggest achievements in science, as well as in human history and it really brings *in silico* biology and *in vivo* and *in vitro* biology together for knowledge discovery in complex biological systems.

The early applications of computers to molecular biology involved graphical representation of molecular structure, modeling and simulation of biological systems and database creation of biological sequences. However, with the recent fast development of

information technology, computer applications to molecular biology are far beyond merely storing, describing and retrieving. The emphasis is now turning to knowledge discovery from those enormous data sets. Thus bioinformatics is actually the study of how modern information technologies are used to solve problems in biology. Some have pointed out that bioinformatics actually studies two important information flows in modern biology. One is the flow of genetic information from DNA up to distinct organism characteristics, and the other is the flow of experimental information to models that can explain them, and to new experiments to test them [6]. Nevertheless, such research is still based on the central theorem in biology, which actually is the flow of the genetic information: genetic information is stored in DNA sequences (RNA sequence in some species), which is translated into sequences of proteins, etc. Through the evolution and natural selection, information can be passed back to DNA, causing different gene codes.

2.2 Why Machine Learning Approaches

Biological systems are the most complicated systems in the world. With the fast development of modern techniques, especially in biological sequencing, huge amounts of biological information are being generated, which makes them nearly impossible to process by hand. However, machine learning algorithms are ideally suited for domains characterized by the presence of large amounts of data, “noisy” patterns, and the absence of the general theory [1].

Machine learning is a subarea of artificial intelligence that studies how computers can learn from experience, improving their performance without human intervention. Machine learning itself is also a multidisciplinary field. It draws on concepts and theories from many other fields, including computational complexity theory, information theory, probability and statistics, artificial intelligence, philosophy, psychology, neuroscience, etc. The inherent

interdisciplinary nature of machine learning makes it an ideal approach for various applications. For speech recognition, algorithms based on machine learning are the best algorithms so far, outperforming all other attempted approaches [2]. Machine learning methods have also been used to predict recovery rates of pneumonia patients [7]. Machine learning methods also perform very well in playing games such as backgammon against world-class human players [8,9]. Also, machine learning methods are routinely used in detection of the fraudulent use of credit cards [3]. In fact, machine learning methods are especially useful in data mining problems where large databases may contain valuable implicit regularities that can be discovered automatically; poorly understood domains where humans might not have the knowledge needed to develop effective algorithms and areas where the program must dynamically adapt to changing conditions. Those features are true in modern biological research, especially in bioinformatics. Although through the past twenty years we have gathered much more knowledge than before because of the development of advanced experimental techniques, there still remains a lot unknown at different levels, such as the molecular level, the cellular level and the population level. Thus, a high degree of uncertainty and many mysteries still remain in many biology problems. People still haven't developed general theories on various biological problems, for example, the protein folding problem, mechanisms for cancer gene initiation and activation and the immune system. However, at the same time, because of ongoing large-scale sequencing projects, accompanied with developments on proteomics and structural biology, many large biological databases containing enormous information are available on the web, such as GENBANK (the DNA databank), SWISS-PROTPDB (the protein sequence databank) and PDB (the protein 3-D structure databank). A goal of machine learning is to extract useful information from large sets of data to build some appropriate model. Machine learning

algorithms, including artificial neural networks, Bayesian learning and other pattern recognition algorithms such as clustering algorithms are currently being actively developed and applied in many computational biology projects.

2.3 Clustering Algorithms and Pattern Recognition

Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes. Originally, pattern recognition lay in theoretical statistics. The advent of computers has dramatically increased the demand for practical applications of pattern recognition, which has led to further theoretical development. Pattern recognition has become an integral part in most machine intelligence systems built for decision making [10].

The application of pattern recognition is as wide as that of machine learning. In fact, approaches taken from both fields are integrated and applied in numerous tasks at most times. Image processing, character recognition, text retrieval, computer-aided medical diagnosis and speech recognition are some main examples of pattern recognition applications.

Different tasks require different pattern recognition approaches. Two main types of pattern recognition are supervised pattern recognition and unsupervised pattern recognition. If a set of training data was available and the classifier was designed by exploiting such prior knowledge, this is known as supervised pattern recognition. On the other hand, if there is no training data with known class labels available, this is known as unsupervised pattern recognition or clustering.

Since clustering results depend on the specific algorithm and the criteria used, a clustering algorithm is also considered as a learning procedure that tries to identify the specific pattern or characteristic underlying the data sets. A clustering algorithm may be

divided into several major categories. Sequential and hierarchical clustering algorithms are the two main categories. Sequential clustering algorithms are quite straightforward and fast methods that produce a single clustering. Hierarchical clustering algorithms build a hierarchy of clusterings, revealing nested cluster-based relationships. Hierarchical clustering algorithms can be further divided into divisive algorithms and agglomerative algorithms. Divisive algorithms produce a sequence of clusterings in the opposite direction, i.e. they produce a sequence of clusterings of increasing number of clusters at each step by splitting a previous single cluster into two. Agglomerative algorithms produce a sequence of clusterings of decreasing number of clusters at each step by merging two previous clusters into one.

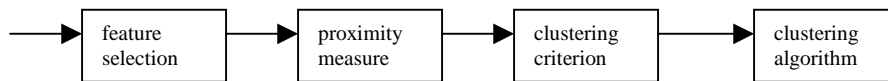


FIGURE 2.1: The basic stages involved in the design of a clustering task

Figure 2.1 shows the various steps involved in developing a clustering task. The first step is feature selection, which is to select the appropriate characteristics of the task (called *features*) so as to encode as much information as possible. Proximity measure is the next step, i.e. find a good measurement that can be used to compute the similarity or dissimilarity between any two objects in the clustering task. Different clustering criteria will result in different clustering outputs. For example, we can group microorganisms into different results based on different clustering criteria, such as nutritional requirements, oxygen dependency, etc. The final step refers to the choice of a specific algorithmic scheme that unravels the clustering structure of the data set after having adopted a proximity measure and a clustering criterion.

A feature may take values from a continuous or a discrete set. If we represent objects in the clustering task as feature vectors, we are able to view the similarity or dissimilarity as mathematical function. For example, the well-known *Euclidean distance* is a dissimilarity measure. Another important dissimilarity measure is the *Hamming distance*, which we used in this project. Because clustering approaches allow us to discover similarities and differences among patterns and to derive conclusions for decision-making by grouping instances into different clusters, the applications of clustering algorithms are very broad. Many clustering algorithms, especially hierarchical clustering algorithms, are routinely applied in life science, medical science and social science [11,12]. Recent advances in DNA sequence analysis and DNA microarray technology are for the first time offering researchers comprehensive snapshots of cells' genetic mechanisms [13]. Clustering has become a general strategy to extract useful information from microarray data, such as phenotype classification [14], genetic network modeling [15] and gene expression analysis [16,17].

In this research we chose a hierarchical clustering algorithm (commonly used in biological taxonomy and other fields in modern biology) to identify homogenous clusters (*groups*) from DNA fingerprinting binary vectors. Hierarchical clustering algorithms produce a hierarchy of *nested* clusterings. A clustering R_1 containing k clusters is said to be *nested* in the clustering R_2 , which contains r ($<k$) clusters, if each cluster in R_1 is a subset of a set in R_2 and at least one cluster of R_1 is a proper subset of R_2 . For example, the clustering $R_1 = \{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5\}\}$ is nested in $R_2 = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}$ [10].

Recall the definition of dendrogram that is derived from clustering process discussed in Section 1.3. At the highest level, the data all lie in one cluster. As the tree branches out, clusters are subdivided into smaller clusters in such a way that similar data points remain together in the cluster and dissimilar points are separated. In this research, we start with each

datum of X (the set of binary vectors) in its own cluster and agglomerate the clusters in pairs. This procedure is categorized as an *agglomerative hierarchical algorithm*. Each time a pair of clusters is merged into one, the number of clusters is reduced by one. The final cluster contains the single set X , that is, the initial set of binary vectors. These algorithms involve N steps, as many as the number of data vectors.

Let $d(C_p, C_j)$ be a function defined to measure the distance between C_i and C_j for all possible pairs of clusters of X . $d(C_p, C_j)$ will be represented by entry (i, j) in a dissimilarity matrix. Let t denote the current step of clustering. The general agglomerative algorithm (GAS) may be stated as follows:

1. Initialization:

1.1 Choose $R_0 = \{C_i = \{x_i\}, i=1, \dots, N\}$ as the initial clustering.

1.2 $t=0$.

2. Repeat:

2.1 $t=t+1$

2.2 Among all possible pairs of clusters (C_p, C_j) in R_{t-1} find the one, say (C_p, C_j) , such that $d(C_p, C_j)$ is minimized

2.3 Define $C_q = C_i \cup C_j$ and produce the new clustering $R_t = (R_{t-1} - \{C_p, C_j\}) \cup \{C_q\}$.

3. Until all vectors lie in a single cluster

In this research, since we already built a dissimilarity matrix, we adopted a version of GAS, called the Matrix Updating Algorithmic Scheme (MUAS) [11]. The initial input is the $N \times N$ dissimilarity matrix M_0 and M_0, M_1, \dots, M_n is a series of distance matrices with entry $(i, j) = d(C_p, C_j)$. When two clusters are merged into one, the size of matrix M_t becomes $(N-t) \times (N-t)$. The matrix is updated at each step by (a) deleting the rows and columns

corresponding to merged clusters and (b) adding in a row and column giving the distance between the new cluster and the remaining clusters. The two clusters to be merged at each step are found by searching the matrix for the smallest distance between clusters, i.e. the minimum off-diagonal entry. The MUAS clustering algorithm can be formulated as follows:

1. Initialization:

$$1.1 \quad R_0 = \{\{x_i\}, i=1, \dots, N\}.$$

$$1.2 \quad M_0^{ij} = d(C_p, C_j)$$

$$1.3 \quad t=0.$$

2. Repeat:

$$2.1 \quad t=t+1$$

$$2.2 \quad \text{Find } C_p, C_j \text{ such that } d(C_p, C_j) = \min_{r,s=1, \dots, N, r \neq s} d(C_r, C_s).$$

$$2.3 \quad \text{Merge } C_p, C_j \text{ into a single cluster } C_q \text{ and form } R_t = (R_{t-1} - \{C_p, C_j\}) \cup \{C_q\}.$$

$$2.4 \quad \text{Update the matrix } M_{t-1} \text{ into } M_t \text{ in equation (2.1).}$$

3. Until all the binary vectors lie in the same cluster.

The weighted pair group method average (WPGMA) algorithm is used to update the matrix value (the distance between each pair of vectors), which is obtained according to the following equation:

$$d(C_q, C_s) = \frac{1}{2}(d(C_i, C_s) + d(C_j, C_s)). \quad (2.1)$$

Thus, in this case the distance between the newly formed cluster C_q and an old one C_s is defined as the average of distance between C_i and C_s and C_j and C_s .

2.4 Microbial Population Dynamics

Application of dynamics system analysis methods and the formulation of mathematical models are playing an increasing role in investigating complex biological systems. The behavior of the whole system depends not only on the behavior of the individual components, but also on the manner of their interactions. Viewing the system as dynamic can reflect the real property of the system. Microbial population dynamics analysis is very important for understanding microbial diversity and evolution by modeling and investigating the fluctuations, performance and distribution of different components in the population. Also, it is essential for better control and management of different microorganisms.

Normally a single population is referred to as a group of organisms of one species. For microbial populations, one population may refer to the same species, but include several different strains. Not only do we want to know some obvious features such as size, life-cycle, abundance and growth function, but we also want to learn something of their genetic makeup and pattern in the population. In this respect, population dynamics deals with the study of changes of individual component of the population, the factors influencing these changes, and the interactions among these individual components. Some sample questions that may be asked in microbial population dynamics studies are:

1. What is the general structure of this particular population, how many individual components exist in this population? What is the dependence of the population on certain variables, such as time, temperature, etc.?
2. Why are some strains scarce and others abundant? Is any relationship between different strains?

3. Is any rapid change in abundance of some specific strains due the introduction of external variables, such as some antimicrobial agents?

Figure 2. 1 shows a possible scenario of the microbial population dynamics.

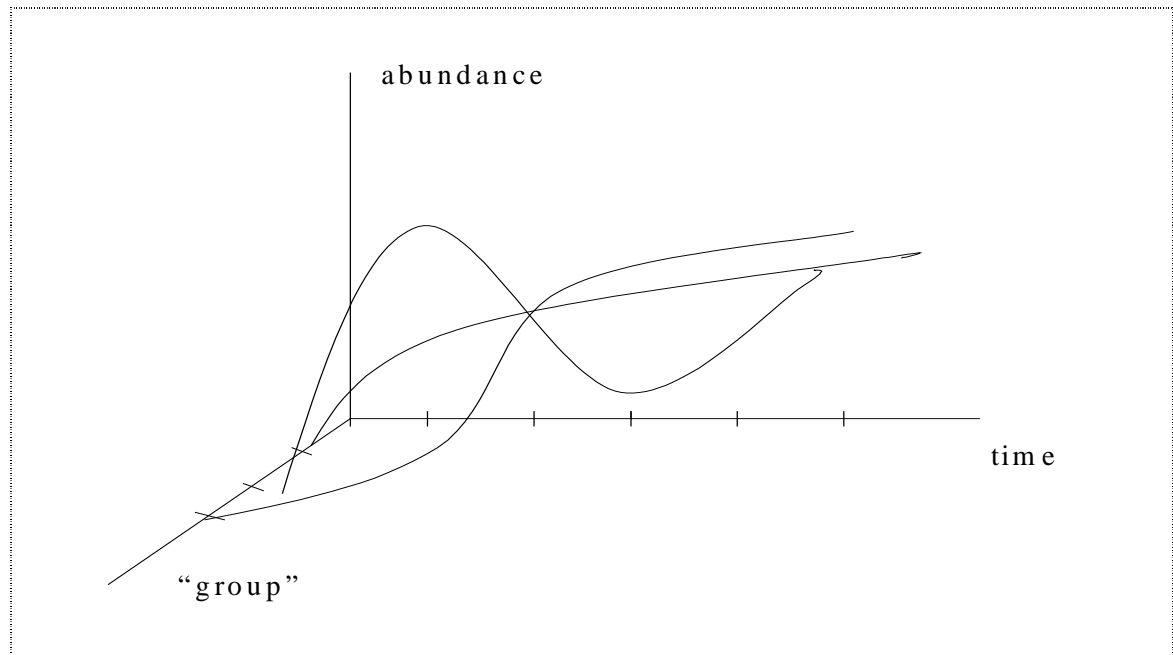


Figure 2.1 Example of Microbial Population. The abundance of different homogenous groups in the population changes over time. To model such behavior and predict the future change is one of the main goals for microbial population dynamics analysis.

2.4.1 General Population Dynamics Models

Models for describing patterns of growth and decline or decay in biological populations and individual organisms has received extensive attention in population dynamics research. Traditionally, there are three different classes of models widely used in research in terms of different research emphases and methods [18].

A. Segregated vs. Distributed Models

Since in microbial systems the functional unit is generally the individual cell, those mathematical models considering such segregation with discrete units or cells are referred to

as segregated models. On the other hand, measurement of a microbial population based on some component concentrations, which are assumed to be distributed uniformly over the living space, is referred to as a distributed model.

B. Structured vs. Unstructured Models

A second basis for differentiating microbial models is based on whether they provide for a measurement of the physiology or structure of the population [19,20,21]. Structured models assume that there are structural differences between similar cells, while the unstructured models ignore such differences and assume that the biological population responds immediately to environmental changes. The structured model is much more complicated and difficult to use than the unstructured model, although the structured model can better represent the real world.

C. Deterministic and Stochastic Models

Deterministic models predict a direct cause-and-effect relationship, while stochastic models deal with randomness based on probability. Most simple models are deterministic.

The choice of the type of the model is a tradeoff between more realistic modeling and easy mathematical representation. Although real populations and individuals are subject to random influence, simplifying the situation and ignoring some complication is always the normal starting point. Exponential growth and decline are the basic models for describing the change in size of biological populations. Population dynamics can be described either in terms of discrete time or in terms of continuous time. In a microbial population, changes appear in continuous time. For example, the bacterium *Escherichia Coli*, an inhabitant of the human intestine, is capable of very fast growth through cell division or fission, which may involve millions of cells. However, those two approaches are interchangeable simply by some mathematical transformation, such as a Laplace transform [22]. Based on the continuous

time model, successive microbial population sizes over a small increment of time are represented as:

$$S(t + \Delta t) = S(t) + \Delta S,$$

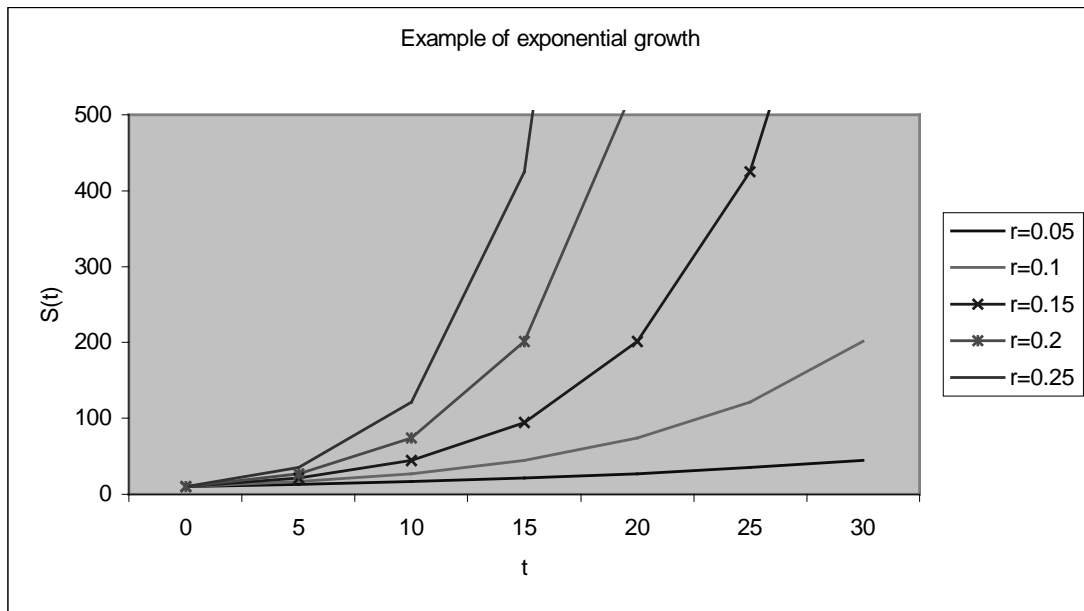
and the growth increment can be derived as:

$$\Delta S = \eta S \Delta t,$$

where η is referred to as the rate constant of growth or relative growth rate. By mathematically solving the differential equation, the exponential growth function is obtained as:

$$S(t) = S(0)e^{\eta t},$$

where $S(0)$ is the population size at time $t=0$. The important property of exponential growth is that the logarithm of population size increases as a straight line with slope equal to the rate constant.



Figures 2.2 Examples of exponential growth in continuous time for different values of the rate constant and initial population size. Here $S(0)=10$ and $r=0.05$ (bottom), 0.10, 0.15, 0.20 and 0.25 (top)

It is well known that any population increase cannot continue indefinitely. There is an upper bound on the maximum population density because of shortage of resources and lack of space, which may be further exacerbated by increased competition. Eventually, these environmental changes may prevent population growth because of increased mortality rate or reduced reproduction rate. There will be no possible further increase whenever the numbers reach a saturation point. The idea of limited population growth was formulated by T.R.Malthus in 1798. Later, in 1859 Charles Darwin adopted this important idea to develop his famous theory of evolution through natural selection [23]. The characteristic of limited population growth is that a relative growth rate is not constant but decreases as the population size increases. As a generic property, most microbial growth curves can be simply characterized by the following three features: the maximum specific growth rate μ_{\max} , the lag time λ , and the asymptotic value A . Among these models, a particular case is the logistic

model, which has been widely applied in theoretical and experimental study of population growth. Microbial growth can be modeled as a function of some variables, such as time and temperature. In the case of time function, if the logarithm of the relative population size is defined as a function of time t , then the typical sigmoidal or S-shaped curve is obtained.

The simplest case is the logistic model in which the relative growth rate declines linearly with population size so that the rate change of population size is then given by the differential equation:

$$\frac{dS}{dt} = \eta S \left(1 - \frac{S}{K}\right).$$

The parameter η is referred to as the initial relative growth rate, which can be also thought of as the rate constant for the potential exponential growth of the population. The parameter K is called the carrying capacity or equilibrium level. When the population size reaches K , the growth rate is zero and the population remains at this level. Solving the differential equation gives the logistic equation for population growth:

$$S(t) = \frac{K}{1 + e^{-\eta(t-h)}},$$

where h is the time at which the population reaches half the carrying capacity.

The temperature is also very important factor for microbial growth. Many models have already been developed, such as the growth-temperature model with the equation:

$$\sqrt{\mu_{\max}} = b_1(T - T_{\min}),$$

where b_1 is a parameter and T_{\min} is the theoretical minimum temperature [24]. Another model is the lag time-temperature model with a hyperbolic function:

$$\ln(\lambda) = \frac{p}{T - q},$$

where p and q are two parameters and T is the time [25].

Because microorganism growth is influenced by many environmental factors such as temperature and pH, the growth model parameters are commonly determined by multiple linear regression. However, growth models are always under considerable debate because of the inherent complexity. Those environmental as well as biological factors may be population density-dependent, or inverse to population density, or may be population density-independent. As a consequence, many populations are subject to considerable fluctuation in abundance. Some are changing rapidly, some slowly, some strikingly, some rather little. Sometimes, one particular factor can be identified as the chief cause of the fluctuation. Furthermore, some effects are direct, while others are indirect. For example, antibiotics may directly impact certain bacteria by killing some of them, or it may affect others because of the certain coexisting bacteria killed by the antibiotic or just because of some changes in the environment.

Another important feature in microbial population dynamics is population interaction, which is based on the influences of individuals or groups within a population (intraspecific), and on the influences of populations of different species (interspecific). Deeply understanding the interaction relationships among different microbial population components is the key to modeling and predicting the changes and distribution of the whole

population. Also, it is the key to realizing the functional dependency between different population elements. It also provides a new way to monitor and control interested individual strains in the population. Because of that, the realm of microbial population dynamics modeling has been extended to much more complex systems and with a wide range of environmental conditions. In these situations, many basic modeling approaches are joined together to form a very large and complex system.

2.4.2 Current Trends of Microbial Population Dynamics Analysis

Other significant impacts on microbial population dynamics research in recent years came from the fast development of molecular tools, which dramatically change the focus of the research. Traditionally, as discussed above, many microbial population dynamics studies focused on modeling and predicting growth dynamics in terms of biological or environmental variables such as time, temperature, nutrients, etc. Little research has been done at the gene level, such as the changes of a microbial genotype over a period of time, the interaction relationship among genotypes in the same population, and the changes of microbial genotype over some introduced variables such as antimicrobial agents. The reason may come from several aspects. First, traditional identification of microbial genotype or strain involves a process of isolation and culture, which is not only time-consuming, but also sometimes it is impossible to culture certain microorganisms. The fact is that less than 1% of microbial diversity has been cultured in the laboratory. Without knowing the exact microbial genotype or strain, we cannot go further to investigate their relationship in the population at the genome level. Second, although some approaches have been tried in different situations, there is still controversy because of the inherent complexity of such systems. Molecular biologists created a revolution by introducing techniques such as PCR (Polymerase Chain Reaction), fingerprinting, etc., to identify microbes. By applying techniques of molecular

biology and phylogenetic analysis of DNA or RNA sequences, we can circumvent the requirement for cultivation of the microbial diversity and we are able to investigate population dynamics directly at the molecular level by genotyping [26,27]. We are also able to cluster similar DNA sequences into homogenous groups and study the patterns of those groups' distribution in nature by examining changes of the genomic content and corresponding homogenous groups in the population. Such population dynamics depend on not only individual growth and decline, but also involve gene transfer, mutation and selection.

At the same time, because of the development of computer science and technology, researchers now are able to analyze more data than ever before, which allows us to consider and model biological systems in more complex ways. Such tendencies will lead to more accurate models. Microbial population dynamics analysis now extends to nearly every level, for example, the genome level, which can be applied to DNA polymorphism modeling and molecular evolution [28] and the cellular level, which can be used to model cell division and branching processes [29]. Nevertheless, microbial population dynamics analysis is very broad field and many more models and algorithms will be applied in order for better understanding the underlying mechanism and characteristics.

Chapter 3 Data Analysis Approaches

In this chapter, we will talk about two approaches applied in this research on artificial abundance data. First, we will discuss the standard linear regression method, which is to fit a straight line to the data to minimize squared error. We also talk about the importance of data transformation on some data to find the possible relationship between abundance data and time points. Then we will describe the general idea of the on-line gradient descent (GD) algorithm and how GD is incorporated into this research to find the best regressor based on our artificial abundance data model.

3.1 Linear Regression Analysis

Observations on biological populations and processes often involve some degree of random variation. This variability tends to obscure real differences. Some statistical methods comparing population parameters such as the mean and variance are typically the first approach to detect differences among the random samples. Regression is a routine statistical method for finding the relationship among different variables in the system. Regression is a technique for describing how one variable varies with the values of other variables. What we seek to define is what is commonly termed the "best fit" line through the data. The criterion for "best fit" that is generally employed utilizes the concept of *least square error*.

Let us tentatively assume that the regression line of variable Y on variable X has the form $\beta_0 + \beta_1 X$. Then we can write the linear, first-order model

$$Y = \beta_0 + \beta_1 X + \varepsilon.$$

That is, for a given X , a corresponding observation Y consists of the value $\beta_0 + \beta_1 X$ plus an amount ε , the increment by which any individual Y may fall off the regression line. Although β_0 , β_1 and ε are unknown and in fact ε would be very difficult to discover since it changes for

each observation Y . However, we can derive the *estimates* b_0 and b_1 of β_0 and β_1 . Thus we can write

$$Y' = b_0 + b_1X,$$

where Y' denotes the predicted value of Y for a given X , when b_0 and b_1 are determined. Suppose we have n sets of observations (X_1, Y_1) , (X_2, Y_2) , ... and (X_n, Y_n) , then the sum of squares of deviation from the true line is

$$S = \sum_{I=1}^N \varepsilon_I^2 = \sum_{I=1}^N (Y_I - \beta_0 - \beta_1 X_I)^2.$$

S is also called the *sum of square function*. We shall choose the estimates b_0 and b_1 to be the values that, when substituted for β_0 and β_1 , produce the least square possible value of S . We can determine the b_0 and b_1 by differentiating S first with respect to β_0 and then with respect to β_1 and setting the results equal to zero. So the estimates b_0 and b_1 are solutions of the two equations

$$\sum_{i=1}^n (Y_i - b_0 - b_1 X_i) = 0,$$

$$\sum_{i=1}^n X_i (Y_i - b_0 - b_1 X_i) = 0.$$

A straight line relationship may also be valuable even when we *know* that such a relationship cannot be true, because a straight line relationship might provide a perfectly adequate representation *in some range*. However, the relationship could be used for predictive purposes outside the range.

Another important issue is how to measure the association between variables that are ordinal or continuous. The widely used measure is the linear correlation coefficient. For pairs

of quantities (x_i, y_i) , $i=1, \dots, N$, the linear correlation coefficient r (also called the product-moment correlation coefficient, or Pearson's r) is given by the formula

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}},$$

where \bar{x} is the mean of the x_i 's and \bar{y} is the mean of the y_i 's.

The value of r ranges from -1 to 1 , inclusively. When $r = 1$, it means complete positive correlation with data points lying on a perfect straight line with positive slope, with x and y increasing together. When $r = -1$, it means complete negative correlation with data points lying on a perfect straight line with negative slope, with y decreasing as x increases. A value of r near zero indicates that the variables x and y are uncorrelated.

In this research, we want to evaluate how the standard linear regression and the gradient descent algorithms (described below) perform on artificial data generated by the exponential growth model. It is known if we just use the original abundance data, we won't be able to find a line to fit the data. Because of this fact, we use $\ln Y$, the natural logarithm of Y (logarithm to the base e , or \ln) instead of Y . By such transformation, we might find that a simpler planar equation gives a better, as good, or almost as good an explanation of the relationship between the abundance data and time point.

3.2 Gradient Descent for Finding the Best Regressor

In addition to the regular regression analysis based on least squares theory, several machine learning algorithms such as the Gradient Descent (GD) algorithm, the Normalized Exponentiated Gradient (EG) algorithm and the Un-normalized Exponential Gradient (EGU) algorithm can be applied to find the best regressor [30]. Those algorithms are known as on-line learners, which are algorithms that learn their hypotheses (regression coefficients)

continuously during the training. Those algorithms have performance guarantees bounds on total square loss and can tolerate *concept shift*, which is when the best regressor changes over time. This is important in this research because the growth model has different phases. In this project, we want to evaluate the empirical performance of GD by applying it to different series of artificial *E. coli* abundance data generated by the procedure discussed in Section 4.1. GD is one of the oldest optimization algorithms, which can be used to search the hypothesis space of possible *weight vectors* (regression coefficient) to find the weights that best fit the training examples. A weight vector is a real-valued vector that determines the contribution of input to the output value. The prediction is a function of the dot product of input and weight vector. Suppose the learning proceeds in *trials* $t=1,2,\dots,N$. GD maintains a weight vector denoted by W_t . At each step the algorithm receives a training example (X_t, Y_t) . It will produce some prediction for instance X_t based on W_t . By comparing the prediction value with the real label Y_t , GD updates W_t in order to minimize the difference. Such a difference is called *training error* or *loss*. This kind of learning procedure is called *on-line* learning. Although there are many ways to define the training error, the square loss is frequently used because it is convenient. The square loss $E(w)$ can be stated as follows:

$$E(w) = \frac{1}{2} (\sigma(w_t x_t) - y_t)^2.$$

By changing the function σ , the hypothesis class varies. Whenever $E(w)$ is differentiable, one can try to find its minima by the following iterative procedure:

$$W_{t+1} = W_t - \eta \frac{\partial E}{\partial W_t}$$

where η is the learning rate, which is to moderate the degree to which weights are changed at each updating step.

In this research, we use a variation of GD because different growth phases occur for the exponential growth model, which implies different predictors may be best for different parts of the time series. *Concept shift* refers to the changing over time of the best regressor. Herbster proposed a variation of GD called Constrained Generalized Gradient Descent (C-GD) to handling the shift case. The basic idea of their technique is to project the weight vector into a suitably chosen convex region at the end of each trial. Then in this modified GD algorithm, two updates exist. The first is the general gradient descent update, and the other is the projection update. The projection of a weight vector W_t onto a closed convex set S is the point in S closest to W_t . The convex region plays a key role in bounding the size of weight vector. Such a constraint set S is highly related to the loss bounds of the original algorithm. The bound of GD has been proved to depend on the 2-norm of both the instance X_t and predictor, i.e W_t [31].

In this research, we are interested in how C-GD performs on finding the best linear regressor on the *E. coli* genotype abundance data. The weight vector W has the form of $\{W_0, W_1\}$. W_0 and W_1 correspond to the intercept and the slope of the derived linear regression line, respectively. The training example has the form $\{X, Y\}$, X is the vector of the input value, which is the time value. Y is the natural logarithm of the abundance data. E_w is the square loss function. The constraint set S has the form $\{W: \|W\|_2 \leq \lambda\}$. λ is a constant parameter that can be carefully tuned to assure the 2-norm of the predictor W located in the convex region. C-GD used in this research can be stated as follows:

1. Initialization
 - 1.1 Initialize the weight vector to some random value.
 - 1.2 Initialize the learning rate η and λ .
2. Repeat
 - 2.1 For each training example
 - 2.2 Compute the square loss:

$$E(w) = \frac{1}{2} (w_t x_t - y_t)^2$$

- 2.3 General GD update:

$$W_t = W_t - \eta \frac{\partial E}{\partial W_t}$$

- 2.4 Projection GD update:

$$W_{t+1} = \begin{cases} W_t & W_t \in S \\ \frac{\lambda W_t}{\|W_t\|_2} & W_t \notin S \end{cases}$$

3. Until all training examples been processed.

We applied C-GD on the abundance data generated by the exponential growth model without the concept shift, i.e. only exponential growth phase exists. Also, we applied C-GD on the abundance data generated by the exponential growth model with concept shift, i.e. a saturation phase follows the exponential growth phase. In this situation, there was a single line that fit the exponential growth phase and a separate line fitting the saturation phase. We investigated how different learning rates η affect the performance of C-GD and

how many time points needed for C-GD to converge to the best weight vector under those different situations.

Chapter 4 Experimental Results and Discussion

In this chapter, we first discuss the artificial data generation process in detail. Then we present some empirical results obtained from the standard regression analysis and GD, which were applied on some abundance data generated from the artificial data generator.

4.1 Growth Model Implementation (Artificial Data Generation)

One of the contributions of this thesis is an artificial abundance data generator based on known growth models. Recall the starting file contains real *E. coli* strain genetic information from two diet-controlled cows at ten time points, with twenty isolates at each time (Section 1.3). So our datasets actually have three parameters: the number of animal hosts (in this case, the number of cows), the number of time points, and the number of isolates at each time point. Thus a total $2 \times 10 \times 20 = 400$ strings of binary data from the starting file serve as the template for artificial data generation.

Those three parameters actually define the total number of bit strings we need to generate for the new starting file. For each possible bit string in the new starting file, we select one bit string uniformly at random from the existing animal starting file and copy it into the new file. Then in order to better reflect reality, we flip each bit with probability p (in this research we set $p = 0.2$). With that approach, we are able to assure that when we introduce some degree of variation in the new starting file from the real data set, yet we preserve some degree of stability because it is highly unlikely to find a dramatic change in *E. coli* strain population under normal conditions unless mutation occurs by some influencing factors, such as antimicrobial agents.

The clustering procedure discussed in Sectiond 1.3 and 2.3 identifies different groups from the new data file and generates a dendrogram. We trace through the dendrogram to

compute the abundance value for each group at the starting time point. For the abundance value at successive time points, different growth models were randomly assigned to one specific group. In this research, three general classes of growth models are used. One is an exponential growth model, one is a linear decline growth model and the last one is a stagnant growth model. The linear decline growth model means that a group's abundance decreases approximately linearly after the starting time point. The stagnant growth model means that a group's abundance remains approximately the same over the time. The exponential growth model has multiple variations. One is exponential growth over all the time points. Another is that the growth becomes stagnant after the exponential growth phase. Finally, either growth remains stagnant or a linear decline phase begins. There are two ways implemented to define the saturation point. One is to specify the saturation abundance value. Whenever the abundance increases up to this value, the stagnant phase starts. Another is to explicitly specify the time point as a saturation point. The stagnant phase starts from this time point.

The *E. coli* growth rate for the exponential growth model is determined by results from biological experiments conducted in Dr. Andrew Benson's Lab in Department of Food Science and Technology (University of Nebraska-Lincoln). With the growth rate (4 per time unit) as the mean and some predefined variance, we introduce some noise for the genotype coming from different hosts using a Gaussian distribution. Also, at each time point, we introduce some noise into the abundance data by the Gaussian distribution procedure [32].

Let t be the current time point, A_t be the abundance at time t , and R be the mean growth rate. e_1 and e_2 are two independent noise factors generated according to some Gaussian distribution. The formula for exponential growth may be stated as follows:

$$A_{t+1} = A_t \times (R + e_1) + e_2.$$

The formula for stagnant growth model may be stated as follows:

$$A_{t+1} = A_t + e_1.$$

The formula for linear decline growth model may be stated as follows:

$$A_{t+1} = A_t - C + e_1,$$

where C is a constant.

Since the exponential growth model with saturation actually contains the other two models and because of the simplicity of other two models, abundance data of the group with the exponential growth model will be the main focus of our work.

4.2 Data Analysis Experimental Results

Recall we use the natural logarithm of original abundance data in the data analysis (Section 3.1). In order to investigate the effect of the number of hosts and the effect of the number of times on the standard linear regression analysis methods, we generated different data files with different numbers of time points t ($t=4, 15, 20$). The number of isolates at each time is fixed at 20 same as that in the real data file to keep the same degree of representation of the whole *E. coli* population. Also, if the number of hosts in the data file is more than 1, we use the mean of the abundance data at each time points as the input for the standard linear regression analysis. We explicitly specify the saturation value at 55000, which implicitly defines the saturation point about $t=6$, depending on the growth rate. Results show that when time $t=4$, the standard regressor finds a good fit for the observed data, as illustrated in Figure 4.1. The reason is because the saturation point is not reached in this data set. The natural logarithm of the observed data is a straight line.

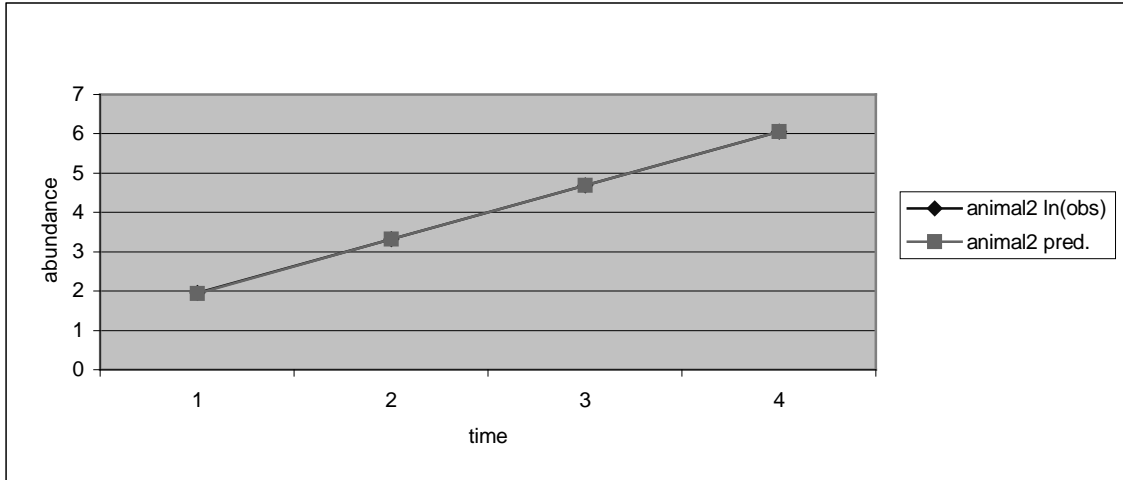


Figure 4.1 Plot predicted value by the standard linear regressor with observed data with 4 time points. *Animal2 ln(obs)* refers to the natural logarithm of the observation data from the data file with 2 animals; *Animal2 pred.* refers to the predicted value of the abundance by the linear regression.

Increasing the number of time points allows the data generated to reach the saturation phase, which implies that a single straight line won't fit all the data. Figure 4.3 shows that the regression line won't be able to precisely match the original data line with the additional time points.

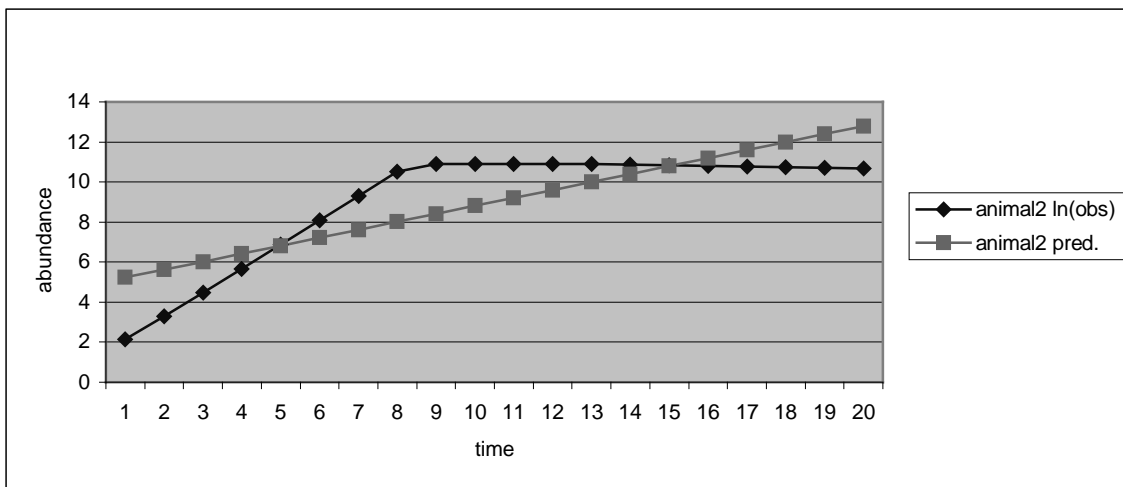


Figure 4.3 Plot of predicted value by the standard linear regressor and observed data over 20 time points. *Animal2 ln(obs)* refers to the natural logarithm of the observation data from the data file with 2 animals; *Animal2 pred.* refers to the predicted value of the abundance by the linear regression.

For the data file with 2 animals and 40 time points, we randomly generated 30 data files. We plot the *average square loss* vs. time points in Figure 4.3 (*Average Square loss* is obtained by taking the average square loss from 30 files).

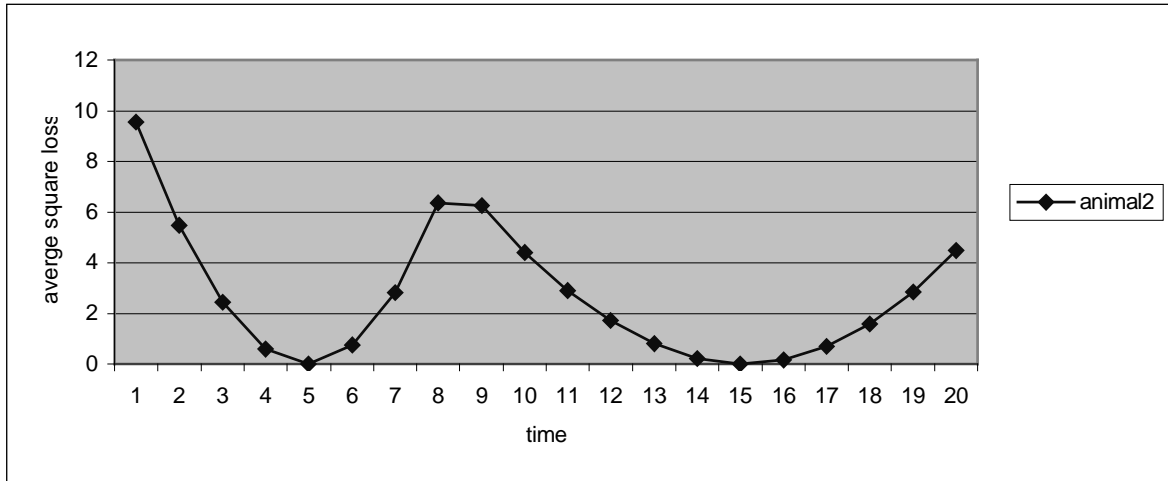


Figure 4.3 The average square loss between predicted value by standard regression and observed data over 20 time points. *Animal2* refers to the data file with 2 animals.

From Figure 4.3, we see that the performance of the standard regressor is poor with the smallest square error loss at $t=4$ and the largest at $t=20$ time point. This is because as the length of the exponential growth phase becomes larger, the abundance reaches the saturation point, which means single line can't fit all the data, as indicated by increasing error loss.

In order to investigate how GD handles the concept shift (i.e best regressor changes over time) and to compare the performance between the standard regressor and GD, we generated data files with a single host and 40 time points. The number of isolates is kept at 20. Let T_e denote the length of the exponential growth phase and $T_s=40-T_e$ represent for the length of the stagnant phase, i.e. after the saturation point. In this case, we vary the value of T_e as 10, 20, 30, and 40 time points. That means T_s is 30, 20, 10 and 0 respectively. For each value of T_e , we randomly generated 30 independent data files. Note that GD updates its

weight vector upon receiving a new data point, i.e. at each step, GD finds a line that best fits the current and previous data points. In order to find the line of the pre- and post-saturation phases that best fits the data points, we define *minimum line square loss* denoted by L as the minimum square loss among all the lines found by GD at each time point. Since the exponential growth model has two phases (the exponential growth phase and the saturation phase), we compute the minimum line square loss for each phase. I.e. we have two minimum square losses for each data files; each one is the best line for its corresponding part. Since the length of each phase varies among different data file, we use the *average minimum line square loss* for each phase denoted by L_e and L_s , respectively. For example, if the data file has $T_e=10$ and $T_s=30$, we divide the minimum line square loss by 10 for L_e and divided the minimum line square loss by 30 for L_s . Also, we take the mean from randomly 100 runs for each case. Recall GD has learning rate η and a constant constraint λ discussed in Section 3.2. For this research, we set λ value as 20000 to minimize the effect of the projection update. Then we varied the learning rate η as 0.0005, 0.001 and 0.002 for different data files. Table 4.1 lists the L_e with different learning rate and different T_e . Results show that if T_e is less than 20 time points, generally increasing the learning rate improves the performance by finding the better line to fit all the data for the pre-saturation part. For example, our experiments show that when $T_e=10$, L_e is 122.8 for $\eta=0.0005$ and L_e decreases to 51.12 and 6.88 for $\eta=0.001$ and $\eta=0.002$, respectively. When $T_e=20$, L_e is 1.31 when η is 0.0005, while when $\eta=0.001$, L_e decreases to 0.014. However, if T_e is larger than 20, increasing the learning rate won't improve the performance much. Actually, the performance even became worse, as indicated by the fact that L_e increased from 0.093 when $\eta=0.001$ to 0.0116 when $\eta=0.002$ for the data file with $T_e=40$. At the same time, increasing the T_s , the length of the exponential growth phase, improves the overall performance of GD, e.g. L_e decreased from 122.8 when $T_e=10$

to just 0.0098 with $T_e=40$. However, when T_e is larger than 30, the effect is not significant, as indicated by $L_e=0.0094$ when T_e is 30 and $L_e=0.0098$ when T_e is 40. That means the improvement of GD performance on pre-saturation part starts to be stable with increasing T_e .

Table 4.1 Average minimum square loss for pre-saturation part (L_e) with different T_e (10, 20,30, 40) and different learning rate η (0.0005, 0.001, 0.002) from 100 runs.

T_e	$\eta =0.0005$	$\eta =0.001$	$\eta =0.002$
10	122.8	51.12	6.88
20	1.31	0.014	0.0146
30	0.0094	0.0083	0.0109
40	0.0098	0.093	0.0116

For the post-saturation part, increasing the learning rate doesn't appear to have a significant improvement on GD's performance. Actually the performance starts to slightly decrease as indicated by Table 4.2. That means for post-saturation part, lower learning rate would be better in terms fo GD's performance.

Table 4.2 Average minimum square loss for post- saturation part (T_e) with different T_e (10, 20,30, 40) and different learning rate η (0.0005, 0.001, 0.002) from 100 runs.

T_e	$\eta =0.0005$	$\eta =0.001$	$\eta =0.002$
10	19.23	19.41	19.72
20	25.64	25.64	26.32
30	11.44	13.59	15.7
40	N/A	N/A	N/A

However, we were surprised by the improvement in performance on post-saturation part with increasing T_e from 20 to 30, as indicated by the decline of the average minimum square loss from 23.75 to 11.44 when learning rate is 0.0005. Intuition suggests that a decreasing number of data points would degrade the performance in general. The reason for this kind of performance improvement is maybe because at some point, especially in the post-saturation period, if the weight vector found by GD at pervious point is too away from the current observation, GD tends to offset such effect by adjusting the current weight vector aggressively, which is heavily influenced by larger learning rate.

Figures 4.4, 4.5, 4.6 and 4.7 show the learning process by GD for the data file with learning rate at 0.0005 and different T_e (10, 20, 30, 40). We also include the predicted line by GD and standard regression method in the plot. Those figures show that for the data file with 40 time points, both methods find a nearly perfect line for the original abundance data. However, the line found by standard regression analysis is a better fit than the one found by GD, with the square error loss at only 0.00005 for standard regression analysis, compared to the square error loss at 0.011 for GD. For other data files with 30, 20 and 10 time points, the GD performs better than standard linear regression because it can handle the concept shift and at the same time GD is still able to find a good fit for pre-saturation part.

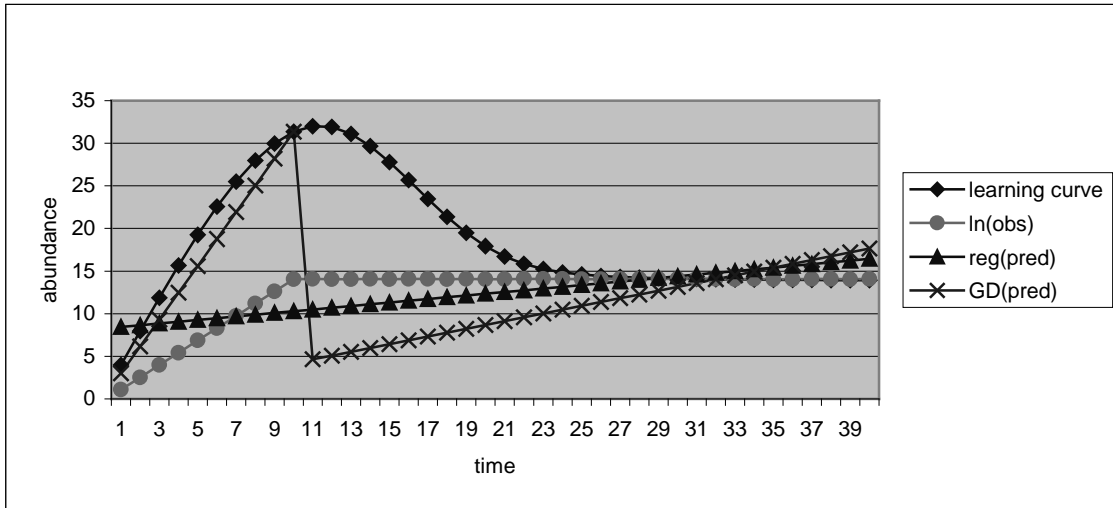


Figure 4.4 The learning process by GD for $T_e=10$ and $\eta=0.0005$. The learning curve refers to the weight vector update process by GD; $\ln(\text{obs})$ refers to the natural logarithm of observed data; $\text{reg}(\text{pred})$ refers to the predicted line by standard regression analysis; $\text{GD}(\text{pred})$ refers to the predicted line by GD.

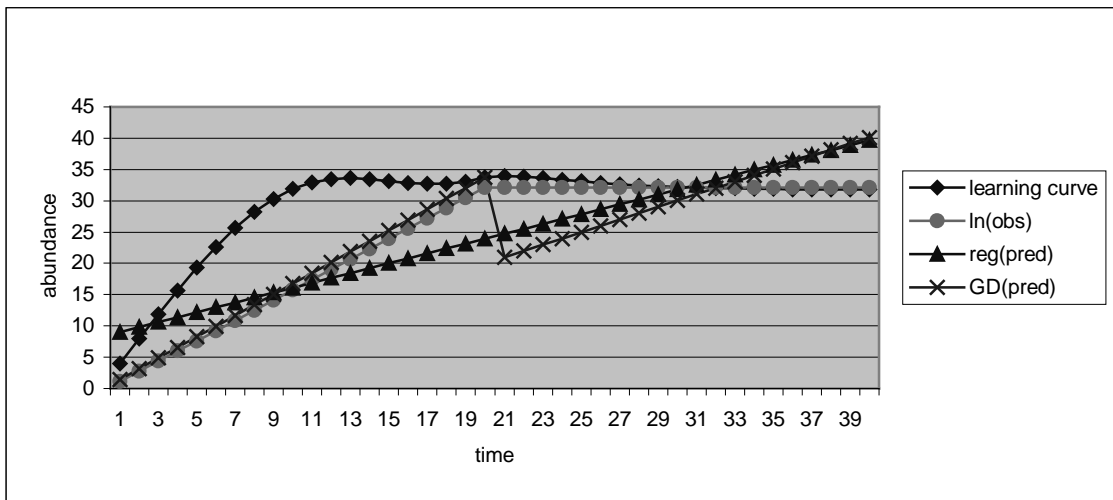


Figure 4.5 The learning process by GD for $T_e=20$ and $\eta=0.0005$. The learning curve refers to the weight vector update process by GD; $\ln(\text{obs})$ refers to the natural logarithm of observed data; $\text{reg}(\text{pred})$ refers to the predicted line by standard regression analysis; $\text{GD}(\text{pred})$ refers to the predicted line by GD.

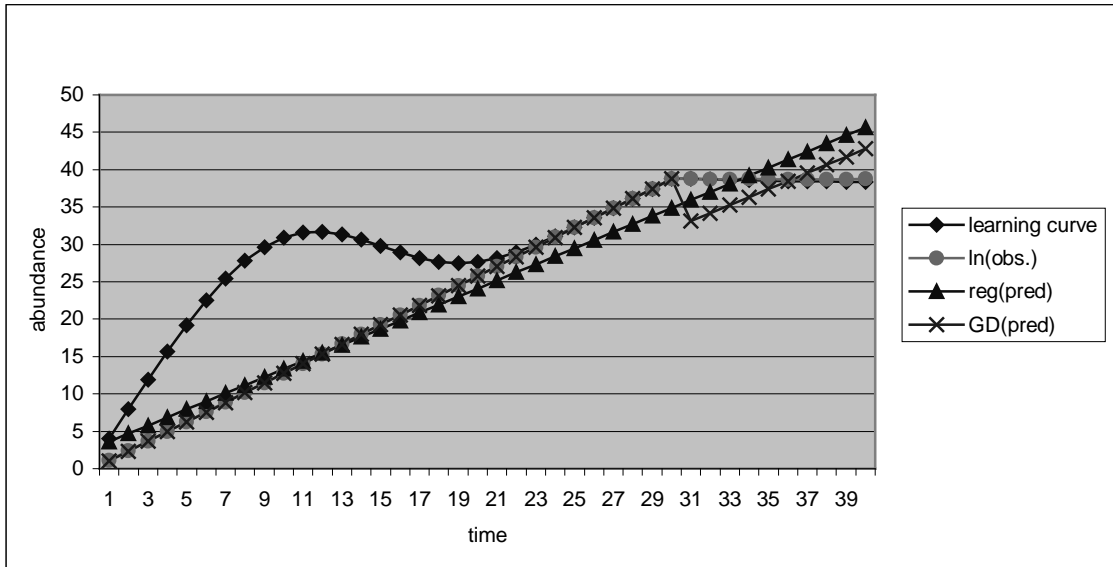


Figure 4.6 The learning process by GD $T_e=30$ and $\eta=0.0005$. The learning curve refers to the weight vector update process by GD; $\ln(\text{obs.})$ refers to the natural logarithm of observed data; $\text{reg}(\text{pred})$ refers to the predicted line by standard regression analysis; $\text{GD}(\text{pred})$ refers to the predicted line by GD.

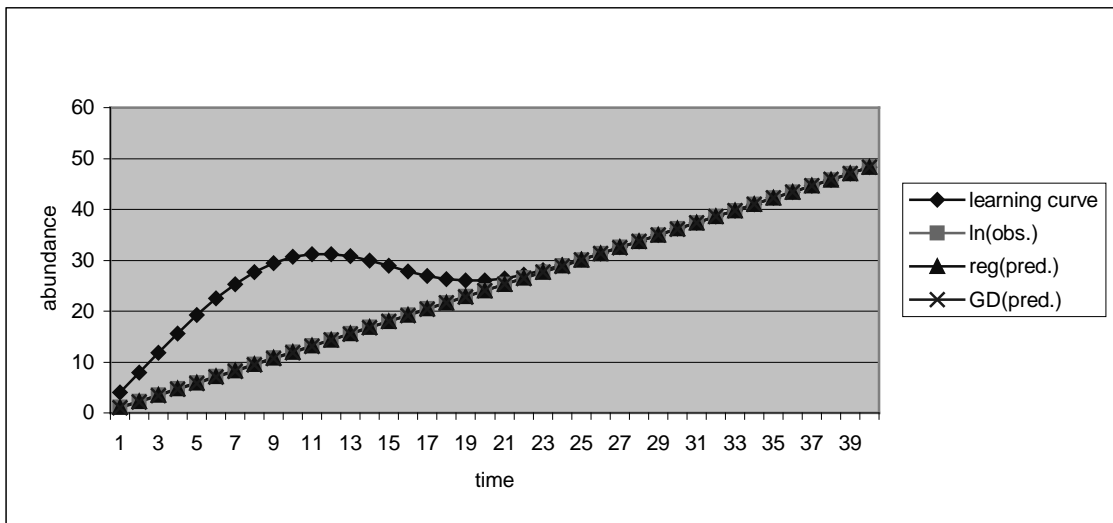


Figure 4.7 The learning process by GD $T_e=40$ and $\eta=0.0005$. The learning curve refers to the weight vector update process by GD; $\ln(\text{obs.})$ refers to the natural logarithm of observed data; $\text{reg}(\text{pred.})$ refers to the predicted line by standard regression analysis; $\text{GD}(\text{pred.})$ refers to the predicted line by GD.

Chapter 5 Conclusions and Future Work

In this thesis, we made an attempt for applying machine learning techniques to microbial population dynamics analysis. Also, we designed a framework based on a clustering algorithm to identify similar groups from bit strings, which corresponds to *E. coli* DNA fingerprinting gel pattern. Because of a lack of data, we also implemented a synthetic data generator based on parameterized mathematical models of microbial population growth and saturation, which can be used to directly compare regressors and classifiers. The artificial data generator consists of two steps. The first step is to generate different starting data files by randomly choosing bit strings from a small real data set and flipping the bit with probability 0.2. The second step is to generate abundance value at successive time points based on some known microbial growth model. Three classes of growth models were implemented in this research. One is an exponential growth model, one is a linear decline model and the last one is a stagnant model. The exponential growth model is the focus of this thesis.

We also implemented various regression algorithms and evaluated their performance on the synthetic data. From the empirical results based on our data generation model, since increasing time points allows the abundance to reach the saturation point, the performance of the standard linear regression method decreases because it cannot find a single line that fits both the pre-and post-saturation point phases. On the other hand, GD can find two regressors for pre- and post-saturation phases in the exponential growth model. For the pre-saturation part, GD performs well when the length of exponential growth phase is larger than 20 time points with learning rate as 0.0005. In order to achieve good performance of GD, the minimum data points for pre-saturation point should be greater than 30 ($T_e > 30$).

However, for the post-saturation part, although GD can find a line to fit the data, the line is still not perfect, indicated by the larger average minimum square loss. Based on our results, the increasing learning rate generally decreases the performance of GD. Also, from those experimental results based on our model, GD converges slowly, e.g. GD normally needs 21 time points on average to find the best weight vector with the minimum square error loss. It is well-known that another algorithm called exponentiated gradient (EG) usually converges much faster than GD [34]. They both maintain a weight vector using simple updates. For the GD algorithm, the update is based on subtracting the gradient of the squared error made on a prediction. The EG algorithm uses the components of the gradient in the exponents of factors that are used in updating the weight vector multiplicatively. Research has shown that if few components of the input are relevant for each prediction, EG has a much smaller loss [34]. We can apply EG on the artificial data we generated or on the real data to evaluate its performance.

Based on our experiments, we found if we can define the saturation point, we can apply the standard regression method to each phase, which actually gives the optimum solution. To do this, we can derive an equation as follows:

$$f_e(t)g(t) - f_i(t)g(t - t_s) + f_s(t)g(t - t_s),$$

where $f_e(t)$ is the function for the exponential growth phase, $f_s(t)$ is the function for the stagnant growth phase, i.e. after the saturation point. $g(t)$ is the unit step function,

$$g(t) = \begin{cases} = 0, & t < 0, \\ = 1 & t \geq 0. \end{cases}$$

The unit step function can be approximated by a mathematical function such as a sigmoid function:

$$\sigma(t) = \frac{1}{(1 + e^{-t})}$$

By this approach, we may be able to model the relationship between abundance and time as a single equation, which certainly has advantages compared to other approaches that model each phase individually.

With increasing computational challenges posed by the recent information explosion in biological research, there will be more and more interest to apply different computer science algorithm, such as machine learning and pattern recognition algorithms, to those areas. This thesis made an attempt to that direction.

References

1. Baldi, P. and Soren Brunak (1998). *Bioinformatics: A Machine Learning Approach*. MIT press, Boston.
2. Lee, K. (1989). Automatic speech recognition: The development of the Sphinx system. Kluwer Academic Publishers, Bonston.
3. Pomerleau, D. A. (1989) ALVINN: An autonomous land vehicle in a neural network. (Technical Report CMU-CS-89-107). Pittsburgh, PA: Carnegie Mellon University.
4. Fayyad, U. M., Smyth,P., Weir, N., Djorgovski, S. (1995). Automated analysis and exploration of image databases: Results, progress, and challenges. *Journal of Intelligence Information Systems*, 4:1-19.
5. Kim, J., J. Nietfeldt, and A.K. Benson. (1999). Octamer-based genome scanning distinguishes a unique subpopulation of E. coli O157:H7 strains in cattle. *Proc. Natl. Acad. Sci. USA.*, 96.
6. Altman, R. B. (1998). Bioinformatics in Support of Molecular Medicine. In C.G. Chute, Ed., Proceedings of the 1998 AMIA Annual Symposium, Orlando, FL., 53-61.
7. Cooper, G. (1997). An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine*,8.
8. Tesauro, G. (1992). Practical issues in temporal difference learning. *Machine learning*, 8:257-260.
9. Tesauro, G.(1995). Temporal difference learning and TD-gammon. *Communication of the ACM*, 38(3): 58-68.

10. Theodoridis, S. and Konstantinos Koutroumbas. (1999). *Pattern Recognition* Academic Press.
11. Stringer, P. (1967). Cluster analysis of non-verbal judgement of facial expression. *Br. J. Math. Statist. Psychol.*, 20:71-79.
12. Soloman, H. (1971). Numerical taxonomy. *Mathematics in the Archaeological and Historical Sciences* (Hobson F.R., Kendall D.G., Tautu P.A., eds.), University Press.
13. Eisen, M. and P. Brown. (1999). DNA arrays for analysis of gene expression. *In Methods in Enzymology*. 303:179-205.
14. Califano, A., G. Stolovitzky, and Y. h. Tu (2000). Analysis of gene expression microarrays for phenotype classification. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*.
15. Van Someren, E.P, L.F.A. Wessels, and M.J.T.Reinders (2000). Linear modeling of genetic networks from experimental data. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*.
16. Eisen, M., Spellman, P., Brown, P. and Bottstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Science of the USA* 95(25):14863-14868.
17. Sharan, R. and R. Shamir (2000). CLICK: A clustering algorithm with applications to Gene Expression Analysis. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*.
18. Solomon, M. E. (1976). *Population Dynamics*. The Gamelot Press Ltd. Great Britain.
19. Tsuchiya, H. M., A. G. Fredrickson , and R. Aris. (1996). Dynamics of microbial cell populations, *Adv. Chem. Eng.* 6:125.

20. Fredrickson, A. G., R. D. McGee, and H. M. Tsuchiya. (1970). Mathematical models for fermentation process. *Adv. Appl. Microbiol.* 13: 419-421.
21. Fredrickson, A. G., D. Ramkrishna, and H. M. Tsuchiya. (1971). The necessity of including structure in mathematical models of unbalanced microbial growth. *Chem. Eng. Prog. Symp. Ser.*, (1971) 67:53.
22. Lewis, R. E. (1977). *Network Model In Population Biology*. Springer-Verlag Berlin.
23. Bazin, J. M. (1982). *Microbial Population Dynamics*. CRC Press.
24. Ratkowsky, D.A., J.Olley, T.A. McMeekin and A. Ball (1982). Relationship between temperature and growth rate of bacterial cultures. *Journal of Bacteriology* 149:1-5.
25. Van Impe, J.F., B. M. Nicolai, Mia Schellekers, T. Martens, J. D. Baerdemacker (1995). Predictive microbiology in a dynamic environment: a system theory approach *Internatinal Journal of Food Microbiology* 25:227-249.
26. Schmidt, T. M. and D.A. Relman. (1997). Phylogenetic identification of uncultured pathogens using ribosomal RNA sequences. In V. Clark and P. M. Bavoil (eds.) *Bacterial Pathogenesis*, 93-109.
27. Lenski,R. E., J. A. Mongold, P. D. Sniegowski, M. Travisano, F. Vasi, P. J. Gerrish and T. M. Schmidt (1998). Evolution of competitive fitness in experimental populations of *E. coli*: What makes one genotype a better competitor than another. *Antonie van Leewenbock*.
28. Guttman, D. S. and Dykhuizen, D. E. (1994). Detecting selective sweeps in naturally occurring Escherichia Coli. *Genetics* 138:993-1003.
29. Norvak, B. and J. J.Tyson (1995). Quantitative Analysis of a Molecular Model of Mitotic Control in Fission Yeast. *J.Theor. Biol.*173:283-305.

30. Herbster, M. and M. K. Warmuth. (1998). Tracking the best regressor. *Proceedings of the 12th Annual Conference on Computing Learning Theory*. 24-31.
31. Kivinen, J. and M. K. Warmuth. (1997). Relative loss bounds for multidimensional regression problems. *In Advances in Neural Information Processing Systems*. Cambridge, M.A., MIT Press.
32. Brown, P. and P. Rothery (1993). *Models in Biology: Mathematics, Statistics and Computing*. John Wiley & Sons Ltd. England.
33. William, H. P., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery (1992) *Numerical Receipts in C: The Art of Scientific Computing*. Cambridge University Press.
34. Kivinen, J. and M. K. Warmuth. (1995). Exponentiated Gradient Versus Gradient Descent for Linear Predictors. *Proceedings of the 27 Annual ACM Symposium on the Theory of Computing*. New York, ACM Press.

Appendix B: Example of Clustering Procedure

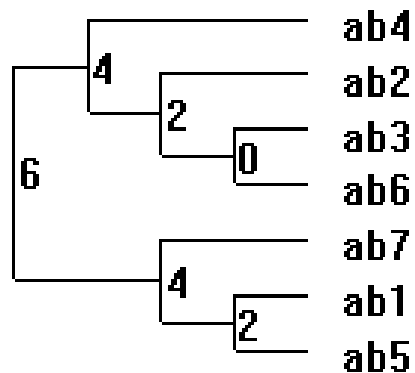
B.1 Example of a small data file containing 7 bit strings converted from corresponding DNA electrophoresis gel pattern

ab1	0 0	0	0	0 0	0	0	0	1
ab2	0 0	0	1	0 1	1	1	1	1
ab3	0 0	0	1	1 1	1	0	1	1
ab4	1 1	1	1	1 1	1	1	1	1
ab5	1 0	0	0	0 0	0	0	0	0
ab6	0 0	0	1	1 1	1	0	1	1
ab7	0 0	1	1	0 1	0	0	0	1

B.2 The dissimilarity matrix at the initialization step build from the above data file through MUAS clustering algorithm

	ab1	ab2	ab3	ab4	ab5	ab6	ab7
ab1	0						
ab2	5	0					
ab3	5	2	0				
ab4	9	4	4	0			
ab5	2	7	7	9	0		
ab6	5	2	0	4	7	0	
ab7	3	4	4	6	5	4	0

B.3 The Dendrogram produced by MUAS clustering algorithm for the above data file



Appendix C: Experimental Procedure for Artificial Data

