7-28-2015

Individual-Based Modeling and Data Analysis of Ecological Systems Using Machine Learning Techniques

Morteza Mashayekhi
University of Windsor

Individual-Based Modeling and Data Analysis of Ecological Systems Using Machine Learning Techniques

By

Morteza Mashayekhi

A Dissertation
submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
at the University of Windsor

Windsor, Ontario, Canada
2015

Individual-Based Modeling and Data Analysis of Ecological Systems using Machine Learning Techniques

by

Morteza Mashayekhi

APPROVED BY:

_____

Dr. T. White, External Examiner
Carlton University

_____

Dr. T. Pitcher
Great Lakes Institute for Environmental Research

_____

Dr. L. Rueda
School of Computer Science

_____

Dr. A. Ngom
School of Computer Science

_____

Dr. R. Gras, Advisor
School of Computer Science

April 29, 2015

Declaration of Co-Authorship / Previous Publication

I. Co-Authorship Declaration

I hereby declare that this dissertation incorporates material that is result of joint research with Dr. Robin Gras, my supervisor. This dissertation also incorporates the outcome of a joint research undertaken in collaboration with Brian McPherson, Abbas Golestani, Meisam Hosseini, Yasaman Majdabadi Farahani, Armin Sajadi, Ryan Scott, Elham Salehi, and MD Sina under the supervision of Dr. Robin Gras. The collaboration is covered in Chapters 2, 5, and 6 of the dissertation. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author, and the contribution of co-author was primarily through the provision of required background biological information. In Chapter 2, Ms. Khater also contributed in explaining some parts of the materials.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my dissertation, and have obtained written permission from each of the co-author(s) to include the above material(s) in my dissertation.

I certify that, with the above qualification, this dissertation, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This dissertation includes 8 original papers that have been previously published/submitted for publication in peer reviewed journals and conferences, as follows:

Dissertation Chapter	Publication title/full citation	Publication status*
Chapter 2	R. Gras, A. Golestani, M. Hosseini, M. Khater, Y.M. Farahani, M. Mashayekhi, M. Sina, A. Sajadi, E. Salehi and R. Scott, EcoSim: an individual-based platform for studying evolution, European Conference on Artificial Life, pp 284-286, 2011.	Published
Chapter 2	M. Mashayekhi, A. Golestani, Y.M. Farahani, R. Gras, An enhanced artificial ecosystem: Investigating emergence of ecological niches, International Conference on the Simulation and Synthesis of Living Systems (ALIFE 14), pp 693-700, 2014.	Published
Chapter 4	M. Mashayekhi, R. Gras, Speciation prediction based on spatial distribution and spatiotemporal information from an Individual-Based Ecosystem Simulation, 2nd Advanced topics in artificial intelligence ATAI conference, pp.56-62, 2011.	Published
Chapter 4	M. Mashayekhi, R. Gras, Investigating the effect of spatial distribution and spatiotemporal information on speciation using individual-based ecosystem simulation, Journal of Computing 2:1, 2012.	Published
Chapter 5	M. Mashayekhi, M. Hosseini, R. Gras, Can we predict speciation and species extinction using an individual-based ecosystem simulation? 15th International conference on artificial intelligence ICAI, pp.301-307, 2013.	Published
Chapter 5	M. Mashayekhi, B. MacPherson, R. Gras, A machine learning approach to investigate the reasons behind species extinction, Ecological Informatics, 20, 58-66, 2014.	Published
Chapter 6	M. Mashayekhi, B. MacPherson, R. Gras, Species–area relationship and a tentative interpretation of the function coefficients in an ecosystem simulation, Ecological Complexity, 19, 84-95, 2014.	Published
Chapter 7	M. Mashayekhi and R. Gras, Rule extraction from random forest: RF+HC methods, in Advances in Artificial Intelligence, 2015, pp. 223–237.	Published

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my dissertation. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my dissertation does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my dissertation, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my dissertation.

I declare that this is a true copy of my dissertation, including any final revisions, as approved by my dissertation committee and the Graduate Studies office, and that this dissertation has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Artificial life (Alife) studies the logic of living systems in an artificial environment in order to gain a deeper insight of the complex processes and governing rules in such systems. EcoSim, an Alife simulation for ecological modeling, is an individual-based predator-prey ecosystem simulation and a generic platform designed to investigate several broad ecological questions, as well as long-term evolutionary patterns and processes in biology and ecology.

Speciation and extinction of species are two essential phenomena in evolutionary biology. Many factors are involved in the emergence and disappearance of species. Due to the complexity of the interactions between different factors, such as interaction of individuals with their environment, and the long time required for the observation, studying such phenomena is not easy in the real world. Using data sets obtained from EcoSim and machine learning techniques, we predicted speciation and extinction of species based on numerous factors. Experimental results showed that factors, such as demographics, genetics, and environment are important in the occurrence of these two events in EcoSim.

We identified the best species-area relationship (SAR) models, using EcoSim, along with investigating how sampling approaches and sampling scales affect SARs. Further, we proposed a machine learning approach, based on extraction of rules that provide an interpretation of SAR coefficients, to find plausible relationships between the models' coefficients and the spatial information that likely affect SARs. We found the power function family to be a reasonable choice for SAR. Furthermore, the simple power function was the best ranked model in nested sampling amongst models with two coefficients. For some of the SAR model coefficients, we obtained clear correlations with spatial information, thereby providing an interpretation of these coefficients.

Rule extraction is a method to discover the rules explaining a predictive model of a specific phenomenon. A procedure for rule extraction from Random Forest (RF) is proposed. The proposed methods are evaluated on eighteen UCI machine learning repository and four microarray data sets. Our experimental results show that the proposed methods outperform one of the state-of-the art methods in terms of scalability and comprehensibility while preserving the same level of accuracy.

DEDICATION

To my parents for their encouragement

To my wife and daughter for their support

To my brothers and sisters for their kindness

ACKNOWLEDGEMENTS

I would like to gratefully thank Dr. Robin Gras, my supervisor, for his support and continued guidance to apply and extend my knowledge in a cross disciplinary field, working at the forefront of computer science. I would like to thank my committee members Dr. White, Dr. Pitcher, Dr. Rueda and Dr. Ngom for accepting to allocate part of their valuable time to evaluate my research.

Part of this work was made possible by the facilities of Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca) and Compute/Calcul Canada.

Contents

List of Tables

List of Figures

Chapter 1

1. Introduction

1.1. Motivation

Artificial life (Alife) investigates systems related to life including the processes and the evolution in software, hardware, and biochemical [1]. Alife studies the logic of living systems in an artificial environment in order to gain a deeper insight of the complex processes and governed rules in such systems. The general modeling approach in Alife is to model and simulate the generic principles underlying life [2].

Among biological disciplines, behavioral ecology has a strong tradition of accounting for the role of organism–environment interactions in behavior [3]. Behavioral ecology and the related field of optimal foraging theory [4] model animal behavior in terms of optimal adaptation to environmental niches. The goal is to interpret the behavior of organisms and also to generate testable hypotheses, rather than test whether organisms actually behave optimally [5]. One approach for understanding the behavior of complex ecosystems is individual-based modeling, which provides a bottom-up approach allowing for the consideration of the traits and behavior of individual organisms [6]. It models every single individual agent and their interactions with the other agents and also their reactions to the environmental conditions such as food resources or predator stress in an artificial ecosystem. The main outcome of the artificial ecosystem is the emergence of some high level phenomena, which are the results of the whole set of interactions. By simulating the general interaction rules of real ecosystems, patterns similar to what are observed in nature could emerge in the artificial ecosystem. These patterns include population migration, shape of spatial distribution of individuals, extinction, and speciation.

Ecological modeling is still a growing field, at the crossroad between theoretical ecology, mathematics and computer science [7]. Since natural ecosystems are very complex (in terms of number of species and also ecological interactions), ecosystem models aim to characterize the major dynamics of ecosystems, in order to synthesize the understanding of such systems, and to allow predictions of their behavior [6]. Ecosystem simulations can also help scientists to understand theoretical questions regarding the evolutionary process, the speciation, and the extinction of species. One of the main interests of such ecosystem simulations is that they offer a global view of the evolution of the system, which is difficult to observe in nature [8].

Such artificial ecosystem simulation can provide vast amount of data related to every single individual, something that is not available in nature or it is hard to measure. Having those data is very beneficial because data can be turned into information and then information gives rise to insight. Therefore, data analysis plays an important role after running the simulation in order to turn the generated raw data into insight. Data analysis is an iterative process consisting of collecting, processing, cleaning, transforming, and finally modeling the data to be able to discover useful knowledge and drawing conclusions [9], [10]. Machine learning is one of the most popular methods in data analysis. They are able to extract useful knowledge, suggesting conclusions, and helping decision-making by learning from the raw input data [11]. Regression, classification, feature selection, rule extraction are examples of machine learning methods that can be used for this purpose.

1.2. Objective

The main objective of this dissertation is to investigate to what extent the combination of Alife and artificial intelligence (AI) contributes in ecology. To answer this question, we use EcoSim as the main platform. EcoSim [12], [13], [6] is an Alife simulation for ecological modeling and an artificial individual-based predator-prey ecosystem simulation. It is a generic platform designed to investigate several broad ecological questions, as well as long-term evolutionary patterns and processes in biology and ecology. We investigated the creation and disappearance of species and their spatial distribution in EcoSim to evaluate and verify the conformity of the output patterns of EcoSim to the real ecosystems. Not only did these studies aid the model validation, but they also provided deeper understanding of those phenomena in the natural ecosystems. For data analysis and extracting meaningful explanation, machine learning techniques were applied to the vast amount of data generated by EcoSim. Our main challenge concerns the reasons behind speciation and extinction of species and their spatial distribution pattern using species-area relationship, three fundamental subjects in ecology. The geographical and spatial distribution of individuals in one species is a leading phenomenon for speciation. Therefore, we investigated the speciation mechanism by studying how the spatial and spatiotemporal distributions of individuals influence speciation. Applying machine learning techniques, we wanted to investigate how this information influences speciation in EcoSim. In another words, our aim was to test if there is some correlation between such features and speciation. Positive answer to this question would also be a confirmation of the validity of the model of evolution and species emergence in EcoSim.

Speciation and extinction of species can be affected by several factors. Based on Darwin's theory, natural selection is the main reason for speciation and emerging genetics studies strengthened this

theory by explaining variation in a population via genetic operations [14]. Pre- and post-zygotic barriers, which lead to reproductive isolation, are also very important in speciation. Geographically isolated populations tend to form new species as well [15], [16]. Moreover, sexual selection plays an important role in speciation [17]. Likewise, there are many factors involved in extinction that can be classified into three main areas of demographics, genetics and environmental factors [18], [19]. Demographic factors can affect the birth rate and the death rate of the population. Additionally, the effect of demographic stochasticity is greater in small than in larger populations [20]. There is also possibility of genes being lost when a huge reduction occurs in a population and the gene frequencies may be changed due to drift or inbreeding [21]. Diminishing genetic variation may increase extinction risk by limiting the adaptation ability to stressful environments [18]. Lastly, environmental factors such as natural catastrophes, availability of food, competitors, predators, and diseases influence the population by changing the demographic parameters and increase the likelihood of extinction. Predicting these two phenomena and discovering important factors involved, would bring new insights in evolutionary and conservation biology. Observing and studying species in nature to extract species information is a difficult and time consuming process. In addition, speciation and extinction processes occur at very long time scales and most of the time is not possible to observe them in nature. However, using simulated ecosystem facilitates such studies. Our aim was to predict these two events based on several demographical, genetic, environmental, and spatial features, which are likely effective on these two events. Afterward, based on prediction results, we investigated the important factors (or features) involved in these events and analyzed their accordance with biological evidence. Being able to demonstrate that the emergence of species and their extinction in EcoSim is similar to what happens in nature would allow ecologists to propose more specific studies that can be performed using EcoSim.

Identifying the best species-area relationship (SAR) model using EcoSim, along with investigating how sampling approaches and sampling scales affect SARs was the third objective. Further, we attempted to determine a plausible interpretation of SAR model coefficients for the best performing SAR models. The species-area relationship (SAR) is one of the most well-known and oldest patterns in ecological modeling that has a number of practical applications for managing natural communities [22], [23], [24]. Identifying the most biologically appropriate mathematical SAR model to characterize these behaviors has been one of the most important and controversial issues in biodiversity. Our aim was to answer questions such as: Is the power function the best suited SAR model overall? How do nested sampling and random sampling

affect the shape of the SAR curves? Do different sampling scales affect the SAR models? Is there any correlation between SAR model coefficients and spatial information? We employed EcoSim and machine learning techniques to answer such questions.

One of the most common approaches for data analysis and inferences in the above mentioned objectives was rule extraction. Rule extraction is a technique to simply explain the underlying predictive model, which is in general mostly a black-box model. Therefore, we were looking for a rule extraction method that not only gives simple and comprehensible rule set, but it also provides high accuracy. Random forest (RF) [25] is an ensemble learning method and one of the high performance predictive models. Despite its good performance, one possible limitation of RF is that it generates a forest consisting of many decision trees. Therefore, it is viewed as a black box model because of its multitude of rules. Our aim was to build a rule extraction method based on RF that both maintain the accuracy level close to RF's accuracy and drastically reduce the numbers of rules compared to RF.

1.3. Contributions of the thesis

1. First speciation was studied in EcoSim [26], [16], [27]. We analyzed the ability of spatial and spatiotemporal information about species in our artificial ecosystem for the prediction of speciation events. We showed that some generic traits exist in EcoSim that characterize the speciation events. In addition, the effectiveness of demographics, genetics, environment, and spatial distribution features in speciation prediction was demonstrated. We extracted several simple rules from the constructed prediction model. These rules were semantically clear and sound reasonable based on biological evidence. This is an important result as the proposed approach has proven to have the capability of generating realistic rules when compared with real biological data.

2. The second contribution was to study the reasons behind species extinction in EcoSim [27], [28]. We used three broad categories of genetic, environmental, and demographic features for this purpose. Afterward, we obtained a rule set for each category and showed that these rules can predict extinction in the next 100 time steps with a very high level of accuracy. We also demonstrated that these rules are generic by applying a model built on a training set to a validation set constructed using completely different simulation runs. The proposed approach was able to extract important features in extinction effectively, especially when there is a plethora of features and there is no exact knowledge about them. Second, the categorization idea helps to study the effect of features in a more fine-grained way and to extract the associated rules

accompanied by an evaluation of their accuracy. This may prove to be beneficial for conservation biologists from the point of view of being able to detect early signals of extinction. Further, this approach can be applied to test new hypotheses regarding new factors involved in extinction. While our results are not directly valid for real situations given that our model involves a high level of abstraction as well as being a simplification of the real world, our results provide interesting insights that could be of use to biologists in formulating new hypotheses relating to species extinction.

3. We employed EcoSim, to investigate the SAR in terms of the best SAR model, effect of sampling strategy on SAR model, and finally study the correlation between SAR model coefficients and spatial information [29]. Our study demonstrated that although there is no unique function that best describes all species-area relationships, functions in the power family were the best ranked functions. Amongst them, the power function is the simplest model with the fewest coefficients and hence it is normally easier to fit the simple power function to the data. However, for more accurate results, a more complicated model may better fit the data. Furthermore, we demonstrated that a number of factors, such as sampling scale and sampling strategies, should be taken into account because they affect the shape of the SAR models. We found different models to be the most suitable function for different sampling methods and sampling scales. We proposed, for the first time, a machine learning approach to discern the meaning of the SAR functions' coefficients by providing several rules associated with their probability of prediction. We were able to determine the meanings of the SAR coefficients from these extracted rules.

4. Random forest (RF) is a tree-based learning method, which exhibits a useful ability to generalize on real data sets. Nevertheless, a possible limitation of RF is that it generates a forest consisting of many trees and is viewed as a black box model because of its multitude of rules. We proposed, a procedure for rule extraction from a RF: the RF+HC methods [30]. Once the RF is built, a hill climbing algorithm is used to search for a rule set such that it reduces the number of rules dramatically, which significantly improves comprehensibility of the underlying model built by RF. The proposed methods are evaluated on eighteen UCI machine learning repository [31] and four microarray data sets. Our experimental results show that the proposed methods outperform one of the state-of-the-art methods, CRF method, in terms of scalability and comprehensibility while preserving the same level of accuracy.

1.4. Outline of thesis

Chapter 2 reviews existing literature on evolutionary systems and the application of individual-based modeling in ecology. It also gives an overview of the platform simulation used in this study, EcoSim, a predator-prey ecosystem simulation, which is a useful tool to study general and fundamental ecological and biological theories.

Chapter 3 reviews rule extraction methods and mostly concentrated on the rule extraction methods from ensembles of decision trees.

Chapter 4 discusses the effectiveness of various features on speciation events. It explains two distinct experiments to study speciation.

Chapter 5 investigates the reasons behind species extinction using machine learning techniques. A rule set for each category of features is extracted, that show the conformation of species extinction with the extinction in real nature.

Chapter 6 studies the species-area relationship (SAR), one of the most well-known and oldest patterns in ecological modeling. In addition to investigating the best mathematical model for SAR, the effect of different sampling strategies on the shape of SAR is discussed and finally the relationship between SAR model coefficients and ecological factors are investigated.

Chapter 7 proposes the rule extraction methods derived from random forest. The experimental results discuss how the proposed methods improve the scalability and comprehensibility of one of the state-of-the-art methods.

Chapter 2

2. Background and Literature Review

2.1. Artificial life

Artificial life is a field of study devoted to understanding life by creating artificial systems to acquire general theories underlying biological phenomena, and recreating these dynamics in other forms such as computer simulation [32]. There are three broad methods to implement such a system: software implementation of digital organisms ('soft'), hardware implementation of life-like systems ('hard'), and using biochemical substances to synthesize living systems ('wet') [1]. The first Alife system was designed using self-reproducing, computation-universal cellular automata by Von Neumann [33] and at the same time using information theory and the analysis of self-regulatory processes by Wiener [34] to study fundamental of characteristics of the living systems. The goal of Alife is to provide a different perspective for biology researchers. Alife offers a synthetic perspective by constructing phenomena from their primitive units while biological research is mainly analytic, attempting to break down complex phenomena into their basic components. Alife implements simple rules and concepts, and combines them leading to the emergence of complex phenomena. Emergence is one of the main characteristics of Alife systems where phenomena at a certain level arise from interactions at lower levels [35], [36]. Alife also has overlap with computer science topics, especially artificial intelligence [37], as in both some form of intelligence is required for living in a changing environment. Moreover, both fields study natural phenomena [1]. However, there is a major difference in their modeling strategies. Most traditional AI models construct top down serial systems with a centralized and complicated decision controller that decides based on the knowledge about all aspects of global state. On the other hand, Alife is mainly concerned with gaining knowledge about living systems using computational bottom-up complex systems consisting of low-level and simple agents interacting with each other. Agents decide based on their local environment in parallel and their decisions' impact is only on their own local environment. In this way, the global behavior of the whole system is shaped [1].

Complex systems consist of many elements interacting with each other simultaneously. The complex systems that learn or adapt to a changing environment are complex adaptive systems [38], and are the main focus of Alife [1]. Complex adaptive systems exhibit emergence where the behavior of the whole is more complex than the behavior of the parts. The characteristics of emergence are: (a) Emergence happens in systems which compose of different interactive units

that obey simple rules. (b) The interactions between the parts are nonlinear such that the overall behavior of the system cannot be predicted by summing the behaviors of the isolated parts. (c) The system functions change with the modification of context making difficult to predict emergent behavior. (d) The system complexity increases with increasing number of interactions [39]. Evolutionary emergence is an essential feature in Alife [40]. There are no rules in the system that dictates global behavior and any behavior at levels higher than the individuals is emergent. In the Alife systems with evolution mechanisms, there are two types of selection that might bring such emergence: "extrinsic adaptation where evolution is governed by a specified fitness function, and intrinsic adaptation, where evolution occurs automatically as a result of dynamics of a system cause by the evolution of many interacting subsystems" [41].

2.2. Alife for Ecological Modeling

Alife uses individual-based modeling (IBM) which is a bottom-up approach to simulating the interactions among individuals or groups of individuals in an attempt to create complex phenomena. On the other hand, classical equation-based models (EBMs) are typically built up from set of interrelated differential equations. Unlike EBMs, IBM consists of interacting adaptive entities which are able to capture emergent behavior and provide a greater level of useful details. The ease of modeling renders IBM more flexible than EBM. IBM has been used on non-computing related scientific domains such as ecological sciences [42] and social sciences [43].

The benefits of IBM over other modeling techniques can be seen in several points. Agent-based models are a natural way to describe systems comprised of interacting entities. They are flexible and capture emergent phenomena. Finally, they provide access to a greater level of useful detail [44]. For instance, modeling interactions between entities is much easier in agent-based systems than in EBMs, even when one is comfortable with the concepts of partial differential equations. It is usually easy to increase the capacity of a simulation, by adding new agents to see if interesting effects are swamped by agent numbers, or by taking agents away if interesting detail is obscured. It is also possible to look at the results of simulations at different granularity levels such as the level of a single agent, the level of some specific group of agents, or the level of all agents together. All these things are harder to manage in EBMs. In addition to their inherent naturalness and flexibility, agent-based simulations allow one to identify emergent phenomena, which are the result of the actions and interactions of individual agents together and with the environmental factors. However, IBM has its own disadvantages. For instance, some experiments need very large population sizes and simulations over long periods of time. For these situations, IBM costs increase in terms of time and hardware requirements. Moreover, the number of parameters of

such modeling approaches is, in general, very large. Therefore, finding the best initial parameters is not straightforward as thorough exploration of the parameter space is not possible. Therefore, analyzing the effect of every parameter on the simulation and how the results are biased by a specific set of parameter is particularly difficult. In addition, the emergent properties sometime are artifacts from the model or implementation instead of being real features of the simulation [45].

For the past decade there has been an enormous growth in application of IBM addressing different questions in ecology and evolutionary biology. Whereas classical approaches to modeling ecology often ignore individual behavior and instead uses state-variable model that controls birth and death rates, IBM aims to "treat individuals as unique and discrete entities" [46] which provides for a more realistic simulation. IBM has been used in many areas in ecology including forest ecology [47], fisheries and marine life [48], conservation biology and spatial heterogeneity [49]. This approach has been used in the simulation of ecological and evolutionary processes such as ecological speciation [50], conservation applications [51], and gender change [52]. Many ecological IBM systems were not designed to be general platforms that could capture different aspects in ecology and evolution but rather these models answer specific question in a narrow domain. ATLSS (across trophic level system simulation), designed to simulate the ecological functioning of the Everglades region in Florida and model abiotic factors and various trophic levels [53], or individual-tree model of the forests of the northeastern United States [54] are examples of such systems. Other evolutionary IBMs were designed as platforms to study evolutionary behavior, emergence, adaptation, and complexity that are discussed below.

2.2.1. Tierra

Coreworld [55] and its improved version, Tierra [56] were the first experiments with populations of self-replicating computer programs performed in 1990. The Tierra model is the first widely known digital evolutionary ecosystem consisting of self-replicating computer programs based on natural selection. Competition in Tierra results from finite CPU-time and memory space. Tierra is based on a virtual operating system, complete with its own, relatively robust and simple (but universal) machine language and a fixed size address space. An evolutionary run starts by seeding the empty memory space with a hand-written self-replicator program. This replicator then produces a copy of itself which is instantiated as an independent process. A small amount of stochastic behavior is implemented for program execution, the copy process, and programs are also subject to point mutations. These mechanisms are responsible for introducing variety into the populations. If the modified programs retain their ability to replicate, and the modifications alter

their probability of reproduction, Darwinian evolution can occur. A number of interesting results have been obtained from such evolutionary runs. For example, 'parasites' have appeared-short pieces of code which run another program's copying procedure in order to copy themselves. Hyper-parasites (parasites of parasites) have also been observed, along with a number of other interesting ecological phenomena [56]. It was shown that it is possible to build an operating system in which self-replicating computer code can evolve. On the other hand, after a certain amount of time, Tierra fails to produce any new programs but only change in the number of existing ones [57].

A few numbers of other systems were built based on Tierra. Cosmos, a Tierra-like system configured in a two dimensional toroidal like grid environment, was used to study the role of contingency in evolution [58]. Furthermore, in Amoeba [59], the language of the digital organism along with its self-replicating code is also subject to evolution. The Amoeba system, showed the possibility of spontaneous emergence of a self-replicating program.

2.2.2. Avida

Avida is a Tierra-like system [60], [61], in which self-replicating digital organisms consist of a circular list of instructions (its genome) and a virtual CPU evolve. Each organism lives in its own address space, unlike Tierra's shared address space. This enhancement increased the power of digital evolution as an experimental tool. Avida's environment comprises a number of cells; each cell can contain at most one organism, and the size of an Avida population is bounded by the number of cells in the environment. Organisms are self-replicating, that is, the genome itself must contain the instruction to create an offspring. When an organism replicates, a cell to contain the offspring is selected from the environment and its inhabitant organism is replaced (killed and overwritten). Since digital organisms are self-replicating and compete for space, a higher merit (all else being equal) results in an organism that replicates more frequently, spreading throughout and eventually dominating the population. Hence, Avida satisfies the three conditions necessary for evolution to occur: replication, variation (mutation), and differential fitness (competition). Individuals in Avida do not move and in order to measure complexity they use a fixed environment which is rarely seen in nature. This means that the system is only adapting to fix pre-existing environmental conditions. The processes derived from Avida and Tierra are optimization processes, similar to evolutionary algorithms, for which it has been proved that it converge toward a maximum, either local or global. Finally, as with Tierra, the complexity growth in Avida always reaches an upper bound and stops. These results with Avida do not capture the kind of

continual growth in qualitative complexity or long term incremental evolution that we can observe in the biosphere.

Avida was used to study numerous aspects of evolution [62]; issues of complexity in evolution [63], [64]. Furthermore, they investigated the emergence of complex behavior [65]. They showed that complex features do not appear suddenly but only evolve when simpler traits exist which served as a foundation upon which these complex features were built. In a recent study they showed how runaway sexual selection leads to good genes and how they should be viewed as interacting mechanisms that reinforce one another [66]. Evolving digital ecological networks was presented in [62] that models competition, parasitism and mutualism.

2.2.3. Echo

Echo [67] is a generic ecosystem in which agents evolve in a resource limited environment. The world is made up of a square toroid lattice of sites which has different kinds of evolving resources encoded by a letter. Agents interact with their environment and are able to move from one site to another. They gain energy by eating and spend it on their actions such as fighting, trading and mating. Reproduction in Echo happens when an agent has replicated itself with a possible mutation when it has gained enough resources to copy its genome asexually or by sexual mating. Selection is based on the interacting agents rather than by a predefined fitness function. Emergent phenomena arise such as formation of communities and trading networks. Echo was used to study the modeling of food web complexity [68]. Echo was intended to be a general model of intrinsic adaptive system rather than modeling and answering specific questions in evolutionary biology. Due to the high abstraction level of the Echo model, the degree of fidelity to real systems is uncertain.

2.2.4. Polyworld

In PolyWorld [69], more advanced haploid agents, each controlled by an artificial neural network, with a set of primitive behaviors and learning strategies, populate a continuous environment containing a number of energy sources ('food') upon which they rely on for survival. Possible actions for agents include eat, mate, fight, move, focus and light (for vision). Agents evolve under the influence of natural selection and die when their energy is fully depleted or lose a fight with another agent. An agent's genome specifies characteristics of its physiology and neural architecture which is adapted during its life through Hebbian learning [70]. In Hebbian learning, the weight between two neurons is increased if the two neurons activate at the same time, otherwise it is reduced. Therefore, the weights between two nodes which are both either positive or negative simultaneously have high positive weights, while the weights between two opposite

nodes have high negative weights. Yaeger was able to report the emergence of new population behaviors, such as fleeing, grazing, following, and flocking. Polyworld was used to study how evolution guides complexity [71] and the passive and driven trends in the evolution of complexity [72]. Genetic clustering for the identification of species was also presented in [73]. On the other hand, lack of semantics in the genomic structure (nodes) in Polyworld, makes it difficult to reason and link together different aspects of the model. Another criticism of PolyWorld, in the context of perpetual evolutionary emergence, is that learning appears to be overwhelmingly responsible for the results. This integrated learning process adds to the computational complexity of the model. Furthermore, the high complexity of the neural networks agents limits their number making it difficult to study large ecosystem phenomena's.

Geb [74], [75] is another similar artificial neural network system considered to be simpler than Polyworld as it is not trying to mimic the real world as Polyworld do. Agents which are controlled by a neural network each populate a gridded arena and compete for space with no notion of energy. There is no learning process as agents do not change during their lifetime and thus results prove it to be suited to long-term incremental artificial evolution. Geb was proven to be the first autonomous artificial system to pass the Bedau and Packard's evolutionary test [41]. According to Bedau statistics, evolutionary dynamics in Gep was proven to be unbounded [76] and thus based on intrinsic evolution. Bedau et al. [41] developed a statistical measure for testing unbounded evolution.

2.2.5. Framsticks

Framsticks [77] is a 3D life simulation platform addressing both research and education. The platform consists of modules that facilitate the design of various experiments in optimization, coevolution, open-ended evolution, and ecosystem modeling. Agents have both mechanical structure (bodies) consisting of connected sticks and a control system (brain) using an artificial neural network. The neural network brain collects data from sensors and sends signals to the joints which control motion. The world is enriched with a complex topology and a water level along with energy balls consumed by agents. Although some locomotion behaviors have evolved, the high complexity of the model did not present any different results than those obtained from much simpler evolutionary systems. This model is more concerned with the study of emerging motor behavior rather than modeling a multiple level interacting ecosystem.

2.2.6. Sugarscape

Sugarscape [78] is an agent-based social simulation consisting of agents, a two-dimensional environment, and the rules that define the interaction of the agent with each other and the environment. The environment or cellspace is a 51x51 cell grid, where every cell contains either sugar or spice. It has some general properties that control the number of inhabitants and the overall fertility of sugar and spice distributions on the grid. An agent occupies all of one cell and there is no sharing of cells. Each cell has the following attributes: sugar, spice, and pollution. Sugar and spice are consumed by the agents visiting the cell and after harvested grow back based on a simulation parameter. There are also some randomly selected cells that cannot grow one or both of the sugar and spice.

In every step of the simulation, agents look around and find the closest cell containing sugar or spice, then move to that cell and metabolize. Depending on the parameter setting defined at the set-up of the model, agents can leave pollution, die, reproduce, inherit sources, transfer information, trade or borrow sugar, generate immunity or transmit diseases. It can be used to study the effects of social dynamics such as evolution, marital status, and inheritance on populations. Each agent has the following characteristics: 1) id: a unique identifier 2) family: shared name which identifies either paternal or maternal lineage 3) parents: male and female agents and the attributes of their offspring is a mix of the parents attributes. The first generation agents lack parents. Agents inherit their metabolism and their vision properties from their parents. 4) birthYear: specifies the start of the lifecycle of the agent 5) location: the current location of the agent 6) inheritance: initial allocation of sugar and spice received from parents 7) sugar: the total amount of sugar available for consumption, which is the summation of inherited sugar and the net sugar gathered, consumed, and traded 8) spice: the total amount of spice available for consumption, which is the summation of inherited spice and the net spice gathered, consumed, and traded.

The agents need both sugar and spice to survive and shortage of either will lead to death. They also have trading ability randomly assigned at birth. The behavioral modeling of agents has been implemented using some rules. For example the ruleset for gathering is as follow: 1) The agent determines which good is needed urgently i.e. sugar or spice, which is called preferred good. 2) The agent look at its vision range and find the cell with the highest value of the preferred good 3) Then the agent moves to that cell and consume the good 4) If there is no preferred good available in the vision range, the agent relocates to the farthest cells within its vision range.

2.2.7. Other predator-prey simulations

Some of the above mentioned systems like Polyworld and Echo model predator individuals. Other predator-prey models have also been presented focusing more on the ecological predator prey dynamics and interactions. Smith [79] uses the Volterra [80] model which exhibits constant population dynamics, both in terms of oscillations in global populations as well as dynamic patchiness. The model integrated 2D spatial representation to study migration under different predation strategies. He showed that detailed movement patterns in predator and prey can affect their interaction. Smith only models simple predator-prey behavior with simple genomic representation as only migration parameters are able to mutate. In [81] digital predator-prey organisms were used to study the evolution of trophic structure represented by the food web. Bell showed how different energy flow levels among organisms affect species richness and diversity. In another study [82], Lotka-Volterra equations were integrated in an IBM to examine how evolution of prey use by predators affects community stability and whether complexity of the food web increases stability of the predator prey system. The results demonstrated that the number of existing species decreases with increasing complexity.

A predator-prey simulation based on a spatial collection of individual finite state machine agents (animat) was first presented in [83]. This model can locate hundreds of thousands of individuals evolving in a two-dimensional featureless spatial plain. Every animat is represented using a small set of rules that direct its microscopic behavior and, at each time-step of the simulation, each animat executes one of these rules, causing it to: move, eat, or breed. In one study, the effect of introducing camouflage behavior as an available option for predators was investigated [84]. It was shown that individuals who adopt this behavior are relatively successful in obtaining prey and thus prolonging their lives against the threat of dying of hunger. This, in turn, led to higher numbers of successful older predators which caused a crash in the population of prey. In another study a time-delayed gestation period was introduced into the predator-prey selection and adaptation mechanisms [85]. The temporal behavior of individual animats was affected by the gestation period parameter and hence the macroscopic behaviors of the species were also affected.

2.3. EcoSim

Since, in this dissertation, EcoSim has been used to investigate several different biological questions, we give in this section a detailed description of EcoSim using the updated 7-point Overview-Design concepts-Details (ODD) standard protocol [86] for describing individual-based models. Several studies have validated some of the patterns observed in EcoSim. For example, in

[87], the species abundance patterns have been analyzed based on Fisher's log series. They demonstrated that their simulations produce results relating to species abundance patterns that cohere with patterns observed in real ecological systems. In another study, chaotic properties of the patterns generated by the system with multi-fractal properties has been established, which agrees with what has been observed for real ecosystems [88]. Golestani et al. [89] added small, randomly distributed physical obstacles into the simulations to investigate the influence of obstacles on the distribution of populations and species, the level of gene flow, as well as the mode and tempo of speciation. Hosseini et al. [90] were able to predict species extinction in EcoSim with high accuracy. These studies demonstrate the potential of EcoSim simulations to approximate some important features of real ecosystems, although admittedly it does have its limitations such as the absence of abiotic factors (climate, fluctuations in temperature, precipitation, wind, soil changes, or geographic features such as mountains, valleys, rivers, lakes).

2.3.1 Purpose

EcoSim is an individual-based predator-prey ecosystem simulation, which was designed to simulate agents' behavior in a dynamic, evolving ecosystem [6], [12]. The main purpose of EcoSim is to study biological and ecological theories by constructing a complex adaptive system, which leads to a generic virtual ecosystem with behaviors similar to those found in nature. EcoSim uses, for the first time, a fuzzy cognitive map (FCM) to model each agent behavior (see section 2.3.4.1). The FCM of each agent, being coded in its genome, allows the evolution of agents' behavior throughout the epochs of the simulation.

In EcoSim, all the factors determining the reproductive success of an individual are free of pre-defined fitness functions. The overall fitness of an individual, measured as its reproductive success and that of its offspring, depends only on the interaction between its phenotype (behavioral type) and the environment. These interactions result from the usage of the behavioral models of the individuals under various environmental circumstances. At each time step, the individuals in EcoSim consume some energy. This consumption is determined by a cost function that takes into account the complexity of the behavioral model of the individual (the number of edges it contains) and the action it performs. The more complex the model is and the faster the movements performed by the individual (such as escape and exploration) are, the more the energy is consumed. This cost function is pre-defined. Nevertheless, a cost function is not a fitness function since it does not determine the success of a particular behavioral model. A cost function is a 'fix penalty', which is assigned to behavioral models and actions independently of the environment in order to avoid an obvious continuous increase in the behavioral model complexity

and to model energy depletion with time. The success of a behavioral model relies on the tradeoff between the decisions it makes, knowing the current environment and the cost of the actions that are performed throughout the life of the individual. However, this tradeoff is not arbitrated by a predefined extrinsic function but results from the consequence of the actions undertaken.

As a consequence, decisions made by individuals with distinct behavioral models do not rely on any external evaluation (pre-defined fitness function) in the interest of the action. Instead, decisions rely on the knowledge 'learned' from the environment in the behavioral model by the evolutionary process, tuning behaviors to a particular state of the local world, and on the individual perception of the local environment. The model determining the reproductive success of an individual is thus intrinsic to the simulation in the sense that no external information is involved for determining fitness [91]. This feature is very important because the systems with pre-defined fitness function behave as a genetic algorithm. These systems are optimization processes that the fate of the system is directly determined by its pre-defined fitness function. When targeting unbounded evolution and emergence of new adaptive behavior, evolutionary algorithms (using extrinsic adaptation) should be rejected and rather a model based on natural selection (intrinsic adaptation) is more suitable.

2.3.2 Entities, state variables, and scales

Individuals: There are two types of individuals: predators and prey. Each individual possesses several life-history characteristics (see Table 2-1) such as age, minimum age for breeding, speed, vision distance, level of energy, and amount of energy transmitted to the offspring. Energy is provided to the individuals by the resources (food) they find in their environment. Prey consume primary resources, which are dynamic in quantity and location, whereas predators hunt for prey. Each individual performs one unique action during a given time step, based on its perception of the environment. Each agent possesses its own FCM coded in its genome and its behaviors are determined by the interaction between the FCM and the environment (see section 2.3.4.1). Energy is provided by the primary or secondary resources found in their environment. For example, prey individuals gain 250 units of energy by eating one unit of grass and predators gain 500 units of energy by eating one prey. At each time step, each agent spends energy depending on its action (e.g., breeding, eating, and running) and on the complexity of its behavioral model (number of existing edges in its FCM). On average, a movement action, such as escape and exploration, requires 50 units of energy whereas a reproduction action uses 110 units of energy and the choice of no action results in a small expenditure of 18 units of energy. These constant numbers obtained by trial and error while they are logically plausible.

Cells and virtual world: The smallest units of the environment are cells. Each cell represents a large space, which may contain an unlimited number of individuals and/or some amount of food. The virtual world consists of torus-like discrete 1000 × 1000 matrix of cells.

Table 2-1. Several physical and life history characteristics of individuals from 10 independent EcoSim runs.

Characteristic	Predator	Prey
Maximum age	42 time steps (+/- 6)	46 time steps (+/-18)
Minimum age of reproduction	8 time steps	6 time steps
Maximum speed	11 cells / time step	6 cells / time step
Vision distance	25 cells maximum	20 cells maximum
Level of energy at initialization	1000 units	650 units
Average speed	1.4 cells / time step (+/- 0.3)	1.2 cells / time step (+/- 0.2)
Average level of energy	415 units (+/- 82)	350 units (+/- 57)
Maximum level of energy	1000 units	650 units
Average number of reproduction action during life	1.14 (+/- 0.11)	1.49 (+/- 0.17)
Average length of life	16 time steps (+/- 5)	12 time steps (+/- 3)

Time step: Each time step involves the time needed for each agent to perceive its environment, make a decision, perform its action, as well as the time required to update the species membership, including speciation events and record relevant parameters (e.g., the quantity of available food). In terms of computational time, the speed of a simulation per generation is proportional to the number of individuals. An execution of the simulation with an average of 250 000 individuals simultaneously present in the world produced approximately 15000 time steps in 35 days.

Population and Species: On average, in each time step, there are about 250,000 individuals, members of one or more species. A species is a set of individuals with a similar genome relative to a maximum dissimilarity threshold.

2.3.3 Process overview and scheduling

The possible actions for the prey agents are: exploring the environment to gain information regarding food, predators, and sexual partners, evasion (escape from predator), search for food (if there is not enough grass available in its habitat cell, prey can move to another cell to find grass), socialization (moving to the closest prey in the vicinity), exploration, resting (to save energy), eating and breeding. Predators also perceive the environment to gather information used to choose an action from amongst: hunting (to catch a prey), search for food, socialization, exploration, resting, eating and breeding. After each action, the individuals' energy is adjusted and their age in incremented by one. There are also two environmental processes: after all individuals perform their actions, the amount of grass and meat are adjusted.

At each time step, the value of the state variables of individuals and cells are updated. The overview and scheduling of every time step is as follows (algorithm):

1. For each prey individual:
    1.1. Perception of the environment
    1.2. Computation of the next action
    1.3. Performing actions and updating the energy level
2. Updating the list of prey (it's done once for all prey individuals)
3. Updating prey species (it's done once for all prey individuals)
4. For each predator individual:
    4.1. Perception of the environment
    4.2. Computation of the next action
    4.3. Performing their action and update of the energy level
5. Updating the list of predator individuals (it's done once for all predator individuals)

6. Updating predator species (it's done once for all predator individuals)

7. For each cell in the world:

    7.1. Updating the grass level

    7.2. Updating the meat level

8. Updating of the age of the individuals

The complexity of the simulation algorithm is linear with respect to the number of individuals. If we consider that there are N_1 prey and N_2 predators and we exclude the sorting parts, which have a complexity of $O(N_1 log N_1)$ and $O(N_2 log N_2)$ but are negligible in the overall computational time as they are only performed once per time step, then the complexity of part 1 and part 2 of the above algorithm, including the clustering algorithm used for speciation, will be $O(N_1)$ and $O(N_2)$ respectively [92]. The virtual world of the simulation has 1000×1000 cells, therefore the complexity of part 3 will be $O(k = 1000 \times 1000)$. The complexity of part 4 will be $O(N_1+N_2)$. As a result, the overall complexity of the algorithm is $O(2N_1+ 2N_2+ k)$, which is $O(N)$.

2.3.4 Design concepts

2.3.4.1 Basic principles

To observe the evolution of individual behavior and ultimately ecosystems over thousands of generations, several conditions need to be satisfied: (i) every individual should possess genomic information; (ii) this genetic material should affect the individual behavior and consequently its fitness; (iii) the inheritance of the genetic material has to be done with the possibility of modification; (iv) a sufficiently high number of individuals should coexist at any time step and their behavioral model should allow for complex interactions and organizations to emerge; (v) a model for species identification, based on a measure of genomic similarity, has to be defined; and (vi) a large number of time steps need to be performed. These complex conditions pose computational challenges and require the use of models that combine the compactness and ease of computation with a high potential for complex representation.

In EcoSim, a Fuzzy Cognitive Map (FCM) [93] is the base for describing and computing the agent behaviors. Each agent possesses an FCM to compute its next action. The FCM is integrally coded in their genomes and therefore heritable and subject to evolution. FCMs are weighted graphs representing the causal relationship between concepts, allowing the observation of

evolutionary patterns and inference of underlying processes (Figure 2-1) (see section 2.3.4.2 and 2.3.4.6). When a new offspring is created, it is given a genome, which is a combination of the genomes of its parents with some possible mutations.

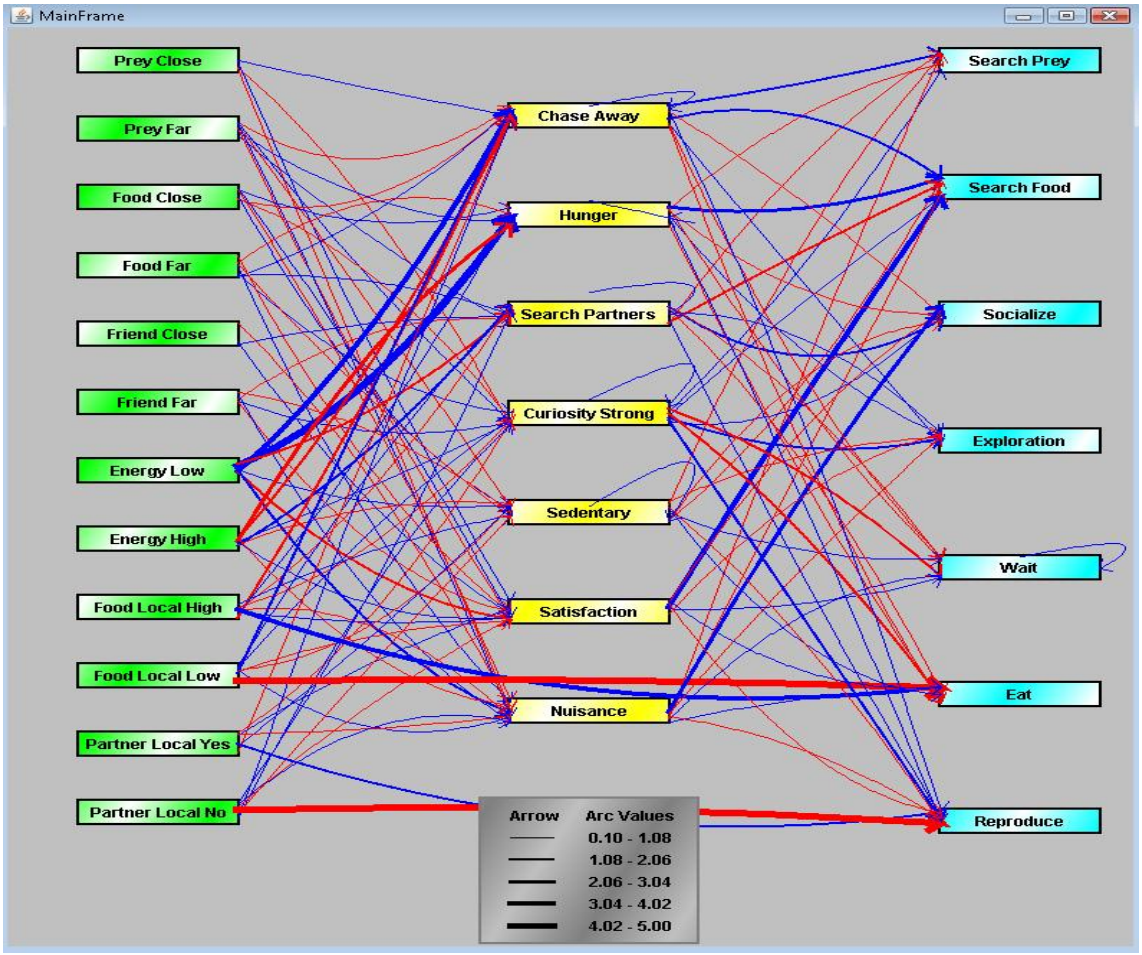


Figure 2-1. A sample of a predator's FCM including concepts and edges. The width of each edge shows the influence value of that edge. Color of an edge shows inhibitory (red) or excitatory (blue) effects.

Formally, an FCM is a graph, which contains a set of nodes C, each node C_i being a concept, and a set of edges I; each edge I_{ij} representing the influence of the concept C_i on the concept C_j. A positive weight associated with the edge I_{ij} corresponds to an excitation of the concept C_j from the concept C_i, whereas a negative weight is related to an inhibition (a zero value indicates that there is no influence of C_i on C_j). The influence of the concepts in the FCM can be represented in an n×n matrix, L, in which L_{ij} is the influence of the concept C_i on the concept C_j. If $L_{ij} = 0$, there is no edge between C_i and C_j. In EcoSim, each individual genome code for its proper FCM, with one gene coding for one weight L_{ij}.

In each FCM, three kinds of concepts are defined: sensitive (such as distance to foe or food, amount of energy, etc.), internal (fear, hunger, curiosity, satisfaction, etc.), and motor (evasion, socialization, exploration, breeding, etc.). The activation level of a sensitive concept is computed by performing a fuzzification of the information the individual perceives in the environment. For an internal or motor concept, C, the activation level is computed by applying the defuzzification function on the weighted sum of the current activation level of all the concepts having an edge directed toward C. Finally, the action of an individual is selected based on the maximum value of motor concepts' activation level. Activation levels of the motor concepts are used to determine the next action of the individual. For example, Figure 2-2 represents two sensitive concepts (foeClose and foeFar), one internal (fear), and one motor (evasion). There are also three influence edges: closeness to a foe excites fear, distance to a foe inhibits fear, and fear causes evasion. Activations of the concepts foeClose and foeFar are computed by fuzzification of the real value of the distance to the foe, and the defuzzification of the activation of evasion tells us about the speed of the evasion (see section 2.3.4.6).

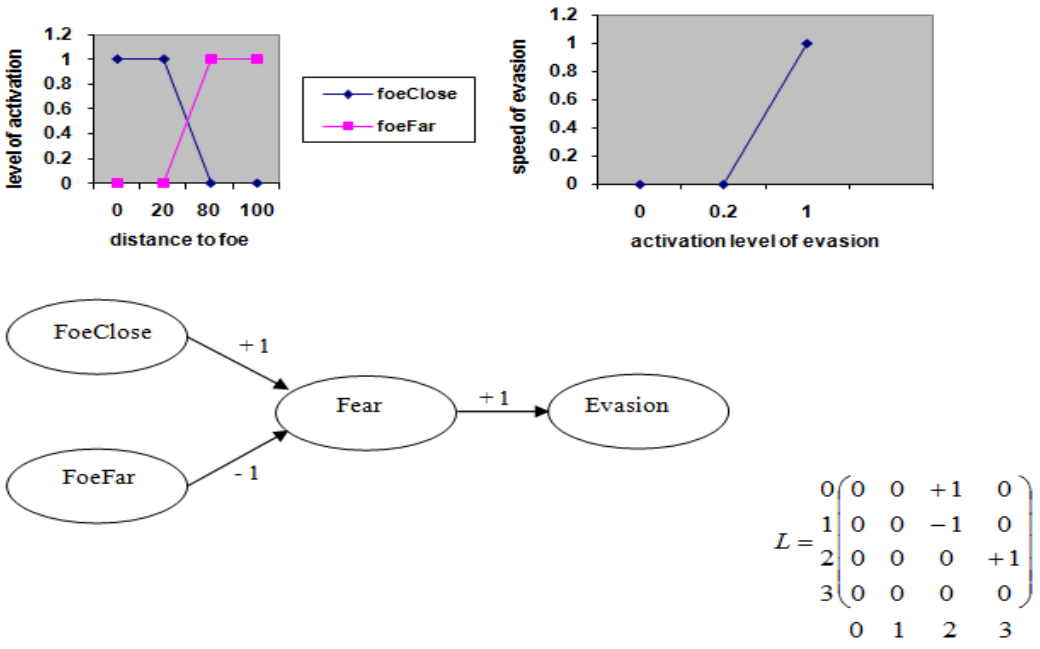


Figure 2-2. An FCM for detection of foe (predator) and decision to evade, with its corresponding matrix (0 for 'Foe close', 1 for 'Foe far', 2 for 'Fear' and 3 for 'Evasion') and the fuzzification and defuzzification functions [94].

2.3.4.2 Emergence

The behavioral model of individuals encoded in an FCM can react to the changes in the environment. For example, it has been shown that the contemporary evolution of prey behavior owing to predator removal is also accompanied by prey genetic change [95]. At the initiation of the simulation, prey and predators are scattered randomly all around the virtual world. Through the epochs of the simulation, the distribution of the individuals in the world is changed drastically based on many different factors: prey escaping from predators, individuals socializing and forming groups, individuals migrating gradually to find sources of food, species emerging, etc. The size of the world is large enough to accommodate population structures and the emergence of migrations. For example, an individual moving at its maximum speed could barely cross half of the world during its life span. Moreover, previous studies demonstrate that the usage of behavioral models lead to a non-random distribution of individuals and species in which individuals form populations that contain agents with similar genomes [89], [96]. Figure 2-3 shows an example of a snapshot of the virtual world after thousands of time steps with emerging grouping patterns.

It has been shown that the data generated by EcoSim present the same kind of multifractal properties as those observed in real ecosystems [97]. Individuals' distribution forming spiral waves is one property of prey-predator models and it is an emerging property in EcoSim (Figure 2-3). Prey near the wave break have the capacity to escape from the predators sideways. A subpopulation of prey then finds itself in a region relatively free from predators. In this predator-free zone, prey start expanding extensively, forming a circularly expanding region. The whole pressure process and spiral formation will be applied to this subpopulation of prey and predators, leading to the formation of a second scale [98]. This process repeats many times, which is a common property of self-similar processes [99]. Because there are consecutive interactions between prey and predators over time, the same pattern repeats itself over and over. The result of this pattern repetition is the emergence of self-similarity in the spatial distribution of individuals. In addition, migration phenomena can be observed, since the relocation of individuals leads to the redistribution in the population [100].

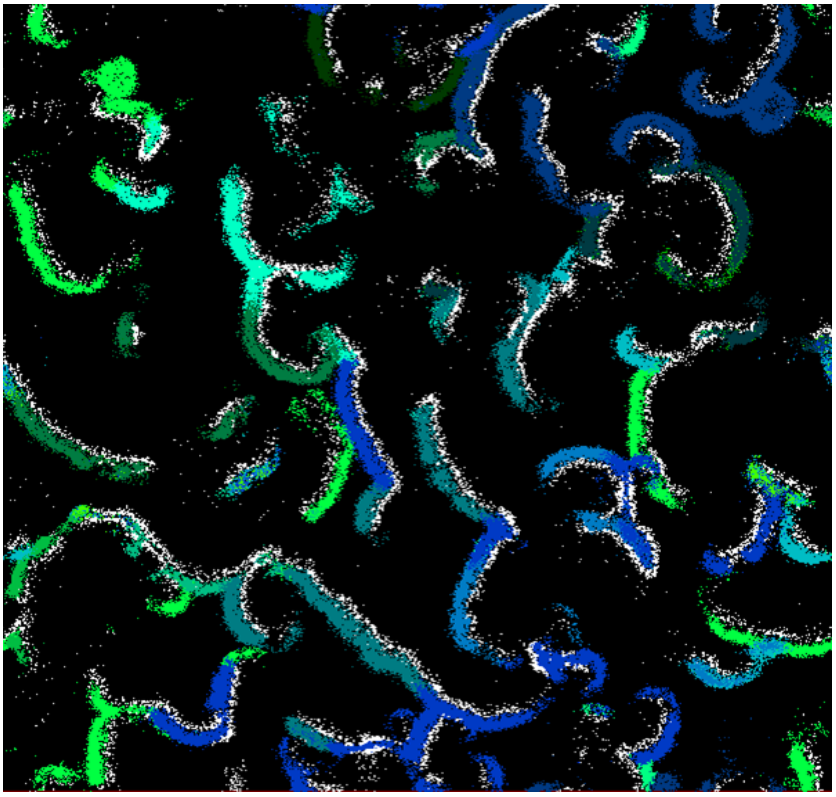


Figure 2-3. A snapshot of the virtual world in one specific time step, white color represents predator species and the other colors show different prey species.

2.3.4.3 Adaptation

The genome maximum length is fixed (390 sites), where each site is a real number and corresponds to an edge between two concepts of the FCM and code for the weight associated to this edge. However, as many edges have an initial value of zero, only 114 edges for prey and 107 edges for predators exist at initialization (see section 2.3.4.1). One more gene is used to code for the amount of energy, which is transmitted from the parents to their child at birth. The value of a site, which is a real number, corresponds to the intensity of the influence between the two concepts. The genome of an individual is transmitted to its offspring after being combined with the genome of the other parent and following the possible addition of some mutations. To model linkage, the weights of edges are transmitted by blocks from parents to the offspring. For each concept, its entire incident edges' values are transmitted together from the same randomly chosen parent. The behavioral model of each individual is therefore unique. Step after step, as more individuals are created, changes in the FCM occur due to the formation of new edges (with probability of 0.001), removal of existing edges (with probability of 0.0005) and changes in the weights associate to existing edges (with probability of 0.005). These low probabilities, compared

to the crossover probability, reflect the fact that changes in genome should be relatively slow to avoid random evolution. Therefore, new genes may emerge from among the 265 initial edges of zero value.

2.3.4.4 Fitness

We calculated the fitness for each species as the average fitness of its component individuals. In order to realistically represent the capacity of an individual to survive and produce offspring that can also survive, fitness was calculated as the sum of age at death of the focal individual with the death age of its children (a post-processing computation). Since the sum involves all direct offspring, it is representative of the fertility and survivability of the individuals [101]. It is important to notice that it is a post-processing computation done only to analyze the results generated by the simulation and that this fitness is never using during the simulation process itself.

2.3.4.5 Prediction

So far, there is no learning mechanism for individuals during their life and they cannot predict the consequences of their decision. The only available information for every individual to make decisions is the information coming from their perceptions at that particular time step and the value of the activation level of the internal and motor concepts at the previous time steps. The activation levels of the concepts of an individual are never reset during its life. As the previous time step activation level of a concept is involved in the computation of its next activation level, this means that all previous states of an individual during its life participate in the computation of its current state. Therefore, an individual has a basic memory of its own past that will influence its future states.

2.3.4.6 Sensing

Every individual in EcoSim is able to sense its local environment inside its range of vision. For instance, each prey can sense its five closest foes, cells with food units, mates within its range of vision, the number of grass units in its cell and the number of possible mates in its cell. Moreover, each individual is capable of recognizing its current level of energy.

It should be noted that the FCM process explained in section 2.3.4.2, enables, for example, distinguishing between perception and sensation: sensation is the real value coming from the environment, and perception is sensation modified by an individual's internal states. For example, it is possible to add three edges to the map presented in Figure 2-2: one auto excitatory edge from

the concept of fear to itself, one excitatory edge from fear to foeClose, and one inhibitory edge from fear to foeFar (Figure 2-4). A given real distance to the foe seems higher or lower to the individual depending on the activation level of fear. Also, the fact that the individual is frightened at time t influences the level of fear of the individual at time t + 1. This kind of mechanism makes possible the modeling of the degree of stress for an individual. It also enables the individual to memorize information from previous time steps: fear maintains fear. It is therefore possible to build very complex dynamic systems involving feedback and memory using an FCM, which is needed to model complex behaviors and abilities to learn from evolution.

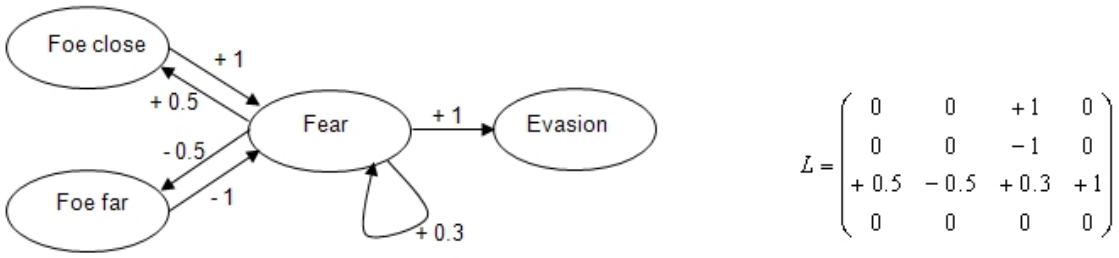


Figure 2-4. An FCM for detection of foe (predator) - difference between perception and sensation [94]**. This map shows different kind of interactions between three kinds of concepts: perception concept (Foe close and Foe far), internal concept (Fear) and motor concept (Evasion).**

2.3.4.7 Interaction

The only action that requires a coordinate decision of two individuals is reproduction. For reproduction to be successful, the two parents need to be in the same cell, to have sufficient energy, to choose the reproduction action and to be sufficiently genetically similar. The individuals cannot determine their genetic similarity with their potential partner. However, if they try to mate and the potential partner is too dissimilar (the difference between the two genomes is greater than a specified threshold (half of the speciation threshold), then the reproduction fails.

Predator's hunting introduces another type of interaction in the simulation. For a predator to succeed in the hunting action, its distance to the closest prey is required to be less than one cell. When a predator's hunting action succeeds, a new meat unit is added to the corresponding cell, and the energy level of the predator is also increased by one unit of meat energy.

Furthermore, there is a competition for prey and predators for food. For example, if in a given cell there is only one food unit and two agents have chosen the action of eating, the younger will act

first, and so it will be the only one that can eat (in this cell) at this time step. This is a way to simulate the fact that older species members help younger species members to survive.

2.3.4.8 Stochasticity

To produce variability in the ecosystem simulation, several processes involve stochasticity. For instance, at initialization, the number of grass units is randomly determined for each cell. Moreover, the maximum age of an individual is determined randomly at birth from a uniform distribution centered at a value associated with the type of agent (see section 2.3.5). Stochasticity is also included in several kinds of actions of the individuals such as evasion and socialization. If there is no predator or partner respectively in the vision range of the individual, the direction of the movement would be random. Furthermore, the direction of the exploration action is always random.

However, to understand the extent of randomness in EcoSim, Golestani et al. examined whether chaotic behavior exists in signals (time series) generated by the simulation. They concluded that the EcoSim is capable of generating non-random and chaotic pattern (time series) [102].

2.3.4.9 Collectives

In EcoSim, the notion of species is implemented in a way that species emerge from the evolving population of agents. EcoSim implements a species concept directly related to the genotypic cluster definition [103] in which a species is a set of individuals sharing a high level of genomic similarity. In addition, in EcoSim, each species is associated with the average of the genetic characteristics of its members, called the ‘species genome’ or the ‘species center’. The speciation method involves a 2-means clustering algorithm [92] in which an initial species is split into two new species, each of them containing the agents that are mutually the most similar. Over time, a species will progressively contain individuals that are increasingly genetically dissimilar up to an arbitrary threshold where the species splits. After splitting, the two sister species are sufficiently similar that hybridization events can occur. Therefore, two individuals can interbreed if their genomic distance is smaller than an arbitrary threshold (half of the speciation threshold) even if they are designated as members of two sister species by our clustering algorithm. The information about species membership is only a label. It is not used for any purpose during the simulation but only for post-processing analysis of the results. Several studies have been conducted to analyze the concept of species in EcoSim. Devaurs & Gras [104] compared the species abundance patterns emerging from EcoSim with those observed in natural ecosystems using Fisher's logseries [105]. Species abundance is a key component of macro-ecological theories and Fisher's

logseries is one of the most widely known classic models of species abundance distribution. The results of this study proved that at any level in sample size, EcoSim generates coherent results in terms of relative species abundance, when compared with classical ecological results [28]. In another study, Golestani et al. [89] investigated how small, randomly distributed physical obstacles influence the distribution of populations and species, showing that there is a direct and continuous increase in the speed of evolution (e.g. the rate of speciation) with the increasing number of obstacles in the world.

2.3.4.10 Observation

EcoSim produces a large amount of data in each time step, including number of individuals, new and extinct species, geographical and internal characteristics of every individual, and status of the cells of the virtual world. Information regarding each individual includes position, level of energy, choice of action, specie, parents, FCM, etc. There is also the possibility to store all of the values of every variable in the current state of the simulation in a separate file, making possible the restoration of the simulation from that state onwards. All of the data is stored in a compact special format, to facilitate the storage and future analysis.

2.3.5 Initialization and input data

A parameter file is used to assign the values for each state variable at initialization of the simulation. These parameters are as follows: width and height of the world, initial numbers of individuals, threshold of genetic distance for prey/predator speciation, maximum age, energy, speed, vision range, and initial values of FCM for prey/predator. Any of these parameters can be changed for specific experiments and scenarios. An example of a list of the most common user-specified parameters is presented in Table 2-2. For other initial parameters see Table 2-3 to Table 2-8.

Different values of initial parameters can lead to an extinction of either the prey or the predators or both of them. The current values lead to stable runs for the simulation. Some parameters like number of individuals are less sensitive than other. However, as long as the equilibrium between amount of grass, number of prey, and number of predators is maintained, the whole system is quite stable and many different combinations of values still tested have led to stable runs. Moreover, as far as the runs are stables, all the general patterns behavior described in section 2.3 emerged and have been observed systematically.

Table 2-2. Values for user-specified parameters in EcoSim.

User Specified Parameter	Used Value
Number of Prey	12000
Number of Predators	500
Grass Quantity	5790000
Maximum Age Prey	46
Maximum Age Predator	42
Prey Maximum Speed	6
Predator Maximum Speed	11
Prey Energy	650
Predator Energy	1000
Distance for Prey Vision	20
Distance for Predator Vision	25
Reproduction Age for Prey	6
Reproduction Age for Predator	8

2.3.6 Submodels

As mentioned earlier, each individual performs one unique action during a time step based on its perception of the environment. Each time step of EcoSim consists of the computation of the activation level of the concepts, the choice and application of an action for every individual. A time step also includes the update of the world: emergence and extinction of species and growth and diffusion of grass, or decay of meat.

At initialization time there is no meat in the world and the number of grass units is randomly determined for each cell. For each cell, there is a probability, probaGrass, that the initial number of units is strictly greater than 0. In this case, the initial number is generated uniformly between 1 and maxGrass. Each unit provides a fixed amount of energy to the agent that eats it. The preys can only eat the grass, and the predators have two modes of predation: hunting and scavenging. When a predator's hunting action succeeds, a new meat unit is added in the corresponding cell and the predator is considered consuming another one. When a predator's eating action succeeds

(which can be viewed as a scavenging action), one unit of meat is removed in the corresponding cell. The amount of energy is energyGrass for one grass unit when eaten by a prey and is energyMeat for one meat unit eaten by a predator. The number of grass units grows at each time step up to maxGrass, and when a prey dies in a cell, the number of meat units in this cell increases by 2, up to maxMeat. The number of grass units in a cell decreases by 1 when a prey eats, and the number of meat units decreases by 1 when a predator eats. The number of meat units in a cell also decreases at each time step, even if no meat has been eaten in this cell. For every action there is a cost, which is associated with the individuals' energy level and is updated based on the number of FCM arcs (nbArcs) and the individual's speed (equation 2-1).

$$e_{t+1} = e_t - (nbArcs / 4 + speed^{0.25}) \tag{2-1}$$

For the reproduction action, there is an extra cost for parents, which is based on following relations.

$$e_{t+1} = e_t - e_{nb} / 2$$
$$e_{nb} = \begin{cases} MaxEnergy \times (rand(Maxsob - sob) + sob) / 100 & if\ (Maxsob - sob) \geq 1 \\ MaxEnergy \times sob / 100 & otherwise \end{cases} \tag{2-2}$$

Where e_{nb} is new born energy, rand is random function, sob is state of birth (parental energy investment) and Maxsob is the maximum value for sob.

1. Evasion (for prey only). The evasion direction is the direction opposite to the direction of the barycenter of the 5 closets foes within the vision range of the prey, with respect to the current position of the prey. If no predator is within the vision range of the prey, the direction is chosen randomly. Then the new position of the prey is computed using the speed of the prey and the direction. The current activation level of fear is divided by 2.

2. Hunting (for Predator only). The predator selects the closest cell (including its current cell) that contains at least one prey and moves towards that cell. If it reaches the corresponding cell based on its speed, the predator kills the prey, eating one unit of food and having another unit of food added to the cell. When there are several prey in the destination cell, one of them is chosen randomly. If the speed of the predator is not enough to reach the prey, it moves at its speed toward this prey. If there are no prey in the current cell and in the vicinity or it does not have enough energy to reach a prey, hunting action is failed.

3. Search for food. The direction toward the closest food (grass or meat) within the vision range is computed. If the speed of the agent is high enough to reach the food, the agent is placed on the cell containing this food. Otherwise, the agent moves at its speed toward this food.

4. Socialization. The direction toward the closest possible mate within the vision range is computed. If the speed of the agent is high enough to reach the mate, the agent is placed on the cell containing this mate, and the current activation level of sexualNeeds is divided by 3. Otherwise, the agent moves at its speed toward this mate. If no possible mate is within the vision range of the agent, the direction is chosen randomly.

5. Exploration. The direction is computed randomly. The agent moves at its speed in this direction. The activation level of curiosity is divided by 1.5.

6. Resting. Nothing happens.

7. Eating. If the current number of grass (of meat) units is greater than 1, then this number is decreased by 1 and the prey's (predator's) energy level is increased by energyGrass (energyMeat). Its activation level for hunger is divided by 4. Otherwise nothing happens.

8. Breeding. The following algorithm is applied to the agent A:

if A.energyLevel > 0.125 × maxEnergyPrey then

for all A of the same type in the same cell

if A.energyLevel > 0.125 × maxEnergyPrey and D(A,A') < T and

A′ has not acted at this time step yet and

A's choice of action is also breeding

then

interbreeding(A,A')

A.sexualNeeds ← 0

A'.sexualNeeds ← 0

If A' satisfies all the criteria, the loop is canceled

If none of the A' agents satisfies all the criteria, the breeding action of A fails.

For every action requiring that the agent move, its speed is computed by the formula

Speed = Ca × maxSpeedPrey => for the preys

Speed = Ca × maxSpeedPredator => for the predators

with Ca the current activation level of the motor concept associated with this action.

The process of generating a new offspring (interbreeding function) consists of following steps. First, the value of birthEnergyPrey is transmitted with possible mutations from one randomly chosen parent to the offspring. Second, the edges' values are transmitted with possible mutations, and the initial energy of the offspring is computed. To model the crossover mechanism, the edges are transmitted by block from one parent to the offspring. For each concept, its incident edges' values are transmitted together from the same randomly chosen parent. Third, the maximum age of the offspring is computed. Finally, the energy level of the two parents is updated.

Table 2-3. The initial parameters of the EcoSim at the first time step of the simulation. There are 42 parameters for each run of EcoSim. The value of these parameters has been obtained empirically and by biologists' expert opinion to preserve the equilibrium in the ecosystem.

Parameter	Initial Value	Comments
Width	1000	width of the world
Height	1000	height of the world
ProbaGrass	0.187	initial probability of grass per cell
ProbaGrowGrass	0.0028	probability of diffusion of grass
ValueGrass	250	energy value for a consumed grass
ValuePrey	500	energy value for a consumed prey
MaxGrass	8	maximum number of grass in a cell
SpeedGrowGrass	0.5	speed of growing grass
MaxMeat	8	maximum number of meat in a cell
NbResources	2	number of food resources in the world
ProbaMut	0.005	probability of mutation to a nonzero gene
ProbaMutLow	0.001	probability of mutation to a zero gene
MinArc	0.075	threshold for an arc to be counted as nonzero
InitNbPrey	12000	initial number of prey
InitNbPredator	2000	initial number of predator
DistanceSpeciesPrey	1.5	threshold of genetic distance for prey species
DistanceSpeciesPred	1.3	threshold of genetic distance for predator species
AgeMaxPrey	46	maximum age for prey
AgeMaxPred	42	maximum age for predator
AgeReprodPrey	6	minimum reproduction age for prey
AgeReprodPred	8	Minimum reproduction age for predator
ClusterPrey	10	number of prey per clusters at initialization
ClusterPredator	20	number of predators per clusters at initialization
RadiusCluster	5	radius in number of cell of each initial cluster
EnergyPrey	650	maximum energy of prey
EnergyPredator	1000	maximum energy of predator

SpeedPrey	6	maximum speed of prey
SpeedPredator	11	maximum speed of predator
VisionPrey	20	maximum vision of prey
VisionPredator	25	maximum vision of predator
StateBirthPrey	30	initial parental energy investment for prey
StateBirthPred	40	initial parental energy investment for predator
nbSensPrey	12	number of sensitive concepts in prey
nbConceptsPrey	7	number of internal concepts in prey
nbMotorPrey	7	number of motor concepts in prey
nbSensPredator	12	number of sensitive concepts in predator
nbConceptsPredator	7	number of internal concepts in predator
nbMotorPredator	7	number of motor concepts in predator
Restore	1	0-no restore, 1-restore
MaxSave	500	0-no save, #-save every # states
MinSave	0	0-no save, #-save every # states
WorldSave	0	0-no save, 1-save world

Table 2-4. Initial FCM values for Prey (See Table 2-5). Every prey individual has a FCM which represents its behavior. At first time step, all prey individuals have an initial FCM. During time and during each generation with operators like crossover and mutation, the FCM of individuals change.

	FR	**HG**	**SP**	**CU**	**SD**	**ST**	**NU**	**ES**	**SF**	**SC**	**XP**	**WT**	**ET**	**RP**
PC	4	0	0	0.1	0	-1	1	0	0	0	0	0	0	0
PF	-4	0	0	0	0	0.5	-0.5	0	0	0	0	0	0	0
OC	0	0.5	0	-0.1	0.1	0.5	-0.5	0	0	0	0	0	0	0
OF	0	0	-0.4	0.2	-0.2	-0.7	0.7	0	0	0	0	0	0	0
FC	0	0	0.5	-0.1	0.1	0.5	-0.5	0	0	0	0	0	0	0
FF	0	0	-0.4	0.2	-0.2	-0.5	0.5	0	0	0	0	0	0	0
EL	0.4	4	-1.5	0	0	-2.2	2.2	0	0	0	0	0	0	0
EH	0	-1	1.5	0.2	-0.2	1.5	-1.5	0	0	0	0	0	0	0
OH	0	-0.2	0	-0.3	0.3	1.1	-1.1	0	0	0	0	0	2.6	0
OL	0	0.2	0	1	-1	-1.1	1.1	0	0	0	0	0	-4	0
PY	0	0	0	-0.4	0.4	0.5	-0.5	0	0	0	0	0	0	1.5
PN	0	0	0.5	0.3	-0.3	-0.8	0.8	0	0	0	0	0	0	-4
FR	0.5	0	0	0	0	0	0	1.5	-0.8	-1	0.3	-1	-1	-1
HG	0	0.3	0	0	0	0	0	-0.8	2.1	-0.7	0.7	-0.5	4	-1.8
SP	0	0	0.2	0	0	0	0	-0.2	0	1.5	0.5	-0.3	-0.4	3
CU	0	0	0	0.1	0	0	0	-0.1	0.5	0.3	1.5	-0.2	-0.3	-0.2
SD	0	0	0	0	0.1	0	0	0	-0.5	-0.3	-1.2	0.2	0.3	0.2
ST	0	0	0	0	0	0	0	-0.1	-0.8	-0.2	-2	1.5	0.8	0.7
NU	0	0	0	0	0	0	0	0.4	1	0.2	2	-1.2	-0.7	-0.7
ES	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SF	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SC	0	0	0	0	0	0	0	0	0	0	0	0	0	0
XP	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WT	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0
ET	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RP	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2-5. Prey/predator FCM abbreviation table. The abbreviation used to present concepts of FCM in EcoSim. These abbreviations have been used in other tables to show values of these concepts.

NodeName	Abbreviation	NodeName	Abbreviation
Fear	FR	PredClose	PC
Hunger	HG	PredFar	PF
SearchPartner	SP	FoodClose	OC
CuriosityStrong	CU	FoodFar	OF
Sedentary	SD	FriendClose	FC
Satisfaction	ST	FriendFar	FF
Nuisance	NU	EnergyLow	EL
Escape	ES	EnergyHigh	EH
SearchFood	SF	FoodLocalHigh	OH
Socialize	SC	FoodLocalLow	OL
Exploration	XP	PartnerLocalYes	PY
Wait	WT	PartnerLocalNo	PN
Eat	ET	PreyClose	YC
Reproduce	RP	PreyFar	YF
ChaseAway	CA		
SearchPrey	SY		

Table 2-6. Parameters of prey defuzzification function (see Figure 2-5). The function that has been used for fuzzifications uses three parameters which shape the fuzzification curve.

NodeName	Activation	Fuzzy Parameter1	Fuzzy Parameter2	Fuzzy Parameter3
PredClose	0	1	3.5	3.5
PredFar	0	2	3.5	3.5
FoodClose	0	1	6	6
FoodFar	0	2	6	6
FriendClose	0	1	5	5
FriendFar	0	2	5	5
EnergyLow	0	1	4	4
EnergyHigh	0	2	4	4
FoodLocalHigh	0	2	4	4
FoodLocalLow	0	1	4	4
PartnerLocalYes	0	2	1000	20
PartnerLocalLow	0	1	1000	20
Fear	0	0	1	3.5
Hunger	0	0	1	3
SearchPartner	0	0	1	3
Curiosity	0	0	1	2.5
Sedentary	0	0	1	2.5
Satisfaction	0	0	1	3
Nuisance	0	0	1	3
Escape	0	0	1	3.5
SearchFood	0	0	2	3
Socialize	0	0	4	3
Exploration	0	0	6	2.5
Wait	0	0	7	3
Eat	0	0	8	3.5
Reproduce	0	0	10	3.5

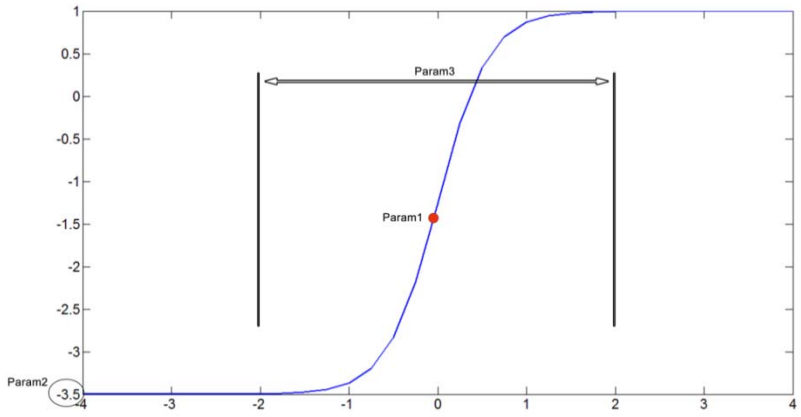


Figure 2-5. The three parameters that specify the shape of the curve. The first parameter specifies the center of curve in the horizontal axis, the second parameter specifies the lower band of curve in the vertical axis and the third parameter specifies the width of curve.

Table 2-7. Initial FCM for Predator (See Table 2-5). Every predator individual has a FCM which represent its behavior. At first time step, all predator individuals have an initial FCM. During time and during each generation with operators like crossover and mutation, the FCM of individuals change

	CA	HG	SP	CU	SD	ST	NU	SY	SF	SC	XP	WT	ET	RP
YC	0.7	0	0	-0.1	0	0.5	-0.5	0	0	0	0	0	0	0
YF	-0.5	0.7	0.1	0.4	-0.4	-0.5	0.5	0	0	0	0	0	0	0
OC	-0.5	0.7	0	-0.1	0.1	0.5	-0.5	0	0	0	0	0	0	0
OF	0.8	-0.2	0.1	0.2	-0.2	-0.6	0.6	0	0	0	0	0	0	0
FC	0	0	0.7	0	0	0.4	-0.4	0	0	0	0	0	0	0
FF	0	0	-0.5	0.3	-0.3	-0.4	0.4	0	0	0	0	0	0	0
EL	3.5	5	-1.2	0	0.2	-1.5	1.5	0	0	0	0	0	0	0
EH	-2	-3	1.4	0.3	-0.3	1	-1	0	0	0	0	0	0	0
OH	-1.5	0.3	-0.2	-0.3	0.3	1	-1	0	0	0	0	0	4	0
OL	1.7	0	0.2	1	-1	-1	1	0	0	0	0	0	-5	0
PY	-0.3	0	0	-0.4	0.4	0.8	-0.8	0	0	0	0	0	0	2
PN	0.3	0	0.5	0.3	-0.3	-0.8	0.8	0	0	0	0	0	0	-5
CA	0.2	0	0	0	0	0	0	1.5	-0.2	-0.4	0.3	-0.4	0	-0.4
HG	0	0.3	0	0	0	0	0	4	2.5	-1.2	0.3	-0.4	3.5	-0.8
SP	0	0	0.2	0	0	0	0	-0.8	-0.8	1.5	0.3	-0.5	-0.6	3
CU	0	0	0	0.1	0	0	0	0.3	0.3	0.3	1.5	-0.4	-0.3	-0.2
SD	0	0	0	0	0.1	0	0	-0.3	-0.3	-0.3	-1.5	0.4	0.3	0.2
ST	0	0	0	0	0	0	0	-0.8	-0.8	-0.2	-1.8	1	0.8	0.8
NU	0	0	0	0	0	0	0	1	0.8	0.2	2	-1	-0.6	-0.8
SY	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SF	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SC	0	0	0	0	0	0	0	0	0	0	0	0	0	0
XP	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WT	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0
ET	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RP	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2-8. Parameters of predator defuzzification function (see Figure 2-5). The function that has been used for fuzzifications uses three parameters which shape the fuzzification curve.

NodeName	Activation	Fuzzy Parameter1	Fuzzy Parameter2	Fuzzy Parameter3
PreyClose	0	1	4	4
PreyFar	0	2	4	4
FoodClose	0	1	5	5
FoodFar	0	2	5	5
FriendClose	0	1	5	5
FriendFar	0	2	5	5
EnergyLow	0	1	4.5	4.5
EnergyHigh	0	2	4.5	4.5
FoodLocalHigh	0	2	1000	20
FoodLocalLow	0	1	1000	20
PartnerLocalYes	0	2	1000	20
PartnerLocalNo	0	1	1000	20
ChaseAway	0	0	1	3
Hunger	0	0	1	3.5
SearchPartner	0	0	1	3
Curiosity	0	0	1	2.5
Sedementary	0	0	1	2.5
Satisfaction	0	0	1	3
Nuisance	0	0	1	3
SearchPrey	0	0	1	3
SearchFood	0	0	3	3.5
Socialize	0	0	5	3
Exploration	0	0	7	2.5
Wait	0	0	8	3
Eat	0	0	9	3.5
Reproduce	0	0	11	3.5

Chapter 3

3. Rule Extraction

3.1. Introduction

Neural networks (NNs), support vector machines (SVMs), and ensemble methods have shown very good performance on the various data sets in different fields. However, they lack explanation ability as they construct black-box models that cannot explain their prediction results. Therefore, inferring the logic behind their constructed models is not straightforward. This is one of the important features of the predictive models in several fields of studies such as medical diagnosis, credit scoring, and computational biology [106], [107], [108]. Rule extraction (RE) methods are a solution to overcome to this problem. When knowledge encoded in a predictive model is more important than its prediction outputs, instead of using an opaque or black-box model, RE, as a white box model, is more beneficial. For example, financial institutions are required to explain specific reasons in case of credit application rejection [107]. In the medical domain, patients expect human-understandable explanations instead of using black box diagnosis systems for physician acceptance. It can also reduce the likelihood of application of regulatory barriers that limit the usage of black-box models for the medical-decision support systems. For example, in United States, there are some restrictions on usage of black-box models that can impact patient treatment [109]. Ecologists are also interested in interpreting fundamental rules behind ecological phenomena when they apply predictive models as a data analysis tool [27], [28], [29]. Also, there are a lot of applications of RE in bioinformatics such as in [110]. In general, automatic knowledge acquisition, induction of scientific theories, and studying the general pattern and behavior of a predictive model is the main aim of rule extraction [111]. RE is a method of presenting a comprehensive description of a predictive model and at the same time approximates the predictive model as accurately as possible. More formally, given an opaque model to predict hypothesis h: f(x) = y and the data set (x,y) on which it is trained where x is a set of features and y is the corresponding output vector, RE produces a description of the h, i.e. h' such that h' is understandable (or comprehensible) yet h ≈ h', which means that h' approximates h as closely as possible [112].

When we talk about explaining the underlying predictive model, a compact and comprehensive representation model i.e., small number of rules and, more preferably, small number of features is desirable. There are different tradeoffs when considering a model for RE. First tradeoff is between accuracy and comprehensibility so that one should be sacrificed to obtain the other. In

other words, a higher number of rules can give better accuracy while it diminishes comprehensibility. For example, decision tree generates a comprehensive model especially by increasing the pruning rate. However, the accuracy can be weakened especially when it is compared with other methods such as SVM and ensemble methods such as Random Forest (RF) [25]. SVM is an opaque model with high accuracy. RF also generates a huge number of rules depending on the number of trees involved to construct the forest, while the accuracy is typically higher in compare to one tree. The second tradeoff is between number of rules and number of uncovered samples. This may happens for the methods that stop learning if a significant part of training examples has been covered [113]. Higher number of rules reduces the uncovered samples while it damages the comprehensibility.

3.2. Categorization of RE methods

There are different rule extraction techniques that can be categorized based on several criteria such as scope of use, dependency type on the underlying model, and format of the extracted rules [111], [112]. Some of the RE algorithms are used for classification [114], [115], [109] or regression [116], [117]. The majority are devoted for one of those while there are few methods that support both such as G-REX [118].

One way to obtain a transparent model is to induce rules directly from the training set. The sequential covering algorithm falls in this category and is a base method for many other algorithms. The general approach works in this way: First one rule is extracted and the samples covered by this rule are removed. Afterward, this process is repeated on the remained samples and it continues until a condition is met. It is obvious that extracting one rule is the key element of the sequential covering algorithm and different methods uses various techniques for this purpose. The stopping condition can be covering all the samples or covering significant numbers of them in a dataset. For example, CN2 [119] induces an ordered list of rules, which uses entropy as its evaluation method and consists of running a beam search to find a good rule, removing the samples covered by that rule and then a control algorithm for repeating the search. Ripper [120] uses a standard separate-and-conquer algorithm and builds a rule set greedily by adding rules to an initially empty rule set repeatedly until all positive samples are covered. After finding a rule, all samples covered by that rule are removed; the rule is grown and then pruned to minimize error of the entire rule set. A combination of cross-validation and minimum-description length techniques is used to prevent over fitting. Minerva is also another example of this category [113].

Another option to obtain a transparent model is to take advantage of the good performance of the existing opaque models such as SVMs, RF, or neural networks and generate rules from them. There are two different rule extraction methods based on an opaque model: decompositional and pedagogical [112]. Decompositional methods extract rules at the level of individual units of the prediction model such as neurons in neural networks, and therefore rely on the model's architecture. In contrast, in pedagogical approaches, the architecture of the predictive model does not matter and it is only used to produce predictions. In other words, the predictive model is used as an oracle. Obviously for this category, there should be one intermediate model such as a decision tree or a heuristic method, which uses those predictions in order to extract the rules. RE algorithms can be either independent or dependent of the underlying model. The independent REs include RE methods which are not designed for a specific opaque model such as SVM, neural network, or ensemble methods and can be applied to different underlying model. However, they need prediction results generated by their opaque model in order to infer the rules usually by solving an optimization problem. For instance, Jiang et al [110] used simulated annealing to find the optimal box in patient rule induction method presented in [121] to search interpretable rules for disease mutations. Johansson et al. [122] used genetic programming to maximize fidelity on the class probability estimation level. In fact, they tried to minimize the difference in class probability estimation between the extracted rules and the opaque model using generalized Brier score function [123]. They extracted rules from random forests and bagged NNs, two opaque ensemble models. On the other hand, dependent RE algorithms use the inner characteristics and architecture of the black box model to generate the rules for a specific opaque model such as methods relying on neural network [124], [125] and support vector machine [126], [127].

There are also methods based on decision trees that are not RE method per se; however rules are generated as part of their learning process. For example, C4.5 [128] is a widely used algorithm in prediction. It is a greedy technique such that, at each step, the most discriminating feature is determined, and a node is split based on this feature. Each node specifies a decision on a single feature, which branches to its possible outcomes of that decision. Each leaf specifies a rule, which can classify a data sample if it matches to all the tests of the internal nodes from the root toward the leaf. C4.5 is not a RE method per se, but it can be used for this matter by extracting the rules correspond to all leaves. PART algorithm [129] is another example which is a combination of C4.5 and RIPPER, a partial decision tree is generated repeatedly. Each time the best leaf (i.e., with largest coverage) is converted to a rule. Then all the samples covered by the rule are

removed and this process is repeated until there are no samples left to cover. The collection of all the rules extracted is the final rule set.

3.3. Motivations for Rule extraction from Decision Tree Ensembles (DTEs)

Rule extraction from NNs and SVMs are widespread in the literature due to their high accuracy. One of the major drawbacks of rule extraction methods from NNs and SVMs is the rule format, which is usually not comprehensible to humans. This problem has been the main obstacles for their practical application [109], [130].

However, DTEs are one of the most important prediction methods as they demonstrate high prediction accuracy such that for some data sets they overcome other prediction methods such as NNs and SVMs [131], [132]. They are very convenient and fast to be trained and easy to implement. Moreover, they can be easily implemented in parallel for big data [133]. Another advantage is that estimating the out-of-bag error often eliminates the need for cross-validation. They are robust to noise and can handle imbalanced data sets [25], [134]. More importantly, they generate a multitude of propositional if-then rules, which is the most widespread rule type in RE domain. Therefore, they have a very high potential to provide clear explanations and interpretations of their underlying model. They can improve the accuracy and the performance due to use of an ensemble of decision trees [135]. The rules are generated as part of the learning process and there is no need to extract the rules as in the other methods such as neural network or SVM based RE. Therefore, they deserve to be considered as one of the main opaque model for rule extraction.

There are different methods to construct DTEs. Here, we briefly explain the main methods. Bagging [136] is an ensemble method that creates different classifiers by training each of them on a random redistribution of the training set. Each classifier's training set is generated by randomly drawing examples with replacement from the original training set. As a result, many of the original examples may be replicated in the resulting training set. Each classifier in the ensemble is generated with a different random sampling of the training set. Breiman [136] showed that Bagging is effective on "unstable" classifier such as decision tree and neural networks where small changes in the training set result in large changes in predictions. Boosting [137], [138] methods focus on producing a series of classifiers. The training set used for each member of the series is selected based on the performance of the former classifier(s) in the series. In Boosting, examples which are incorrectly predicted by previous classifiers in the series are chosen with

higher probability than the correctly predicted examples. Therefore, Boosting attempts to construct new classifiers in favor of the examples that the current ensemble's performance is poor for them. RF [25] is one type of bagging method, which adds an additional layer of randomness to bagging. A different tree learning algorithm is used such that each node is split using the best feature among a subset of features (m) randomly chosen at that node. In decision tree bagging, m is equal to the total number of features (n), but in RF, m is usually equal to $0.5\sqrt{n}$, $\sqrt{n}$, or $2\sqrt{n}$. Recently, a research group in Microsoft proposed decision jungle [139]. They have proposed an ensemble of rooted decision directed acyclic graphs (DAGs) to build a compact and powerful classifier. A DAG allows more than one path from the root to each leaf, unlike the usual decision trees. During training, node splitting and node merging are driven by minimizing the weighted sum of entropies at the leaves. The experimental results on different datasets demonstrated that, compared to conventional decision forests and their variants, the proposed method requires dramatically less memory in addition to improving the generalization capacity of the model [139].

For all DTEs, there is an aggregation mechanism to provide the final result of the model. Most popular techniques to merge the results of different DTs are simple voting or weighted voting in classification problems and for regression problems the average is used instead.

Bagging DTE has better performance than one DT most of the time, but it has often lower performance compared to boosting. Boosting is also sometimes less accurate than one DT and it has overfitting problems for noisy data that degrade the performance. RFs are more robust than boosting to noise and overfitting. It is faster than bagging and boosting and its performance is as good as boosting and sometimes better. As a results, RF has been widely used in the literature recently [140].

3.4. Rule extraction from ensemble of decision trees

There are two broad categories in the literature that focused on constructing comprehensible ensembles of decision trees. The first group approach is to reduce the number of trees in the DTE while the second group focuses on the rules generated during DTE construction. We call the former tree-based and the latter rule-based methods. In addition to these two broad categories, there are also other methods to extract rules from DTEs. Similar to rule extraction from SVMs and NNs, DTEs also can be used as an oracle for rule extraction purposes. In this case, DTEs are used as a black box to generate the target values for the input data. This approach can remove noise and build cleaner data set. Then, the obtained data set is used by the other method such as J48 or Ripper algorithm to generate rules [141]. The last approach is to build a new DT that

mimic the DTE model. A pruning stage is an optional step that can improve the quality of extracted rules in terms of performance and comprehensibility.

3.4.1. DTE Rule Extraction Formalization

Let $L = (x_i, y_i), \quad i = 1,2,...,N$ be a collection of N labelled instances such that $x_i \in \chi$ a vector of features and $x_i = (x_{i1}, x_{i2}, ..., x_{iM})$, where M is number of features and $y_i \in \Phi$, $\Phi = \{1, 2, ..., C\}$ a discrete class label. Consider a learning algorithm that constructs a decision tree, $h : \chi \to \Phi$, from a given training set. The learning algorithm creates a tree by recursively splitting data into subsets using one of the features which maximizes Gini impurity or information gain, two commonly used data splitting criteria. In a DTE, a collection of classifiers (*H*) is built by bootstrap sampling. The final classification is obtained by combining the weighted outputs of all DTs. Therefore, the ensemble classifier is defined as:

$$f : x \to \sum_{h \in H} w_h . h(x) \,|\, w_h \geq 0, \sum_h w_h = 1 \qquad (3-1)$$

An instance is classified according to:

$$\arg\max_{\Phi} \left(\sum_{t=1}^{T} w_t I(h_t(x) = y) \right) : y \in \Phi \qquad (3-2)$$

where T is the number of DTs and $I(true) = 1; I(false) = 0$.

With a different perspective, DTE can be seen as a rule-based ensemble and can be defined as follow:

$$f : x \to \sum_{r \in R} w_r . r(x) \,|\, w_r \geq 0, \sum_r w_r = 1 \qquad (3-3)$$

Where $R = \{r_t \,|\, t = 1, 2, ...N_r\}$; $N_r = \sum_{h \in H} |Nodes(h)|$

Nodes is a function that returns the nodes of a given DT. In a special case, Nodes function only return the DT leaves.

$$r_t(x_i) = \prod_{j=1}^{n} I(x_{ij} \in s_j) \qquad (3-4)$$

n is the number of features in r_t. S_j is a subset of all possible values for feature j and $I(.)$ is an indicator of the truth of its argument. $r_t(x_i) \in \{0,1\}$, when all the conditions are matched for x_i, it is one, otherwise it is zero.

Let $P_{N\times r}$ be the matrix (N: # of input instances; $r \subset R$) indicating whether an input instance falls into a given leaf or, in the other word, $P_{N\times r}$ specifies if the input instance is matched with the corresponding rule of a given leaf.

$$P_{ij} = \begin{cases} 1 & match(x_i, r_j) = 1 \\ 0 & otherwise \end{cases}$$

Therefore, rule extraction of DTE can be considered as an optimization problem to find $r \subset R$ such that: $\arg\min_{w} \|Y - Pw\|^2$, $w \in \{0,1\}^r$, which is an NP-hard problem [142], [143].

3.4.2. Tree-based methods

One way to have a comprehensible DTE is to reduce the number of decision trees, although these methods are mostly ensemble pruning, not RE methods per se. The general idea is to increase DTs diversity in the DTE such that the DTs in the ensemble occupy different points in the hypothesis space and as a result increase the generalization ability of the obtained sub-model. If every DT behaves similarly to the other DTs in DTE, little gain is achieved by combining their predictions. Therefore, various diversity measures have been investigated for DTE construction [144].

One example of this approach, studied by Latinne et al. [145], attempted to reduce the number of trees in RF using the McNemar test [146] of significance on the prediction outputs of the trees. McNemar is a non-parametric test which is preferred to the parametric tests, such as t-test, as there is no need to make any assumption. In addition, it has a low type I error, which is the probability of detecting a difference incorrectly when there is no difference.

The procedure is as follows: C_m and C_n are two subsets of DTs selected from ensemble models, where n>m, such that either selected DTs are completely independent or some of them are common between two sets. Comparing these two sets with McNemar and obtaining d(m,n) leads to the following scenarios (PF is the performance function):

if d(m,n) = 1 then $PF(C_n) > PF(C_m)$; Continue the procedure (selecting DTs from ensemble) with a higher number of classifiers than n.

d(m,n) = 0 => $PF(C_m) \approx PF(C_n)$; keep minimal # of DTs = m and stop.

They used (m,n= 1..200) and by using a grid search, they find the minimum m such that $PF(C_m) \approx PF(C_n)$. Their results show that with lower number of DTs, it is possible to reach to the DTEs performance level.

Another tree-based methods was conducted by Zhang et al. [147] to search for the smallest RF. Similarly, they seek out a sub-forest that can achieve the accuracy of a large RF. They used three measures, i.e., one accuracy measure (they called it "by prediction") and two similarity measures between trees in RF ("by similarity" and "by restricted similarity") in order to determine the importance of trees in terms of their predictive power. In the "by prediction" method a DT is removed if its removal has the minimal impact on the overall performance of DTE. They used a backward removal procedure such that the DT which minimizes PF (DTE) – PF (DTE-DT) is subject to removal, where PF is performance function and DTE-DT is DTE with removed DT.
In the "by similarity" method, the idea is to remove a DT which is similar to other trees in the forest. They defined similarity as the similarity between DTs predicted outcomes. For each DT subject to removal, the average similarity of the DT to other DTs in the ensemble is computed and a DT with maximum average similarity is removed. In the "by restricted similarity", an initial weight (equal to 1) is assigned to all DTs in the DTE. Afterward, pair-wise similarity is computed between every pair of DTs. The pair, DT1 and DT2, the most similar is selected and then the average similarity of both DTs in this pair is calculated as discussed earlier. The one of these two DTs with higher average similarity is subject to removal. Finally, the other DTs weights are updated proportional to the similarity to the removed DT. The DT which is the most similar to the removed DT receives the higher weight update. Unfortunately, they did not mention what the weights are used for and this part has not been described well which makes it hard to evaluate the method completely. The experimental results demonstrate that such a sub-forest with performance as good as a large forest usually exists. The "By prediction" method was better as it reduced the size of the DTE to a manageable level while maintaining the performance. They argued that by reducing the size of a RF, it is no longer a black box. However, it is still far from a comprehensible model to be easy to understand.

Similarly, Simon et al. [148] applied three methods to construct a smaller DTE. They applied sequential forward selection (SFS), sequential backward selection (SBS), and sequential random selection (SRS). At each iteration of the SFS method, each remaining DT is added to the current subset and the one that leads to the highest performance in DTE is retained. Likewise, in the SBS method, each DT of the current subset is removed, and the one for which the remaining ensemble exhibits the best accuracy is discarded. Finally, in SRS method, DTs are removed randomly without considering any criteria. Applying the proposed methods, they observed that a DTE with a smaller number of DTs can be found. However, when they applied the three above mentioned

approaches, they maximized the performance of the obtained sub-DTE on the test set, which implies some bias in the results.

Gashler et al. [149] increased the diversity in DTE by combining different DT algorithms. They used a combination of entropy-reducing DTs and mean margins DTs. The first method builds axis aligned decision boundaries while the second one constructs oblique decision boundaries. Their results showed that a DTE combining 100 of their proposed models can reach an accuracy level equivalent to that of a bagging with 1000 DTs.

Martı´nez-Munoz et al [143] proposed to avoid unspecified order aggregation of classifiers (using voting and averaging for classification and regression respectively) in the ensemble and instead they used six different metrics to specify the order in which CART trees [150] are aggregated in a bagging ensemble. The idea is that the classifiers that are expected to perform better are aggregated first. Appropriate ordering of the aggregation obtains the minimum generalization error at intermediate numbers of classifiers (about 20 classifiers) and it can outperform the whole ensemble. Oshiro et al., [140] also confirmed the findings of the above mentioned studies. They used different sizes for RF over a large number of data sets and observed that a large number of DT in RF sometimes only increases the computational cost and has no significant gain.
Yang et al. [151] computed four different metrics based on the margin distribution of the RF model to evaluate the generalization ability of sub-DTEs and the importance of the DTs in the ensemble. DTs are ranked based on the margin metrics and then the least important trees are removed one by one. The margin is defined to be the difference between the numbers of correct votes and error votes in the ensemble. They believe that similarity based pruning cannot guarantee a good generalization ability of the ensemble classifier. This fact was also observed by Zhang et al. [147].

Another approach for ensemble pruning is orientation ordering [142]. Orientation ordering is a signature vector of a classifier h_t; i.e., an N-dimensional vector (N is the number of samples in the training set) with elements equal to +1 if $h_t(x_i) = y_i$ and -1 if $h_t(x_i) <> y_i$. The average signature vector of all classifiers in an ensemble is called the ensemble signature vector or reference vector. Orientation ordering ranks the classifiers by increasing value of the angle between their signature vector and the reference vector. This ordering gives preference to classifiers that correctly classify those examples that are incorrectly classified by the full ensemble. They reduced the generalization error of a bagging ensemble consisting 200 DTs with only 30 to 60 DTs for different data sets.

All these methods are ensemble pruning and their aim is to build a smaller ensemble with performance as good as the ensemble and usually they result in different number of trees for various data sets. However, when a method prunes a DTE and finally keeps 20 DTs for example, it is still a large number of rules. Therefore, they are not really RE methods from DTE. However, these methods can be applied as the first phase of RE in DTEs. It means that first the size of the DTE can be reduced to a point that it keeps the performance of the initial DTE and then a rule extraction method can be applied for rule extraction.

3.4.3. Rule-based methods

Other methods with different approaches were proposed to select an optimal set of rules generated by RF.

Rule ensembles (RuleFit), a predictive learning algorithm, was proposed by Friedman and Popescu [152]. They built an ensemble model where the base learners are prediction rules in form of prepositional rules that are obtained from CART trees [150]. A large number of CART trees are grown on randomly drawn subsets of the data. When a tree is grown, a rule is obtained from every node of the tree. The main idea is to build a linear function F(X), consisting of rules and features such that it approximates the whole DTE accurately. The rules are functions of the features, taking a value of 1 when the rule applies, and a value of 0 otherwise (see relation 3-4). The trees are grown until a pre-specified number of rules have been generated in the initial ensemble. In addition to the rules, all the features are also considered, to allow for estimation of linear functions. The final model is formed by applying the regularized regression of the response variable (outcome variable) on all prediction rules and features. Whereas with ordinary least squares (OLS) regression the coefficients of prediction functions are estimated by minimizing the residual sum of squares, with penalized regression, an additional penalty is placed on the coefficient. The RuleFit algorithm uses the lasso penalty [153] by default. More formally, they built an ensemble predictive model F(X) as follow:

$$F(X) = \hat{a}_0 + \sum_{k=1}^{K} \hat{a}_k \, r_k(X) + \sum_{j=1}^{m} \hat{b}_j \, l_j(x_j)$$

$$\{\hat{a}_0\}_0^K = \arg\min_{\{a_k\}_0^K} \sum_{i=1}^{N} L\left(y_i, a_0 + \sum_{k=1}^{K} a_k r_k(x_i) + \sum_{j=1}^{m} b_j l_j(x_{ij}) \right) + \lambda \left(\sum_{k=1}^{K} |a_k| + \sum_{j=1}^{m} |b_j| \right) \quad (3\text{-}5)$$

$X=(x_1,x_2,...,x_N)$ is a vector consisting of samples, $\{r_k(X)\}_1^K$ is the set of K rules extracted from the trees in the ensemble. $L(y,\hat{y})$ represents loss or cost function for predicting $\hat{y}$ while the

correct value is y. Variable m is the number of features that are used to build the regression model, N is the number of samples. The first term in Equation 3-5 computes the prediction loss on the training samples and the second term, which is a regularization term or "lasso" penalty, penalizes large values of the coefficients $\{a_k\}_1^K$ and $\{b_j\}_1^m$. $\lambda \geq 0$ is the regularization term. $l_j(x_{ij})$ is a so-called "Winsorized" version of the j-th feature, which is used for robustness against the outliers and is defined as below:

$$l_j(x_j) = \min(\delta_j^+, \max(\delta_j^-, x_j)) \tag{3-6}$$

Where δ_j^+ and δ_j^- are the β and $1-\beta$ quintiles ($\beta = 0.025$) of the data distribution $\{x_{ij}\}_{i=1}^N$ for each x_j .

The good performance of the RuleFit is due to linear combination of the rules and features. Although the rules are simple to interpret, combining the features weakens the comprehensibility.

Node harvest (NH) [154] is another rule extraction method. An initial set of rules is generated randomly (default is 1000). Rules can be selected from a fitted tree ensemble such as RF. The rules that satisfy the maximal interaction order (number of features in the rule, with default value equals to 3) and minimal rule size (number of samples that match to the rule with default value equals to 5) constraints are added to the initial rule set, provided that they are not already selected. NH's aim is to find suitable weights on rules by minimizing the following empirical loss function under some constraints, which is a quadratic program with linear inequality constraints (see [154] for detailed solution).

$$\hat{w} = \arg\min_w \|Y - Mw\|^2 \tag{3-7}$$

$$M_{ig} = \begin{cases} \mu_g, & \text{if } X_i . \in Q_g \\ 0, & Otherwise \end{cases} \tag{3-8}$$

$$\mu_g = \frac{\sum_{i=1}^n 1\{X_i . \in Q_g\} Y_i}{\sum_{i=1}^n 1\{X_i . \in Q_g\}} \tag{3-9}$$

$$w \in \{0,1\}^q$$

Where $. \in$ means a sample is matched by a rule, X shows the samples and Y is the target value or response. N is the number of samples, q is the number of rules initially selected, M is N×q matrix, and μ_g is the mean of all samples that are matched with rule Q_g.

If a new sample is covered by a unique rule, its prediction would be the mean response of all samples within this rule. If a new sample is covered by several rules, its prediction is the weighted average of the mean responses of all these rules. The weight of each selected rule is computed using quadratic programming with linear inequality constraints. Only few rules will have non-zero weight. The main important feature of NH is that the generated rules are particularly short. The reason is that NH considers not only the leaves in RF but also intermediate nodes in the trees as candidate rules for the initial rule set provided they conform with the pre-defined constraints. The prediction accuracy is comparable with RF, however its performance is better for smaller signal-to-noise ratio. NH can be applied for regression and classification problems with multivariant features and it can also handle missing values.

Liu et al. [155], [156] used RF as an ensemble of rules and proposed a joint rule extraction and feature selection method (CRF). They viewed RF as a collection of decision rules. They used a binary encoding mapping method such that for each sample x_i the corresponding encoded vector is $X=[X_1, ..., X_q]^T$,where q is the number of rules in the RF. The value for a given X_j is equal to 1 if x_i is matched with the j-th rule in RF, otherwise it is 0. Therefore, matrix X shows the active and inactive rules for every sample in the training set. Then X is considered as the training set and the aim is to find weight vectors such that

$$y = \arg \max_{k \in \{1,...,K\}} (w_k^T X + b_k) \tag{3-10}$$

$$X = \{(X_1, y_1), (X_2, y_2), ..., (X_N, y_N)\}$$
$$y_i \in \{1, 2, ...K\}$$

where K is the number of classes in the data set. Then rule extraction is formulated using 1-norm regularization:

$$\min_{w_k, \xi_{ik}} (\lambda \sum_{k=1}^{K} ||w_k||_1 + \sum_{i=1}^{N} \sum_{k \neq y_i} \xi_{ik}) \tag{3-11}$$
$$\text{s.t. } (w_{y_i} - w_k)^T X_i + b_{y_i} - b_k + \xi_{ik} \geq 1$$
$$\xi_i \geq 0, \ i = 1, ..., N$$

The first term of the equation 3-11 controls the number of rules in the final rule set and the second term specifies the error term, which is the number of misclassified samples. Therefore, the tradeoff between the final number of rules and error is determined by λ. This parameter is

selected by cross-validation on the training set. They employed linear programming to solve the above optimization problem and reached to 96% accuracy of RF by selecting only 1% of the rules generated by RF.

3.4.4. Other methods

There are also some other methods to increase comprehensibility of an ensemble or RF by compacting them into one decision tree. For example, a single decision tree was used to approximate bagging of decision trees. In this method, class distributions were estimated from the ensemble in order to determine the tests to be used in the new tree. They argued that a decision tree is able to represent any function as is the ensemble. Instead of computing the information gain from the original data set to determine the best test for each node in the DT, they used an ensemble to approximate it. Therefore, class distribution predicted by ensemble was used for that purpose. At the end, if all training examples end up in a node and are classified identical to the ensemble, no further splitting will be performed and the node will become a leaf. The experimental results showed that the proposed method can approximate an ensemble of 25 DT with one tree with size 2.5 times larger than the tree generated by J48 [157]. They assumed the tree obtained from J48 is comprehensible; however, the size of the tree in J48 can be very large depending on the pruning rate, which is not reported in that study. Although the proposed method seems interesting, we could not find any comparison of the proposed method with the other similar methods. In addition, the implemented tool is not available and for more evaluation it needs to be re-implemented.

A similar method was employed to approximate a RF with just one decision tree [141]. The aim was to generate a weaker but transparent model using combinations of regular training data and test data initially labeled by the RF with 100 trees, which is called oracle coaching. They have two different data sets: one is a training data set, which is the original data set (E) and the second one is the oracle data (X); which is the test set with corresponding predictions from the RF as target values. To obtain the oracle data set, they train the RF with (E). Afterward, they used J48 to extract rules from different combination of the data sets i.e., the original data set (E), oracle data (X), and training data and oracle data (IX). 10x10-fold cross-validation was applied for purpose of evaluation. They obtained an equal accuracy level from both the RF and extracted rules using IX data set that shows the oracle coaching method effectively improved the accuracy. The obtained AUC with IX is 6% less than that of RF in average and the test fidelity is also about 97%, which shows the percentage of the test set that are predicted in both identically. The result for X was worst than IX in terms of accuracy, AUC, and fidelity. However, as IX is using artifact

test set built by RF the results are not significant, although they claimed the approximation of random forest using only one decision tree with a good precision, on the specific test data. They also did not provide evaluation of the size of the tree generated by J48 which does not allow evaluating their method in term of comprehensibility.

Chapter 4

4. Speciation Prediction

4.1. Introduction

A species is a group of individuals that are capable to exchange genes within themselves, but are reproductively isolated from other such groups. Consequently, there is no direct gene flow between two species [158]. Speciation is the division of one single species into two or more genetically distinct ones. It extends through time and leads to a hierarchal tree of historical relationship between species. It consists of two steps [159]. First, a new population should be established. This new population can exist in the same habitat or can be completely separated from the main population, depending on the type of speciation mechanism. For example, in sympatric speciation, a new population emerges from a single local population while in allopatric speciation a physical barrier separates a sub-population from the initial population. Second, based on different factors such as genetic divergence, different habitats, and physical barrier a reproductive isolation should occur, that reduces or prevents gene flow between organisms of different species. Therefore, the geographical and spatial distribution of individuals in one species is a leading phenomenon for speciation [159], [160], [161]. For example, in [162], it has been shown that there is a linear relationship between genetic and geographic distance. It means that an increase in physical distance between individuals leads to increasing their genetic distance. If the genetic distance between individuals of the same population is too high, reproductive isolation will occur and leads to speciation. Consequently, increasing the physical distance between individuals increases the probability of speciation.

However, considering spatial distribution metrics alone is not enough to study speciation. Because it is a continuous, ongoing process, the current spatial distribution of a species is not necessarily a reliable index of the species' historical distribution during its life time. Losos et al. [163] mentioned three pieces of evidence showing that the present spatial distribution of a species is greatly different from the one at its creation time. Therefore, observing species during its whole life time is also important to understand and eventually predict speciation.

Predicting speciation and discovering important factors involved, would bring new insights in evolutionary and conservation biology. However, observing and studying species in nature to extract species information is a difficult and time consuming process. In addition, speciation needs a long time to appear and most of the time is not possible to observe it in nature.

Individual-based modeling is a possible theoretical approach to overcome these limitations. The interest of this approach is that it allows complex interactions between multiple agents to shape the whole behavior of the system making it a powerful tool to study how individuals' actions influence the global ecosystem. Therefore, we applied machine learning techniques on the data generated by EcoSim to evaluate if selected features can predict splitting of species. If we can predict speciation, it means that they have impact on species splitting.

In the first experiment, we wanted to investigate how spatial and spatiotemporal patterns influence speciation. However, speciation can be affected by several factors. Based on Darwinian theory, natural selection is the main reason for speciation and emerging genetics studies strengthened this theory by explaining variation in a population via genetic operations [14]. Pre- and post-zygotic barriers, which lead to reproductive isolation, are also very important in speciation. Geographically isolated populations tend to form new species as well [15], [16]. Moreover, sexual selection plays an important role in speciation [17]. In the second experiment, we used not only spatial distribution information but also demographic, genetic, and environmental features to predict speciation.

4.2. Speciation Prediction using Spatial and Spatiotemporal Features

4.2.1. Preparing Data sets

EcoSim generates huge amount of information for all the objects in the simulation, such as world (the landscape), species, individuals, and food which is stored separately. However, for this study, we only extract spatial distribution and spatiotemporal information for every species.

4.2.1.1. Spatial Distribution Information

In EcoSim, we have access to all the recorded information for each individual. Therefore, it is possible to specify the location of each individual at any time step in a 3-dimensional vector with two spatial and one temporal dimension. The world is a torus, which can be easily implemented by a rectangular array by allowing individuals to pass across one boundary and enter the opposite boundary. Based on the circular condition of the world applying traditional statistics is not possible, thus we used circular statistics for computing the species spatial center [164]. Therefore, we calculated spatial standard deviation, and the sum and the average Euclidian distance of all the individuals to the species center.

4.2.1.2. Spatiotemporal Metrics

As mentioned before, considering spatial distribution metrics is not enough to study speciation, because it is a continuous, ongoing process and current spatial distribution information of species is just a snapshot of its lifetime. Therefore we considered several spatiotemporal metrics described in [165] as well.

These metrics are used to characterize the complex spatiotemporal dynamics of ecological mosaics or categorical maps. This characterization is based on analysis of space-time cubes of data with two spatial dimensions *x*, *y* and time dimension *t*, which we call the 3D world. This cube includes successive spatial information of the environment sampled at uniform time intervals. Each spatial image in 3D world is a grid of cells. By adding temporal dimension, each spatial pixel becomes a 3-dimensional voxel having two spatial and a temporal dimension. Persistent entities, like prey in our simulation, occupied 3-dimensional forms consisting of several voxels that are adjacent in space-time, which are called a blob. In a 3D world, there might be different kinds of blob types. For example, in EcoSim, each blob type corresponds to one unique species. Moreover, each voxel in the 3D world may belong to different blob types because each cell in EcoSim may contain multiple individuals; therefore, it is likely that a voxel contains individuals from different species.

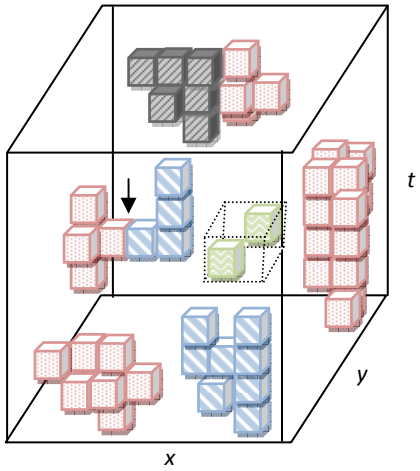


Figure 4-1. A Simple example of four blob types in the 3D world. Arrow shows 2 adjacent voxels with one shared face. The dashed cube is the bounding box of the green (wavy format) blob type.

In addition, each blob type is usually composed of multiple separated blobs in 3D world. For example, one species blob type may consist of four blobs based on the position of its individuals in the 3D world similar to what is shown in the dotted pattern blob type in Figure 4-1.

There are two 3D metrics categories for analyzing blobs: composition and configuration metrics. Volume, surface area, shape complexity, and fractal dimension are examples of composition metrics. A blob volume is the number of voxels it occupies. Surface area is the number of voxels in a blob with faces not shared by adjacent voxels of the same blob type. For calculating adjacency, we used 6-voxel vonNeuman neighbors by considering a voxel as an adjacent if it shares a face with the current voxel (Figure 4-1).

Shape complexity is a ratio between blob volume and volume of its bounding box. For example, assume the dotted line cube volume is four (Figure 4-1). Then, the shape complexity of the wavy pattern blob type would be 0.5. Fractal dimension quantitatively describes how one object occupies its volume. We used count boxing method to calculate fractal dimension for each species. For this purpose, we successively covered the 3D world with a 3-dimensional filling box and recorded the number of boxes (N(r)) required to cover the whole cube provided that containing at least one voxel related to the given species. Afterward, we repeated this procedure with different box size *r*. For example, we used *r*=2, 5, 10, 20 and 25, where the size of 3D world is 1000×1000×50. A graph of ln(N(r)) versus ln(1/r) is generated. The slope of the linear regression line gives the fractal dimension.

Moreover, we calculated some other composition metrics. Space-time density is the ratio of blob type volume and the 3D world volume. Population density is the number of individuals per voxel. Blob number is the number of isolated blobs in a specific blob type. For example, the blob number for the dotted blob type is 4 (Figure 4-1). Blob volume average and standard deviation are average and standard deviation of isolated blob volumes for a specific blob type.

Contagion and STC (spatiotemporal complexity) are two configuration metrics. Contagion was calculated based on Equation 4-1, which measures dispersion or clumpiness of a blob type. This metric is based on voxel adjacencies and probability of finding a voxel of one blob type next to voxels of other blob types. A lower value of contagion shows many small blobs and higher value indicates few large blobs.

$$RC = 1 - \frac{EE}{EE_{\max}} \tag{4-1}$$

where, RC is contagion and $EE_{\max} = b \times \ln(b)$ and b is the number of blob types. Also

$$EE = -\sum_{i=1}^{b}\sum_{j=1}^{b} p_{ij} \ln(p_{ij}) \quad (4\text{-}2)$$

$$p_{ij} = \frac{n_{ij}}{n_i}$$

where n_{ij} is number of adjacencies between voxels of blob type *j* and voxels of blob type *i* and

$$n_i = \sum_{i=1}^{b} n_{ij} \quad (4\text{-}3)$$

STC is a feature to describe how one blob type occupies the three dimensional space. This metric can be applied only to two blob types per space-time cube. Because we may have multiple blob types based on the number of species in the current 3D world, we calculated STC for each blob type separately by considering all the other blob types as background. STC was calculated by counting number of voxels occupied by blob type *i* in a three dimensional windows of dimension $n \times n \times n$ where *n* is much smaller than the 3D world size. For our case, *n* is 5 because it is much smaller than our 3D world dimensions and all the dimensions of the 3D world are divisible by 5 so that it can be fitted by this window size completely. The window moves successively in the space-time cube and measures the different occupation levels from 0 to n^3. Then STC was calculated by a relation as follow:

$$STC = \frac{\sum_{i=0}^{n^3} p_k \ln(p_k)}{\ln(n^3 + 1)} \quad (4\text{-}4)$$

In Equation 4-4, STC is spatiotemporal complexity and 0<STC<1. p_k is the relative frequency of occupation levels. STC is more effective than contagion in describing the complexity of a spatiotemporal pattern and is able to differentiate various patterns such as uniform blob shapes (for example a column), random and complex patterns. STC value is lower for uniform or ordered blob shapes and is higher for complex shapes [165].

In total, we computed three spatial and eleven spatiotemporal metrics. These metrics were computed for every species in five distinct runs of EcoSim for 10000 time steps of the simulation. The length of the time dimension in the 3D world for calculating spatiotemporal metrics was assumed 50. By increasing this length we would have more precise information about species spatial history but it also increases the computational complexity of the defined metrics.

Therefore, the size of the window to calculate spatiotemporal metrics was assumed 1000×1000×50.

4.2.2. Training Algorithm and Evaluation Criteria

For preparing the data sets, we applied the following procedure to all five runs data sets:

1) In each time step of the simulation, we calculated the spatial information for each species.

2) We calculated 3D spatiotemporal metrics by considering the information of the fifty previous time steps for each species to construct the blob types and to compute the configuration and composition metrics.

3) After merging the results of the steps 1 and 2, we constructed one training and one test set. There are two classes in this dataset, positive and negative, which specify if a speciation event will happen in next 100 times.

4) Steps 1 to 4 are repeated for other 4 runs.

These steps result in five training sets and five test sets from five different runs. The main problem in all these datasets is that about 90 percent of samples belong to the negative class and only about 10 percent of them are in the positive class. It means that only 10 percent of species split in the next 100 time steps. As a result, the dataset is strongly imbalanced. There are two main approaches to address unbalanced training sets [166]. One is to assign distinct costs to training examples. The second method is to re-sample, either by under-sampling the major class or over-sampling the minor class. We examined different algorithms and finally, we found out the smote algorithm [167] surpasses other algorithms for our data sets. For each sample of the minority class, smote generates synthetic samples by selecting some of the nearest neighbors and generates new samples along the line segments connecting k minority class nearest neighbors. For example, if the smote percentage is 200%, two nearest neighbors are selected and one sample is generated in the direction of each by multiplying the difference between a given sample vector and its nearest neighbor by a number between 0 and 1, and adding it to the sample vector under consideration. This operation finds this new point along the line segment between two samples. We applied the smote algorithm on all training sets. However, we only used the smote algorithm for the training sets keeping the test sets with the initial unbalanced properties of the whole dataset. C4.5 [128] algorithm was employed to build decision trees for all the training sets. The interest of using such an approach is that the obtained trees can be used for speciation event prediction as well as extracting the rules which can effectively determine the most important factors in speciation according to spatial and spatiotemporal information. Afterward, we

evaluated the classifier performance using test sets. To investigate the impact of different training sets on speciation event prediction, we repeated this procedure for the other four datasets.

The performance of a machine learning algorithm is typically evaluated by overall accuracy. However, it is not applicable for an unbalance dataset where only 10 percent of species split. In this case, the training algorithm mostly is biased towards the major class (negative class) while the minor class is highly important as it shows the correct prediction of samples with a speciation event. Consequently, the overall accuracy is not a good measure to evaluate our classifiers performance. For evaluating the performance of these classifiers, we used two metrics: Recall and area under ROC curve (AUC) [168], in addition to the overall accuracy.

4.2.3. Classification Results

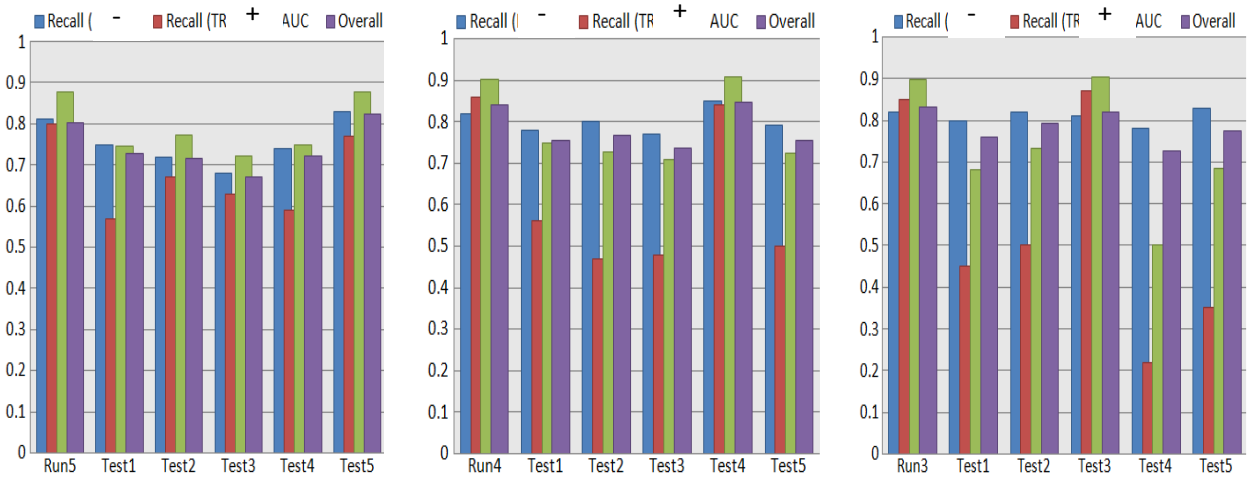
The three data sets Run1, Run2 and Run4, had about the same number of species and they led to almost the same results. To simplify results presentation, we presented only the results for the Run3, Run4 and Run5 representing situations with small, medium and large number of species respectively. Before applying the Smote algorithm on the training sets, we reached a high value for overall accuracy (above 90%) but very low recall for minor class (less than 0.3). This happens because the classifier is biased to the majority class and almost ignores the samples from the minority class.

For all the datasets, the oversampling method used by the Smote algorithm considerably improved the Recall and the AUC values especially for minor class. As expected, we observed that we always had better prediction for the test sets coming from the same run as the training set. For example, in Figure 4-2 (a), Test5 and training set Run5 are from the same run. It shows that the classifier reached a very good result for Test5 in compares to other test sets. Although the results for the test sets from the other runs (Test1, Test2, Test3 and Test4 in Figure 4-2) are not as good as Test5, it shows that the classifier has learned some general rules of EcoSim speciation event. This fact can be seen in Figure 4-2 (b) and (c).

Three different cases appear from these results:

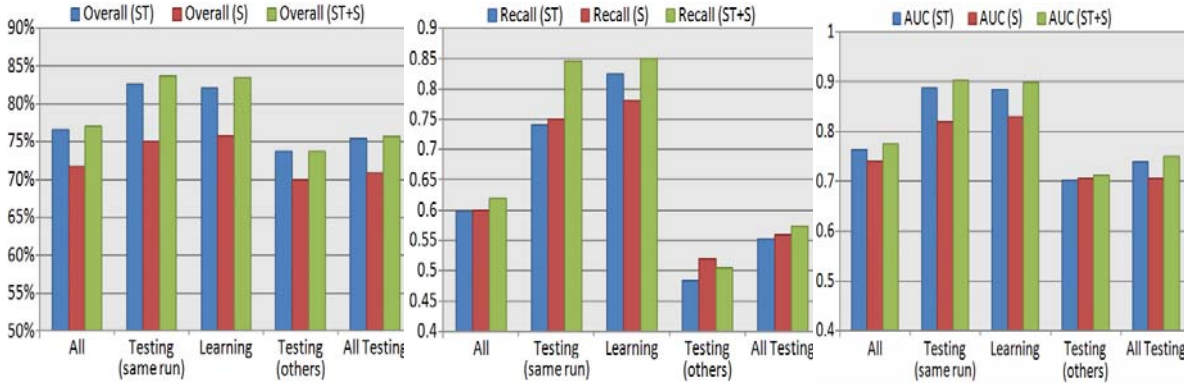
The number of species in Run5 was 438. It means that for Run5 we can expect to have more valuable information regarding speciation in comparison to other datasets such as Run3 with 115 species. It is effectively confirmed by our results; when we used Run5 as a training set we had better predictions for all the test sets as it appears clearly in Figure 4-2 (a). On the other hand, the worst result is obtained when we used Run3 (115 species) as the training set to predict the

speciation for test sets samples (Figure 4-2 (b)). We can also see that the results are much more variables than with the other training sets, confirming the lack of pertinence of the trained model.



(a) (b) (c)

Figure 4-2. Results when Run5, Run4, and Run3 are used as learning sets in (a), (b), and (c) respectively



(a) (b) (c)

Figure 4-3. Comparing overall accuracy, Recall, and AUC. All shows the average result of train and test sets, Testing (same run) is the result for the testing set from the same run, Learning is the result for train set, Testing (others) means the result of the test sets which is built from different run, All Testing is the average result for all test sets (Test sets from the same run and the other run). (a), (b) , and (c) represent results for run5, run3, and run4 respectively. ST, S, and ST+S mean the result for the dataset of only spatiotemporal metrics, only spatial information, and all the features respectively.

Run4 (with 238 species) as a training set had an intermediate performance (Figure 4-2 (c)). Therefore, we found out that if we use a run with more species to build a classifier it has better

generalization ability than a classifier that has been trained with a training set from a run with less species. It also means that some general rules about speciation exist in our system, such as having more examples of speciation in one run help to predict speciation in another run with different conditions.

4.2.4. Effect of spatial and spatiotemporal information on prediction

In order to investigate the effect of the different features on speciation prediction, we repeated this experiment two more times with different combinations of features; first with only spatial distribution information and the second with only spatiotemporal metrics. Figure 4-3 shows the results summary. We showed the average of overall accuracy, recall for positive class and area under ROC curve for all the training and test sets together (the column showed by "All label"), test sets from the same run, training sets, test sets from other runs and finally all test sets for all five runs.

These results clearly show that the best results are related to when all the features are in the training process i.e., ST+S dataset. The most important results are for Testing (others) as they show the generic prediction capacity of our models, however the results for Testing (same run) are also important as they show that some specific property of each run have been captured so that it can be useful to characterize a specific run. Even though S has only three features, it showed good capacity to learn generic rules.

Therefore, spatial distribution information of individuals in the world of EcoSim is very effective in predicting speciation. Moreover, adding spatiotemporal information to the spatial information increases the quality of the prediction. If we build the classifier based on datasets S and S+ST before oversampling; recall or TP rate is very low for the minor class (about 0.05 to 0.08) in S while that of S+ST is around 0.20 to 0.3 with approximately the same overall accuracy. It also improved AUC by approximately 15% on average. Therefore, it shows that by adding spatiotemporal metrics, the classifier is able to predict more minor class samples in presence of unbiased dataset. On the other hand, for biased datasets we observed 5% improvement for both overall accuracy and AUC for dataset S+ST in average for all runs in compare to that of S. However, if we consider the Testing (same run), it improves AUC, overall accuracy and recall for 10%, 8% and 10% respectively.

These results demonstrate that spatial information of individuals in the world has great effect in speciation event prediction and spatiotemporal metrics can improve it. We also observed this fact

in the rules extracted from the classifiers. For example, for most of the predictors, spatial standard deviation is the decision tree root showing its importance for speciation prediction. However, more in-depth analysis of the set of rules generated still need to be done to be able to explain the speciation process based on such information.

4.3. Speciation Prediction using Spatial, demography, environmental and genetic Features

4.3.1. Data Set Preparation

We used the result of 10000 time steps of three different runs of EcoSim for this experiment. In each time step, there was a variable number of species with their corresponding features. Each run is different from the others in terms of demography, environmental and genetic features due to stochastic processes in the model. The features were used for this study are as follows.

We used spatial distribution information such as spatial standard deviation, and the sum and the average Euclidian distance of all the individuals to the species center. Several features [169], [170] were also calculated to characterize the complex spatial dynamics of the world, similar to the ones calculated for the first experiment with the exception that we considered two dimensional world instead of 3D world. We also calculated the genetic diversity of a species, which measures how much diversity exists in the gene pool of the individuals of a species and corresponds to the entropy of the set of genomes. The entropy measure is commonly used as an index of diversity in ecology and increasingly used in genetics [171]. We also calculated several demographic features such as number of species, ratio of individuals in one species to the whole population (popRatio), ratio of new born individuals to the whole population (birthRatio), average population per cell (popDensity), interbreedingRatio, which is the ratio of new born individuals with parents from two different species to the whole number of new born individuals, the ratio of prey killed by the predators to the total number of individuals (killedRatio), and the ratio of dead prey because of old age and low energy to the total number of individuals (deadAgeRatio and deadEnergyRatio respectively). Moreover, several features related to individuals' actions were computed, which show the percentage of the individuals in a species choosing one action such as reproduction (reproductionAction), search for food (SearchFoodAction), and eat (EatAction). Some features related to the individuals' perception were chosen depicting the individual perception of its environment such as distance to predator (distancePred), distance to food resource (distanceFood), distance to other preys (distancePrey), distance to partner (distancePartner), distance to predator (distancePred), etc. These features give some insight about

the local environment properties of the individuals. Besides, several features related to the age and energy of individuals were calculated. For instance, the average energy and the average age of the dead individuals (deathEnergy and deathAge respectively), and the average age (age) and the average energy (energy) of individuals can be mentioned. Further, we used some features related to mating, such as average reproduction age and energy of parents at the breeding time (parentReproduceAge and parentReproduce Energy). Finally, MatingDistance which is the genetic distance between two parents' genome and also stateofBirth, which indicates the amount of energy that prey invests in the breeding process, were calculated.

We created a dataset using 49 features (see Table 5-2. List of the features used to analyze and predict species extinction. Each feature is computed at each time step per species), each sample of the dataset shows the information about one species at a given time step. Species were classified based on a label feature that specifies if one species will split in the next 100 time steps. If the species split within this time period, the label is positive, otherwise it is negative.

Massive raw data were used for this experiment, with an average of 20 (5), 27 (9), and 32 (7) species and 126000 (32000), 179000 (27000), and 197000 (26000) individuals respectively, at any given time step, for the three runs of EcoSim (the values in parenthesis are standard deviation).

4.3.2. Classification Results and Discussion

In this section, we discuss the results of our experiments and also investigate the effect of the different features we used for speciation prediction. First we employed the original training set without changing the class distribution to build the classifier. The result of the classifier for the training, test, and validation sets is shown in Table 4-1.

TP Rate for minor class was low because the classifier tends to learn the samples from the majority class and almost ignore the ones from the minority class. Because the training and test sets were built from the mixture of result of two runs, their TP rates were close, but for the validation set, it was about 12 percent less than for the training and the test sets. It shows that the generalization ability of the classifier is not good enough to classify species for other runs. The F-Measure value was also low for the validation set.

Table 4-1. Result of Speciation Prediction using Imbalanced Training Set

Data Set	TP Rate	FP Rate	F-Measure	AUC	Accuracy
Training	0.749	0.082	0.756	0.934	87.41%
Test	0.743	0.076	0.758	0.935	87.66%
Validation	0.625	0.082	0.679	0.917	83.64%

We applied the smote oversampling technique on the training set to build a balanced data set. However, the test and validation sets retained unchanged. Afterward, we built the classifier and the results indicate that TP Rate improved by 21.2%, 21.2% and 30.7% for the training, test, and validation sets respectively (Table 4-2). We observed only a 10% decrease in TN Rate, on the other hand we improved the TP Rate about 25% on average. F-Measure improvement for validation set was also about 7%.

Table 4-2. Result of Speciation Prediction using Balanced Training Set

Data Set	TP Rate	FP Rate	F-Measure	AUC	Accuracy
Training	0.908	0.168	0.875	0.932	87.00%
Test	0.901	0.154	0.736	0.933	85.83%
Validation	0.817	0.177	0.729	0.904	83.21%

Therefore, the classifier was able to predict the positive class, especially for the validation set, with higher accuracy. In this case, the classifier was more generalized, being able to classify species in a completely different run with a good accuracy. The last experiment was to use the most common features (13 out of 49 features) chosen by different feature selection algorithms such as Best Fit, Greedy Stepwise, Genetic Search, and Ranker (with InfoGain and GainRatio evaluators) in Weka [172] using the default parameter setting (Table 4-3). As it shows, we obtained an improvement in the TP Rate and F-measure, especially for validation set.

Table 4-3. Result of Speciation Prediction using Selected Features

Data Set	TP Rate	FP Rate	F-Measure	AUC	Accuracy
Training	0.911	0.184	0.870	0.924	86.36%
Test	0.916	0.191	0.705	0.925	83.20%
Validation	0.899	0.206	0.738	0.923	82.30%

Using 13 selected features including demographics, genetics, and environmental features, we obtained almost the same accuracy as when all the features are used. In addition, the complexity of the classifier was reduced so that it also decreased the risk of overfitting and made the model easier to interpret.

Removing some features did not mean they were not effective on speciation; instead they might be covered by the selected features. To investigate this coverage, we extracted the dependencies between several features in both problems by applying Bayes Net classifier. For example, genetic diversity, which seems to be an important feature, can be replaced by population ratio. This makes sense because increasing these two features makes a larger gene pool, which increases the speciation probability. Another example is in the spatial information category where patch area ratio can cover some features like SC and patch circumference.

To investigate the validity of obtained results, we extracted several rules (Table 4-4). When patch area ratio is greater than a threshold, it shows the individuals of the species are more dispersed, which increases the possibility of speciation as discussed in [4]. In speciation prediction, population ratio had a critical role. Having more individuals means a gene pool with higher variation. Therefore, one species with more individuals has higher probability for speciation.

Table 4-4. Several Samples of the Extracted Rules. *t* values are thresholds for each feature. Hit ratio is percentage of samples that match to one rule and the accuracy shows the performance of the rule on the matched samples for validation set

Condition	Result	Hit Ratio	Accuracy
patchAreaRatio<= t_a	no speciation	45%	89%
patchAreaRatio>t_a	speciation	54%	82%
indvNoRatio<= t_i	no speciation	44%	90%
indvNoRatio>t_i	speciation	56%	81%

4.4. Conclusion

In the first experiment, we analyzed the ability of spatial and spatiotemporal information about species in an artificial ecosystem for the prediction of speciation events. We used 14 features to extract this information and applying oversampling technique to build classifiers. We obtained very good results when the test set is coming from the same run as the training set. The good results for the test sets from different runs demonstrated that the classifier was able to extract general rules about speciation that exist in our system. For all datasets; S, ST, S+ST, we also

observed better performance for the classifier when the number of species increases in the training set. In other words, giving more examples of speciation events, even if they happen in the same run, leads to a more generic predictor. This indicates that some generic traits exist in our simulation that characterizes the speciation events.

In the second experiment, we computed 49 demographics, genetics, environment and spatial distribution features for the species observed in EcoSim and investigated how these features affect speciation. After adjusting the class distribution, using an oversampling technique, we obtained promising results. The results show that the calculated features are effective in prediction of speciation and can help for better understanding of speciation. Moreover, using feature selection strategies, we were able to reduce the number of features to capture more precise information involved in speciation. Finally, these techniques helped to reduce the size of the tree generated by the C4.5 algorithm, which facilitates the extraction of hypothesis for these two events for future work. We extracted several simple rules from the constructed decision tree. These rules are semantically clear and sound reasonable based on the biological evidence. This is an important result as the proposed approach has proven to have the capability of generating realistic rules when compared with real biological data.

Chapter 5

5. Extinction Prediction

5.1. Introduction

One of the most fundamental questions in population biology and conservation biology relates to species persistence and the risk of extinction where one species cannot survive because its individuals are unable to reproduce, or they simply cannot tolerate the environmental conditions. Species extinctions result from a variety of biotic and abiotic factors, such as population size [20], [173], habitat destruction and degradation, human intervention, infectious disease, reproduction rate, migration rate [174], invasive species [175], environmental variation [176], habitat fragmentation [177], habitat quality and size [178], the Allee effect [179], genetic inbreeding [180], genetic diversity [181], initial population size [182], patch size [183], age [184], and energy [185]. These factors increase the probability of extinction, and can be classified into three broad categories: demographic stochasticity, genetics and environmental factors [186], although admittedly there is overlap between these broad categories.

Random fluctuations in demographic factors such as birth rate and death rate can have dramatic effects on populations. The effect of demographic stochasticity is greater in smaller populations than in larger ones [187]. In addition, there are factors relating to the transmission of genes from one generation to the next. Genes may be lost from a small population and the gene frequencies may be modified due to drift or inbreeding [187], [180]. Diminishing genetic variation may increase extinction risk by limiting the ability to adapt to stressful environments. Lastly, environmental factors such as natural catastrophes (including fires, floods, earthquakes, and volcanoes), temperature, availability of food, competitors, predators, and diseases influence the population by changing the demographic parameters. For example, Gregory and Courchamp [188] advanced experimental evidence suggesting that predators can produce the so-called Allee effect (a reduction in population size making extinction more likely). Further, there is reason to believe that volcanic activity had a major role to play in five mass species extinctions [189].

Using mathematical modeling to study species extinction is prohibitively difficult, and consequently most results in this area are approximations at best, especially if a consideration of a mixture of relevant factors is desired [190]. Similarly, results obtained from laboratory experiments often conflict with field studies [186]. Moreover, observing and studying species in nature in order to extract species information is a highly time consuming and complicated process

given that populations exist within an interacting network of species, along with being distributed in a patchy manner over a heterogeneous space.

The overall aim of this study is to use an individual-based modeling approach to investigate a wide variety of important factors contributing to extinction, along with investigating their predictive potential using methods that circumvent the difficulties with empirical studies and mathematical modeling. To achieve this aim, we analyzed the information gathered from EcoSim followed by the integration of the extracted knowledge to verify species extinction realization in EcoSim under the three broad categories of genetic, environmental, and demographic in line with [186]. We used individual-based computer simulations that take into account species interactions (including the effects of predation) and which are relatively inexpensive to run and which take a relatively short period of time to complete. We designed an approach based on a combination of feature selection, focusing on the most informative features, and predictive model building. We evaluated the accuracy of the predictive model building to assess the quality and the generality of the models obtained, with an eye towards extinction prediction. In addition, this predictive model helped us to extract some effective prediction rules based on these filtered features. This approach increases the testability of ecological and biological mechanisms of species extinctions.

5.2. Data Preparation

We extracted our data from nine different runs of EcoSim, each run involving 10000 time steps, including all the applicable demographic, genetic, and environmental features for prey individuals. Additional details about these runs are provided in Table 5-1. The runs are different from one another in terms of demography, environmental, genetic, and internal features, although they were initiated with the same parameter sets. This variance in the simulation results originated from internal variability due to stochastic processes in the model and chaotic properties of the overall system. The 49 computed features are shown in Table 5-2 along with their definitions. Most of the features have been defined in chapter 4. Besides, we computed some action features such as explore, escape, search for food, and eat which show the percentage of the population choosing these actions. In addition, some perceptual features were chosen depicting the individual's perception of its environment such as predator distance, food distance, partner distance, etc. These features provide insight regarding local environmental characteristics. Additional features related to age and energy were calculated such as the percentage of dead individuals, the energy of dead individuals, the average age of dead individuals, and the average

age and energy of living individuals. In addition, we computed various mating features such as average reproduction age and the energy of parents at the breeding period.

Table 5-1. General information for nine different runs of EcoSim including the number of species average, average population, extinction rate and speciation rate with standard deviation in parenthesis

	Prey				Predator			
Runs	**Species Number**	**Population**	**Extinction Rate**	**Speciation Rate**	**Species Number**	**Population**	**Extinction Rate**	**Speciation Rate**
Run1	21 (5.9)	160717 (48465)	0.0085 (0.020)	0.0091 (0.022)	5 (2.4)	19339 (6793)	0.0008 (0.012)	0.0010 (0.017)
Run2	26 (8.7)	177477 (36641)	0.0096 (0.020)	0.0101 (0.022)	11 (7.4)	32052 (6815)	0.0013 (0.013)	0.0016 (0.016)
Run3	29 (9.5)	182151 (43434)	0.0079 (0.017)	0.0083 (0.019)	15 (7.8)	32970 (9779)	0.0015 (0.011)	0.0016 (0.014)
Run4	25 (9.5)	162410 (49773)	0.0078 (0.019)	0.0082 (0.021)	9 (5)	19258 (6239)	0.0012 (0.013)	0.0015 (0.017)
Run5	27 (8.5)	161180 (42741)	0.0072 (0.017)	0.0077 (0.019)	10 (3.7)	19375 (9955)	0.0014 (0.012)	0.0016 (0.015)
Run6	34 (12)	188194 (36851)	0.0076 (0.016)	0.0080 (0.019)	14 (5.5)	24121 (8178)	0.0013 (0.010)	0.0015 (0.013)
Run7	23 (7.6)	148550 (47019)	0.0089 (0.020)	0.0095 (0.023)	5 (3)	17270 (6398)	0.0010 (0.015)	0.0014 (0.022)
Run8	34 (10.7)	226632 (41173)	0.0088 (0.017)	0.0093 (0.019)	15 (5.4)	34779 (6032)	0.0012 (.009)	0.0014 (0.013)
Run9	26 (8.9)	183714 (43584)	0.0091 (0.020)	0.0097 (0.022)	10 (5.9)	19297 (5801)	0.0009 (0.010)	0.0010 (0.013)

Table 5-2. List of the features used to analyze and predict species extinction. Each feature is computed at each time step per species

Feature	**Definition**
indivNo	The total number of individuals
specNo	The total number of species
deathRatio	The ratio of the total number of deaths to the whole population
deadAgeRatio	The ratio of the number of deaths due to oldness to the whole population
deadEnergyRatio	The ratio of the number of deaths due to lack of energy to the whole population
killedRatio	The ratio of the number of killed individuals by predators to the whole population

reproducRatio	The ratio of the newborn individuals to the whole population
reproducFailRatio	The ratio of the number of failed reproduction to the whole population
compactness	The average number of individuals per cell, also called the population density
interbreedRatio	The ratio of the births due to the interbreeding to the total births
birthRatio	The ratio of the number of new born individuals to the whole population
energy	The average energy of all individuals in the species
predClose	The average perception of the predators' distance
foodClose	The average perception of the food's distance
preyClose	The average perception of the friends' distance
localFoodQuant	The average perception of the quantity of food in the vicinity
localPartnerQuant	The average perception of the quantity of partners in the vicinity
geneDivers	The diversity of alleles for all loci based on the entropy calculation
geneCompl	The number of loci having active alleles
evolDist	The average genetic distance between the reference genome (origin) and the current genomes
matingDist	The average genetic distance between mates
speed	The average speed (number of cell per time step) of all individuals in the species
age	The average age of all individuals in the species
deadAge	The average age at the time of death
escapRatio	The ratio of escape from predators to the whole population
foragRatio	The ratio of searching for food to the whole population
socializRatio	The ratio of socialization (try to find other prey) among preys to the whole population
explorRatio	The ratio of exploration to the whole population
eatRatio	The ratio of food consumption to the whole population
sedentRatio	The ratio of immobile individuals to the whole population
parentInvestEnergy	The ratio of energy which is transferred to a new individual at the birth time and decreases the parents' energy (cost of the offspring care)
innerEnergy	The average perception of the amount of individual's energy
deadEnergy	The average energy at the time of death
parent1-matingAge	The average age of choosy partner at the time of mating
parent1-matingEnergy	The average energy of choosy partner at the time of mating
parent2-matingAge	The average age of chosen partner at the time of mating
parent2-matingEnergy	The average energy of chosen partner at the time of mating
spatialDivers	The dispersal of individuals based on the species center
patchAreaRatio	The ratio of the area of a species patch (the number of cells that a species occupies) to

	the species population
patchCircum	The number of outer cells in the species patch
patchShapeCompl	The ratio of the area of a species patch to the area of its bounding box (smallest square box that covers the patch area)
spatialCompl	Measure that shows how the species patches occupies the world
spaceRatio	The ratio of the species patch area to the area of the world
patchNoRatio	The ratio of the number of patches of one species to its population
patchSizeAvg	The average patch area of one species patches
multiSpeciesCellRatio	The ratio of the number of cells that are shared between more than one species to the species patch area
contagion	Measure that shows how disperse are the whole patch types
fractalDim	Measure that describes how one species occupies its area

5.3. A machine learning approach

To study the important features involved in the extinction of species in EcoSim, we used the 49 features defined in section 5.2 to build a predictive model. Species were labeled according to whether they reach extinction in the next 100 time steps of the simulation. Afterwards, we classified the different species with respect to various demographic, environmental, and genetic characteristics in order to discriminate between extinct and non-extinct species. For purposes of classification, we required a training set, a test set, and a validation set. The training set was used to build the classifier, the test set was used to evaluate its accuracy, and the validation set verified that the classifier was able to capture generic rules corresponding to the problem. In order to guarantee the generality and accuracy of the extracted features and rules, we employed three different combinations of three runs (run1+run2, run1+run3, run2+run3), each combination containing roughly 400,000 data samples. Each time, 40,000 data sampled from one combination were used to build the training set, and the remaining samples were used to build the test set. The data from the other runs (runs 4 through 9) were used to construct the validation sets (about 1500 000 data samples in total).

We employed several feature selection algorithms such as Best Fit, Greedy Stepwise, Genetic Search, and Ranker (with InfoGain and GainRatio evaluators) in Weka [172] using default parameters setting in order to remove irrelevant features. The factors selected by a majority of the algorithms were retained. This approach helps to avoid overfitting in the prediction model by selecting the most relevant features.

Because many of the remaining features were intercorrelated, we calculated the correlation between each pair of features using correlation analysis by measuring the correlation coefficient. Correlated and irrelevant features have the potential to degrade the performance of the prediction model [191]. For this reason, in highly correlated groups of features, we retained one feature in each group and removed the rest of them in order to obtain the most parsimonious but adequate model for predicting extinction risk.

In the next step, we categorized the remaining features according to the three broad categories of genetic, demographic and environmental described by Griffen and Drake [186]. We then applied the C4.5 [128] algorithm to build a decision tree, which is our prediction model. The rationale for using decision trees is that the obtained trees can be used as a prediction model and also for extracting rules based on critical threshold values of the features. In order to determine the class of one new sample, one should start from the root node and, based on the test outcome, move to the next level of the tree, repeating this procedure to reach a leaf. However, in order to focus on the most significant rules, only the leaves of the decision tree that match a large number of data samples with high accuracy were considered. We pruned the decision tree in order to retain only the rules that match with at least 1000 samples to obtain more compact rule set.

First, we built the decision tree for each feature to discern their significance separately, given that this is a widely used method for testing correlates of extinction risk [192]. Further, we used the decision tree to discern the significance of each broad category (demographic, genetic, and environmental) independently with respect to extinction.

We used different metrics to evaluate the classifier such as the TP rate (or Sensitivity or Recall), which is the percentage of actual positive samples (species extinction in this case) that are correctly identified as such; the TN rate (or specificity) which is the proportion of actual negative samples (no extinction) that are correctly identified as such; precision (positive predictive value or PPV), which is the proportion of all true positives against all the predicted positive results; negative predictive value (NPV) which is the proportion of all true negatives against all the predicted negative results; F-measure which is a harmonic mean of precision and sensitivity and finally the Area Under the ROC Curve (AUC) where the ROC curve depicts the relative trade-off between sensitivity and (1- specificity) in order to evaluate the accuracy of each classifier [193]. Because we wanted to focus on the rules, we picked the combination run1+run2 as the train set for rule extraction and the rest of the runs as the validation set.

5.4. Results and Discussion

5.4.1. Feature selection and correlation analysis

The results of the feature selection algorithms using 10-fold cross-validation are summarized in Table 5-3. Most of the feature selection methods selected the same features, although there were some differences in rank-based methods in comparison with other methods. For example, indivNo was not selected by the first three methods, whereas its rank was low using ranking methods where lower rank shows the feature to be more important.

Table 5-3. The features selected by applying five different feature selection methods to the initial 49 features resulting in the reduction of the number of features to 25. In the first three algorithms, the numbers specify the number of folds in 10-fold cross-validation for which the feature has been selected by the algorithm. Therefore, the higher values show the importance of the features and for the InfoGain and GainRatio algorithms the numbers are the average rank of the feature based on two different ranking criteria i.e., information gain and gain ratio and lower values show the more important features). Other features out of 49 were discarded by the feature selection methods.

Feature	BestFit	Genetic	Greedy	InfoGain	GainRatio
interbreedRatio	10	10	10	22	18.9
killedRatio	10	10	10	17	5.5
deadEnergy	10	6	6	21	18.4
parentInvestEnergy	7	9	7	23	23.1
preyClose	10	10	10	20	13
Foraging	4	7	4	25	24.9
socialRate	10	10	10	18.1	16
waitRate	7	7	7	24	22.3
parent1_reprodAge	9	9	9	10	6.6
parent1_reprodEnergy	10	10	10	15	2.8
parent2_reprodAge	8	7	7	14	9.4
parent2_reprodEnergy	10	10	10	16	8.6
matingDist	10	10	10	11.6	2.8
spaceDens	10	10	10	3.1	1.2
patchsizeAvg	2	2	1	12.1	10.1
shapeCompl	8	2	0	12.3	14.2
fractalDim	7	8	7	5.8	17.6

reprodRate	7	8	7	3.9	10.5
reprodFailRatio	5	7	6	8.4	21.7
birthRatio	5	6	4	8.4	13.9
explorRatio	3	4	3	18.9	23
patchArea	6	2	6	5.9	11.1
Compactness	1	4	1	6.5	11.3
patchCircum	1	3	1	1	3.4
indivNo	0	0	0	2	14.7

Afterwards, we found four groups of highly correlated features (Table 5-4), when we considered correlations (Pearson correlation coefficient) greater than 0.7. This value is a reasonable tradeoff because with higher values there is the risk of selecting too few correlated features, whereas with lower values, there is the risk of selecting too many features. We selected indivNo, birthRatio, parent1_reprodAge, and popDens from groups 1 through 4 respectively as they are more informative than the other features to explain the extinction and removed the rest.

Table 5-4. Four groups of highly correlated features (>0.7) using the correlation analysis method. The numbers in the parentheses refer to the correlation between the given feature and the features below it in that column.

Group 1	Group 2	Group 3	Group 4
indivNo (+0.97, +0.98, +0.75)	birthRatio (+0.7)	parent1_reprodAge (+0.93, +0.84, +0.8)	patchArea (-0.84, -0.92, -0.89)
patchCircum (+0.99, +0.76)	reprodRate	parent1_reprodEnergy (+0.84, +0.86)	fractalDim (+0.7, +0.75)
spaceDens (+0.76)		parent2_reprodAge (+0.93)	popDens (+0.88)
shapeCompl		parent2_reprodEnergy	patchsizeAvg

In group 1, indivNo had a very high positive correlation with patchCircum and shapeCompl because these features are calculated based on the cells which individuals inhabit. The reason that we do not see very high correlations between features in group 2 is that not all reproductive acts are successful in EcoSim due to a variety of factors. To have a successful reproduction, both parents must have a high energy level and also reach a high maturity level (both defined by a threshold). As a result, we observed high correlations in group 3. In group 4, patchArea has a

negative correlation with the other features, since increasing the patch area decreases fractal dimension, population density, and patch size average.

5.4.2. Feature Reduction and Categorization

We reduced the number of features to 14 using feature selection and correlation analysis. We then categorized the remaining features into three broad groups: demography, genetic, and environmental (Table 5-5) in line with the categorization proposed in [186]. In their proposed categorization, demographic features are associated with fluctuations in population due to variability in growth, reproduction, and lifespan. Thus, we placed all the features related to population, population density, birth ratio, reproductive age, and energy into demography.

Table 5-5. Three broad categories (demographic, genetic, environment) for the 14 reduced features

Demographic	Genetic	Environment
indivNo	matingDist	killedRatio
birthRatio	parentInvestEnergy	waitRatio
reprodFailRate		explorRatio
preyClose		
deadEnergy		
parent1_reprodAge		
socializRatio		
interbreedRatio		
Compactness		

The genetic category is associated with inbreeding depression and so in this category we placed matingDist, which characterizes the genetic distance between two individuals, and parentInvestEnergy, which is the amount of energy transferred to the offspring at birth, related to the genetic makeup of the parents. The environmental stressors in EcoSim are predators and food scarcity and so in this category we placed killedRatio, the ratio of prey killed by predators, and waitAction which is the attempt by an individual to save energy or avoid predators. Food scarcity may lead to exploration and so explorRatio was included under Environment. As Griffen and Drake [186] suggest, some of these features may affect one another. For example, if there is a larger number of prey (demographics) then this may positively affect the killedRatio (environmental) since there are more prey available to predators. Finally, as will be discussed

below, Table 5-6 and Table 5-7 illustrate that the three broad categories of demographics, genetics, and environment are effective predictors of extinction.

5.4.3. Extinction prediction rules

We evaluated each of the reduced 14 features to determine their ability to predict extinction. Eight out of the 14 features demonstrated a sufficiently high level of accuracy as shown in Table 5-6. The remaining features failed to adequately predict extinction on their own, although they demonstrated better extinction predictability when combined with other features.

Table 5-6. The extracted rules along with their levels of accuracy. For Recall and Precision columns, the values for the extinction rules are TP Rate (ability to identify extinction samples) and Positive Predictive Values (shows how many percentages of extinction samples predicted). The values for the no extinction rules are TN Rate (ability to identify no-extinction samples) and Negative Predictive Values (shows how many percentages of no-extinction samples predicted) respectively. F-Measure is a harmonic mean of precision and recall. AUC is the area under the ROC curve. The hit ratio represents the percentage of the dataset covered by the rule.

Rule Code	Recall	Precision	F-Measure	AUC	Hit Ratio
Ext_R1	0.96	0.84	0.90	0.91	32%
NoExt_R1	0.92	0.98	0.95	0.91	68%
Ext_R2	0.94	0.80	0.97	0.90	29%
NoExt_R2	0.98	0.97	0.98	0.90	62%
Ext_R3	0.94	0.86	0.90	0.91	30.1%
NoExt_R3	0.93	0.97	0.95	0.91	69.8%
Ext_R4	0.91	0.81	0.86	0.85	27.8%
NoExt_R4	0.90	0.96	0.92	0.85	56.3%
Ext_R5	0.96	0.80	0.87	0.90	27.3%
NoExt_R5	0.91	0.98	0.95	0.90	72.6%
Ext_R6	0.96	0.82	0.88	0.90	29.4%
NoExt_R6	0.91	0.98	0.95	0.90	70.6%
Ext_R7	0.91	0.84	0.87	0.85	27%
NoExt_R7	0.93	0.96	0.94	0.85	65%
Ext_R8	0.88	0.85	0.87	0.90	32.8%
NoExt_R8	0.92	0.94	0.92	0.90	61.3%

Table 5-7. Combined Extinction/non-extinction Rules (E = extinction, ~E = no extinction, → = there is a tendency towards) E.g., Ex_R1 reads "if the population number falls below a critical threshold, T_p, then there is a tendency towards extinction.".T_p, T_{pr1}, T_{pr2}, T_{pr3}, T_{d1}, T_{d2}, T_c, T_b, T_{s1}, T_{s2}, T_m, T_{k1}, T_{k2} are threshold values for the following rules such that $T_{d1}<T_{d2}$, $T_{pr1} \leq T_{pr2}< T_{pr3}$, $T_{s1}< T_{s2,}$ $T_{k1}< T_{k2}$)

Features	Category	Rule Code: Extinction / No extinction Rules
indivNo	Demographic	Ex_R1: (indivNo $\leq T_p$) → E NoEx_R1: (indivNo>T_p) → ~E
Parent_reprodAge	Demographic	Ex_R2: (parent_reprodAge$\leq T_{pr1}$) → E NoEx_R2: (T_{pr2}<parent_reprodAge$\leq T_{pr3}$) → ~E
deadEnergy	Demographic	Ext_R4: (deadEnergy$\leq T_{d1}$) → E NoExt_R4: (deadEnergy> T_{d2}) → ~E
Compactness	Demographic	Ext_R5: (compactness $\leq T_c$) → E NoExt_R5: (compactness > T_c) → ~E
birthRatio	Demographic	Ext_R6: (birthratio$\leq T_b$) → E NoExt_R6: (birthratio> T_b) → ~E
socializRatio	Demographic	Ext_R8: (socializRatio$\leq T_{s1}$) → E NoExt_R8: (T_{s1} <socializRatio$\leq T_{s2}$) → ~E
Mating distance	Genetic	Ext_R3: (mating distance $\leq T_m$) → E NoExt_R3: (mating distance > T_m) → ~E
killedRatio	Environmental	Ext_R7: (killedRatio$\leq T_{k1}$) → E NoExt_R7: (T_{k1}<killedRatio$\leq T_{k2}$) → ~E

Based on accuracy metrics, the rules in Table 5-6 are highly efficient with respect to extinction prediction. These results are the average results for training, test, and validation sets. In addition, because we had three different training sets and the threshold parameters associated with the features changed slightly, we repeated the experiments for every training set and calculated the average. As an example, in the genetic category we observed 0.0389, 0.0354, and 0.0584 as

thresholds for mating distance when we applied three different train sets. Some of the rules in Table 5-6 do not cover 100% of the dataset. For example, Ext_R2 and NoExt_R2 combined cover 91% of the dataset and Ext_R4 and NoExtR4 combined cover only 84.1% of the dataset such that Ext_R shows extinction rule and NoExt_R shows no extinction rule. This is to be expected, as a decision tree uses a combination of rules to make predictions, so that one rule does not necessarily cover all the data samples.

5.4.4. Interpretation of combined extinction/no extinction prediction rules

The rules in Table 5-6 can be combined for extinction/no extinction and categorized in terms of demographic, genetic, and environmental rules (Table 5-7) in order to more fully discern their biological significance.

5.4.4.1. Rules Based on Demographic Features

There are six extinction/no extinction prediction rules based on the demographic features of compactness, deadEnergy, birthRatio, socializRatio, indivNO and parent_reprodAge (Table 5-7). Rules R1 (Ext_R1 and NoExt_R1) and R5 (Ext_R5, and NoExt_R5) indicate that if the population number (indivNo) and population density in the prey species goes below critical thresholds then the species tends towards extinction. Small populations are more likely to become extinct because of demographic stochasticity [194]. Further, there is the potential for individuals to suffer reduced fitness from insufficient cooperative interactions with conspecifics (leading to inbreeding depression and hybridization) or individuals may have difficulty encountering potential partners resulting in socialization ratios and birth ratios falling below critical thresholds (rules R8, R6 respectively). These effects can cause negative growth rates of populations and lead to an unstable equilibrium at small population sizes, below which the population is more likely to become extinct [195].

Drake and Griffen [174] maintain that the causes of a population's decline are very important factors in predicting extinction. Rule R2 (Ext_R2 and NoExt_R2) relating to the parental reproductive age indicates that when parental reproduction is within an optimal range (not too young, not too old), the species does not tend towards extinction, whereas when the parental reproduction age is lower than the lowest threshold value, T_{pr1}, the species tends towards extinction. However, in the samples where the ParentRepAge was less than the threshold valueT_{pr1}, the birth ratio was zero or very close to zero. This means that the part of rule R2

predicting extinction has as its basis the indirect effect of the birth ratio. The rule relating to the feature of deadEnergy (the average energy at the time of death), viz., R4 (Ext_R4 and NoExt_R4) predicts that the species will tend towards extinction if the amount of energy that individuals have at the time of death is below a critical threshold. Lower energy at the time of death suggests either food resource depletion or predator stress such that prey do not have enough time to find food. If there are not sufficient food resources available to sustain a larger number of individuals, then there will be a population decline and hence an increased chance of species extinction.

5.4.4.2. Rule Based on Mating Distance (Genetic Feature)

Rule R3 implies that closer mating distances below a critical threshold are associated with species extinction. Mating distance is connected with how genetically similar the two parents are. If the average mating distance of the population is less than a critical threshold the species will tend towards extinction (Table 5-6 and Table 5-7). The feature of mating distance is clearly associated with inbreeding depression. The lower value for this feature indicates that mating has occurred between two individuals with very similar genes such as siblings, which decreases individual fitness and population growth rates as has been observed in nature [196], [197], [198], [195]. Moreover, Frankham [199] demonstrated that inbreeding decreases the effective population size and can lead to extinction. He also mentioned that the most probable relationship between extinction and inbreeding is a threshold relationship with low probability of extinction below that threshold and higher thereafter, similar to what we obtained in EcoSim.

5.4.4.3. Rule Based on KilledRatio (Environmental Feature)

Rule R7 indicates that when the ratio of prey killed by predators is below a lower critical threshold, T_{k1}, the species tends towards extinction, whereas when the kill ratio is within an optimal range (between the lower threshold, T_{k1}, and a higher threshold, T_{k2}) the species tends towards no extinction. Although the first part of the rule predicting extinction when the kill ratio is below T_{k1} may not initially seem to make sense, it becomes understandable when applied to lower population species which are less affected by predation vs. larger population species. We observed lower predation rates for low population species, given that predators tend to gravitate towards cells with higher populations. For example, 82% of the species with less than 5 individuals went to extinction without being affected by predation and 67% of the species with five to 10 individuals went extinct without being affected by predation, illustrating that the most important reason for extinction of species with low populations is their population size. (27% hit ratio of the rules). The second part of the rule predicting non-extinction when the kill ratio is

within an optimal range (between T_{pr2} and T_{pr3}) becomes intelligible when applied to larger population species (usually more than 100 prey) that are affected by predation but generally do not go to extinction (65% hit ratio).

5.4.5. Combining the features in each category

When we combined the features in each category, we obtained more accurate rules than those based on individual features, although the decision tree was more complicated. In all categories we observed better AUC values and generally better F-measures in comparison with when we used just one feature. Sample rules in the decision tree are shown in Table 5-9, with the accuracy metrics for each category presented in Table 5-8. For example, in the demographic category, if the ratio of births and the socialization ratio (conducive to mating and reproduction) are above critical thresholds and the reproductive failure ratio is below a critical threshold, then there is a tendency towards no extinction. The accuracy metrics in Table 5-8 show that the demographic, genetic, and environmental categories are effective predictors of species extinction.

Table 5-8. Prediction results of different categories when we merged all the features in each category to predict extinction

Category	Recall	Precision	F-Measure	AUC	Hit Ratio	State
Demography	0.90	0.92	0.91	0.96	30%	Extinct
	0.96	0.95	0.95	0.96	70%	No Extinct
Genetic	0.88	0.91	0.90	0.92	30.5%	Extinct
	0.95	0.93	0.94	0.92	69.5%	No Extinct
Environment	0.88	0.89	0.89	0.94	30%	Extinct
	0.94	0.93	0.93	0.94	70%	No Extinct

Finally, we combined all of the rules from across the three broad categories (Demographic + Genetic + Environment) and applied them to the datasets. Altogether, the results shown in Table 5-10 indicate that combining all of the rules across the three categories gives rise to even better results than those achieved using rules applied to just one feature or combined rules in an individual category. What this demonstrates is that as rules and categories are combined, the accuracy level and hence predictive potential increase. This is an interesting result since the rules for each feature separately already have a high level of predictive accuracy, although even higher levels of predictive accuracy can be achieved by combining the rules, thus augmenting the ability to make predictions regarding extinction. Combining all of the rules across all categories covered all samples in the dataset. In addition, higher Recall values were reached except for Ext_R2 and

NoExt_R2 combined, although in this case the rule coverage was 91%. Moreover, F-measures and AUC values improved in comparison with when we used just one feature or individual categories.

Table 5-9. Sample rules with combined features in each category (~E = no extinction, → = there is a tendency towards)

Category	Rule
Demography	(birthRatio> T_b & socializRatio>T_s & reprodFailRatio≤T_{rp}) → ~E
Genetic	(matingDist> T_m & parentInvestEnergy<T_{sob}) → ~E
Environment	(T_{k1} <killedRatio ≤ T_{k2} & waitRatio> T_w) → ~E

Table 5-10. Prediction results by applying all rules: Demography + Genetic + Environment (C1: combination of all extinction rules; C2: combination of all no extinction rules)

Rule	Recall	Precision	F-Measure	AUC	Hit Ratio	State
C1	0.96	0.86	0.91	0.93	30.5%	Extinction
C2	0.93	0.98	0.96	0.93	69.5%	No Extinction

5.5. Conclusion

In this study, we investigated 49 features associated with species extinction in EcoSim, a computer-based simulation method shown to agree with real ecosystems. Using several feature selection methods along with correlation analysis, we were able to eliminate a number of these features, resulting in 14 features which we then placed into the three broad categories of genetic, environmental, and demographic. We were able to use these 14 features to investigate whether extinction is a predictable phenomenon. For this purpose, we used data extracted from an individual-based prey-predator ecosystem simulation. We obtained a rule set for each category and showed that these rules can predict extinction in the next 100 time steps with a very high level of accuracy. We also demonstrated that these rules are generic by applying a model built on a training set to a validation set constructed using completely different simulation runs. In the rule extraction phase, we adjusted the pruning rate in the decision tree in order to simplify it. Finally, we combined the obtained rules for each of the three basic categories with the results indicating higher accuracy, in comparison with each feature separately. The acquired results suggest how powerful our proposed machine learning approach can be from several different perspectives. First, the proposed approach is able to extract important features in extinction effectively,

especially when there is a plethora of features and there is no exact knowledge about them. Second, the categorization idea helps to study the effect of features in a more fine-grained way and to extract rules associated with them accompanied by an evaluation of their accuracy. This may prove to be beneficial for conservation biologists from the point of view of being able to detect early signals of extinction. For example, we found that population size of the species and also average genetic distance of parents at breeding time in one species are really important features as we were able to predict extinction using those features alone with a high accuracy comparable to the accuracy level obtained when we used all the features in each categories such as genetic, environmental, and demographic. This is particularly useful for being able to obtain a high level of predictive accuracy based on a minimum amount of information from the environment. Further, this approach can be applied to test new hypotheses regarding new factors involved in extinction. While our results are not directly valid for real situations given that our model involves a high level of abstraction as well as being a simplification of the real world, our results provide interesting insights that could be of aid to biologists in formulating new hypotheses relating to species extinction. Finally, the model we have employed has the potential to be useful for more dedicated studies focusing on hypotheses emerging from the broad type of approach to the prediction of species extinction that we have advanced. Also to be acknowledged is the broader innovation of providing a methodology for ecological data analysis based on machine learning.

Chapter 6

6. Investigating of Species-Area Relationship in EcoSim

6.1. Introduction

The species-area relationship (SAR) is one of the most well-known and oldest patterns in ecological modeling [22], [23], [24]. SARs have a number of practical applications for managing natural communities. For example, SARs can be used for predicting the extinction rate of a species based on habitat loss or reduction [200], [201], for designing optimal reserve sizes [202], for identifying hotspots and geographical regions of high species richness [203], for assessing human impacts on biodiversity [204], for predicting the species richness of certain taxa based on richness of other species [205], and for estimating the species richness of larger regions [206].

The fundamental characteristic of SAR modeling is that species richness increases with the sampling area, with the increment rate decreasing for larger areas. Identifying the most biologically appropriate mathematical SAR model to characterize these behaviors has been one of the most important and controversial issues in biodiversity. Two of the earliest and most frequently applied mathematical models for the SAR, i.e. the power and logarithmic functions, were proposed by Arrhenius and Gleason in the 1920s [22], [23]. Subsequently, a number of researchers investigated how well these simple mathematical models fit the field data set obtained from different taxa [207], [206], [208], [209], [210], [211]. Others investigated a variety of practical applications of SAR models [212], [205], [213].

Still other researchers considered not only the simple mathematical models, but in addition tested new kinds of models based on more complex mathematical functions. Some of these new models are an extension of simple SAR models, while others are completely new functions for this domain. For example, several authors have argued that there is no universal model to describe all data sets and that the best model should be discovered for each data set separately [214], [215], [23], [216]. Others have proposed various models for different spatial scales [215], [217], [22]. Keeley and Fotheringham [218] have argued for a re-adoption of the traditional exponential model for certain kinds of plant data sets while retaining the power model for other kinds of data sets, depending on the structure of the plant community.

However, there is support in the literature for the overall adequacy of the power function family in representing species-area relationships. Plotkin et al. [206] proposed a generalization of the

power function, whereas Dengler [24] suggested using the simple power function as a general model for all kind of species-area data on any scale. Ulrich and Buszko [208], Martin and Goldenfield [219], Drakare et al. [220], Surendra and Singh [221], Azovsky [213], along with Merwe and van Rooyen [222] all advocate the power function as providing an adequate account of species-area relationships with respect to selected data sets. Finally, Triantis et al. [223] reported that the power model along with other simple models best represent the island species relationship (ISAR).

There are a number of plausible explanations in the literature regarding the apparent variation of SARs at different scales, for different types of species, and for various geographic locations. For example, Connor and McCoy [217] argue that the relative abundance distribution of the species or the range of sampling in one area can affect SARs. They also believe that different taxa within various spatial scales could generate a different functional form of SARs. He and Legendre [224], Martín and Goldenfeld [219], and Tjørve et al. [225] have shown that SARs are affected by species abundance and spatial distribution factors like species dominance and the level of aggregation. Sampling methods may also change the SAR model as discussed in Scheiner [226] and Dengler [24]. Drakare et al. [220] have observed that SARs are affected significantly by sampling schemes, spatial scales, and types of taxa or habitat. In other experiments, the effect of spatial distribution and aggregation information, spatial scale, evenness or measure of distribution of relative abundances of different species in a community, species abundance model, latitude, self-similarity, and sampling effort have been investigated [224], [227], [220], [225], [228], [229], [222], [230].

Generally speaking, there are a number of shortcomings in the papers published on SAR models. First, several studies assume the power function as the default SAR model without considering other possible models [220], [221], [231]. Second, sampling methods and sampling scales have been neglected in many studies when researchers search for the best SAR models with several exceptions such as [232], [212], [233]. Finally, articles in the literature that address the issue of how to interpret the SAR coefficients tend to assume the simple power function as the default function without considering their meaning with respect to alternative functions [234], [235], [236], [231], [237].

To help resolve the debate regarding the best SAR function, we employed species richness data sets from computer simulations in order to address the following questions.

1. Is the power function the best suited SAR model overall?

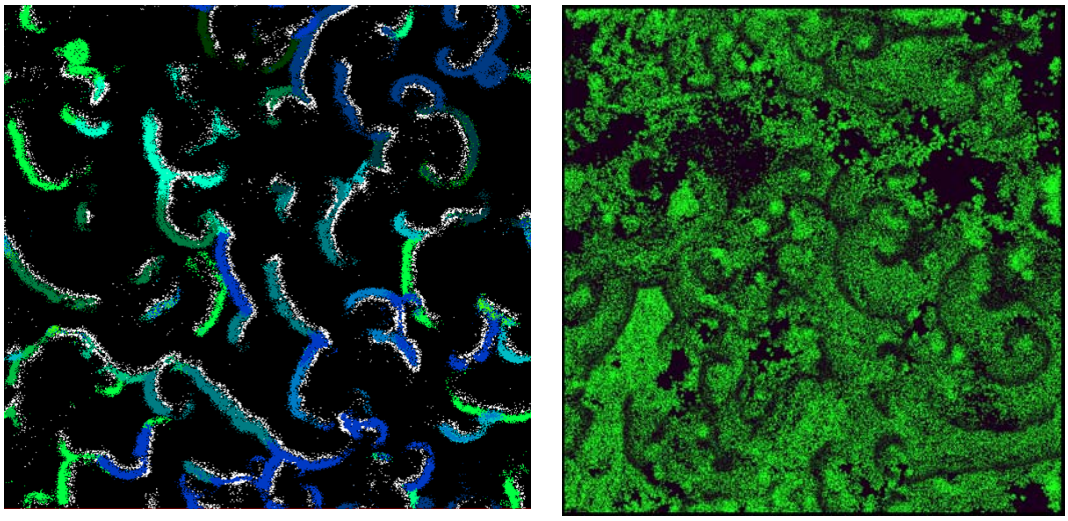
2. How do nested sampling and random sampling affect the shape of the SAR curves?
3. Do different sampling scales affect the SAR models?
4. Is there any correlation between SAR model coefficients and spatial information?

To address these questions, we employed our individual-based modeling simulation, EcoSim, to investigate the SAR. This method helps with the investigation of the species area relationship by considering the abundance and the distribution of species from a finer-grained level of description in terms of the behavior of individual organisms. The number of species in a given region is the outcome of the evolutionary processes of speciation, extinction, and migration to that region which in turn are caused by processes operating at the level of individuals [238]. Thus, it is useful to study the dynamics of the SAR at the level of individual organisms which form the species. To answer the first three questions, we collected 28 different functions through literature searches and examined them for various sampling scales and sampling methods. For the last question, using potentially informative spatial information, i.e., spatial factors, gathered from previous studies and applying machine learning techniques [193], [239], [240], we attempted to find important factors that aid in the interpretation of the models' coefficients.

6.2. Data Generation

The outcome of the interaction between individuals in a given ecosystem gives rise to ecological structures such as the species area relationship. We employed EcoSim to investigate the SAR. The predator-prey feature of EcoSim is useful in the investigation of the SAR since it contributes to species richness along with being a realistic depiction of actually existing ecosystems where there are both predator and prey species present.

Figure 6-1 shows a snapshot of the world in a time step of the simulation with emerging grouping patterns of species and grass distribution. There are several patterns emerging during the simulation such as strong clustering of individuals, a high overlap of species spatial distribution within clusters, forming coherent clusters with a large amount of empty space (which could lead to a barrier effect as for an ocean) among them (in average about 95% of the sub sample are empty cells) and also a partial overlap of predators and their prey (see snapshot of the world in Figure 6-1).



(a) (b)

Figure 6-1. The snapshot of the virtual world in one specific time step. a)The white color represents predator species and the other colors show different prey species. b) The pattern of grass in the world

We used the data extracted from nine independent runs of EcoSim to analyze the spatial distribution patterns of the species using different sampling techniques. General information, along with the standard deviations, is provided in Table 6-1. Although the initial parameters are the same for all runs, they are completely different because of the chaotic properties of the data generated by EcoSim [241], that leads to different interactions between individuals and between individuals and their environment and finally results in different runs. Considering the number of individuals in each simulation (about 200,000) and their interactions with the environment and also considering the large size of the world (1000×1000 cells), and given that each run has 25000 time steps, it follows that each run differs markedly from the other runs. Moreover, there are a few features of the simulations that involve stochasticity. For instance, at initialization time the amount of grass units is randomly determined for each cell (a value between 1 and MaxGrass, which is a parameter of the simulation). As another example, the maximum age of an individual is determined randomly at birth from a uniform distribution centered at a predefined value (MaxAge, see section 2.3.5).

Various sampling methods are used to obtain the census of species in a habitat to determine SAR model. For sampling purposes, we considered four different sampling scales: 150×150 cells or small scale (SS); 300×300 cells or intermediate scale (IS); 600×600 cells or large scale (LS), and 900×900 cells or very large scale (VLS). For each scale, we used two different sampling methods that will be discussed in the next section. In addition, we calculated some spatial information

which will be introduced later. The sampling and calculation of this information was performed every 200 time steps of the simulation to consider different habitat conditions in EcoSim. Considering the average speed of movement of the individuals (2.82 cells/time step), 200 time steps is reasonable to allow clusters of individuals to migrate and therefore to obtain global configurations of spatial distributions that are sufficiently different to be considered as independent. With nine runs and 24000 time steps per run (from 1000 to 25000) and four different sampling scales, we had 4320 sampling data and spatial information on 1080 different habitat conditions.

Table 6-1. General information, along with their standard deviations, about the nine runs used for this study

	Prey		Predator	
Runs	**Species Number**	**Population**	**Species Number**	**Population**
Run1	21 (5.8)	162130 (49254)	5 (3.5)	18414 (6021)
Run2	33 (9.2)	202820 (25536)	19 (5)	32108 (5845)
Run3	30 (6)	185970 (30750)	18 (4.5)	29831 (5350)
Run4	23 (11.5)	162096 (72170)	12 (7.8)	18055 (8058)
Run5	23 (8.6)	151581 (51858)	5 (4)	17309 (7542)
Run6	32 (7)	205620 (32319)	20 (5.97)	32397 (5038)
Run7	30 (8.8)	208020 (29776)	14 (6)	23224 (4855)
Run8	30 (7)	202566 (31781)	19(5)	34326 (5640)
Run9	26(9)	183714 (43584)	10(6)	19297 (5800)

Beta diversity measures the variation in species richness between habitats, which can then be used to explain z-values for the power function family of SAR equations, as was done for example in [242]. In order to measure beta diversity, we used two R packages i.e. Betapart [243] and NStar [244] to calculate four beta diversity indices: the turnover and nestedness components of the Sorensen indices; the familiar Whitakker function measuring beta diversity as a proportion of gamma diversity (species diversity in a landscape) to alpha diversity (species diversity in a habitat within a landscape), and finally, N* which is an extended version of the Whittaker function that measures how species occupancies vary across regions [244]. The Sorrenson and Jaccard indices along with the Whittaker function measure what Storch et al. [245] call 'broad-sense' species turnover where the magnitude of gain or loss of species is ignored. On the other hand, the N* function measures what Storch et al. [245] call 'narrow-sense' species turnover where gain or loss of species are taken into account.

We picked 20 random time steps of the simulation and calculated the beta diversity measures for the four defined sampling scales (SS, IS, LS, VLS) and the two sampling methods (random and nested). For the purpose of replication, we repeated the sampling 30 times for each scale in each selected time step and then computed the average beta diversity measures. Finally, we performed a regression analysis for N* and the slope *z* in order to discern a possible relation between beta diversity and the slope *z* which measures the rate of species increase for a given area [236].

6.3. Sampling methods and Curve fitting

Two main sampling approaches were applied: nested and independent areas sampling, i.e., random sampling [217]. Mean species richness was calculated taking inherent stochasticity into account. For every 200 time steps of the nine simulation runs, we repeated the following procedure for four different sampling scales: We picked 30 random cells as the centres for nested sampling as well as the starting points of the random sampling. The sizes of the subplots are based on sampling scales. For example, for SS, they are equal to ($SS - k \times delta$) ×($SS - k \times delta$) where k=1 for the largest subplot and k=24 for the smallest one. When other sampling scales are used k varies from 1 to 29 and *delta* is 6, 10, 20 and 30 for SS, IS, LS and VLS, respectively. For example, in the case of small scale size (SS), we used a 150×150 plot size as the main plot around all the 30 centres and calculated the mean species richness for them (Figure 6-2 (b)). Then we built inner subplots by decreasing the dimensions of the main plot by a quantity *delta* and again calculated the average richness (Figure 6-2 (a)), repeating this step 25 times for SS and 30 times for the remaining sampling scales. In other words, we placed 30 main plots in the world and then calculated the mean species richness for the subplots with equal sizes located inside the main plots. We used a *delta* value of 6, 10, 20 and 30 for SS, IS, LS and VLS, respectively. For the previous example, subplot sizes were 144 × 144, 138 × 138, 132 × 132 ... . The smallest subplots in SS, IS, LS and VLS are 6×6, 10×10, 20×20, and 30×30 respectively. Similarly, in random sampling, in each main plot we calculated the average richness of the same subplot sizes (decreased by delta each time) as in nested sampling but instead of nested placement, we used random sampling (Figure 6-2 (c)). Similar to nested sampling, we recorded the mean richness of every plot size. Obviously, in random sampling, depending on the initial positions of sub-plots, there is the possibility of overlapping sampling areas.

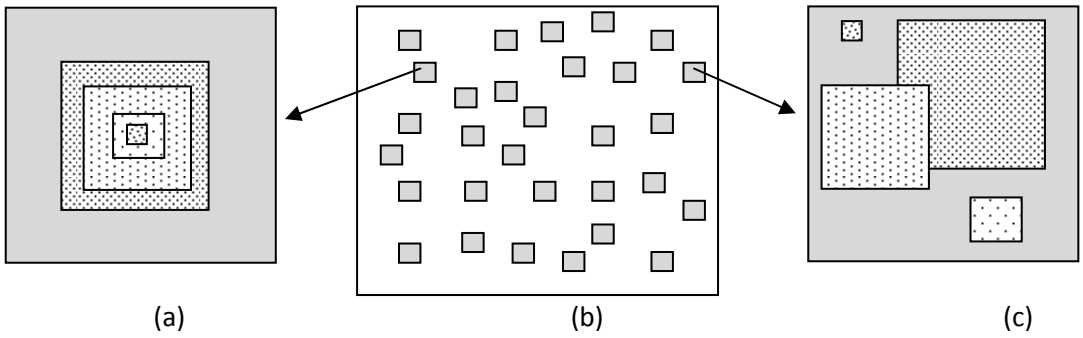


Figure 6-2. The two applied sampling methods for 30 main plots (grey boxes) with four different sizes, SS, IS, LS and VLS sizes. The richness is calculated by averaging over each of the equal size subplots (dotted box) for the 30 main plots. 25 subplots were used for SS and 30 for the rest of sampling scales. The sizes of the subplots are based on sampling scales. a) nested sampling used in every main plot, b) 30 main plots in the habitat, c) random placement

Entire sampling data sets were fitted with 28 functions (Table 6-3) collected from several articles [246], [214], [215], [23], [216], which are widely considered as the most promising functions to describe the SAR. For this purpose, the non-linear regression algorithm implemented in Python libraries was employed (www.python.org). A genetic algorithm [247] was applied to find the best starting point of the regression because sometimes the fitting result was not good due to the random starting points (some parts of the implementation were obtained from http://zunzun.com).

For the evaluation of fit, we applied AICc (Akaike's Information Criterion corrected for small n) (Equation 6-1) as the goodness-of-fit criterion and ΔAICc rank (Equation 6-2) to sort the functions based on average rank in the whole data set, as defined in Burnham and Anderson [248], and Dengler [24]. In fact, for every data set, we calculated the rank of each function based on ΔAICc and then the mean rank of all the functions on all data sets. The function with the lowest ΔAICc has the best rank. R^2_{adj} (Equation 6-5) was also calculated to show the quality of the fit. In addition, we used the extrapolation capability of functions as another criterion. For this purpose, we defined Extrapolation Sum of Square Error (ESSE) as the square of difference between the species richness predicted by the model and the real richness value for 20% of the largest areas in every data set of our 4320 data sets. In this case, for every data set, we removed the 20% largest area, fitted the rest of the data set to all models, and then calculated ESSE for the largest area we removed. Finally, to verify the significance of the fit, the F-test was applied. For normality analysis, the Shapiro-Wilk test was applied. Table 6-2 summarized the developed algorithm to find the best SAR model.

$$AIC_c = AIC + \frac{2k(k+1)}{n-k-1} \quad (6\text{-}1)$$

$$\Delta_i AIC_c = AIC_{c,i} - AIC_{c,\min} \quad (6\text{-}2)$$

$$AIC = n\ln(\delta^2) + 2k \quad (6\text{-}3)$$

$$\delta^2 = \frac{RSS}{n} \quad (6\text{-}4)$$

$$R^2_{adj} = 1-(1-R^2)\frac{n-1}{n-k-1} \quad (6\text{-}5)$$

$$R^2 = 1-\frac{RSS}{SSE} \quad (6\text{-}6)$$

n = # of samples

k = # of function's parameters + 2

RSS : Residual Sum of Squares

SSE : Explained Sum of Squares

$\Delta_i AIC_c$ is the difference between the AIC value for function i and AIC value of the best-fitting model.

Table 6-2. The algorithms to find the best SAR model (Part I) and to build the classifier (Part II) for the selected SAR function

Part I : Find_SAR_Functions_Ranks

1) Run EcoSim in order to generate the species distribution patterns

2) For each 200 time steps of the simulation

2.1) For each area size in {SS, IS, LS, VLS}

2.1.1) Perform nested sampling

2.1.2) Perform random sampling

2.1.3) For each function in {F1 to F28}

2.1.3.1) Apply curve fitting method on the obtained sampling data

End of 2.1.3.

2.1.4) Calculate the rank and the function's goodness of fit

2.1.5) Select the best SAR function

End of 2.1.

2.2) Calculate species distribution patterns (needed for part II)

End of 2.

3) Calculate the average rank of all functions

Part II: Extract_Rules_For_Selected_SAR_Function

1) Extract coefficient values for the selected SAR functions from part I

2) For each coefficient of the selected SAR function

2.1) Discretize the coefficient values in two bins (low and high i.e. class labels)

2.2) Use the spatial information obtained in part I as the features and merge them with the class labels in previous step to build the data sets (training and testing set) for prediction model.

2.3) Apply c4.5 algorithm on data set to build the prediction model

2.4) Extract the high coverage rules from decision tree and calculate the accuracy

End of 2.

Table 6-3. Different SAR functions available in the literature (*x* is the independent variable which shows the area and the parameters are named from '*a*' to'*d*')

ID	Curve name	Function	Parameters (upper asymptote)	Shape
F1	Power	ax^b	2 (no)	convex
F2	Extended power 1 (EPM1)	$ax^{bx^{-c}}$	3 (no)	convex or sigmoid
F3	Extended power 2 (EPM2)	$ax^{(b-\frac{c}{x})}$	3 (no)	sigmoid
F4	Persistence1 (Plotkin) (P1)	$ax^b Exp(-cx)$	3 (no)	convex
F5	Persistence2 (P2)	$ax^b Exp(-\frac{c}{x})$	3 (no)	sigmoid
F6	Logarithmic	$a + b\log(x)$	2 (no)	convex
F7	Kobayashi logarithmic	$a\log(1+\frac{x}{b})$	2 (no)	convex
F8	Negative exponential	$a(1 - Exp(-bx))$	2 (yes)	convex
F9	Chapman-Richards	$a(1 - Exp(-bx))^c$	3 (yes)	sigmoid
F10	Cumulative Weibull distribution	$a(1 - Exp(-bx^c))$	3 (yes)	sigmoid

F11	Cumulative beta-p distribution	$a(1-(1+(\frac{x}{c})^{d})^{-b})$	4 (yes)	sigmoid
F12	Common logistic	$\frac{a}{1+Exp(-bx+c)}$	3 (yes)	sigmoid
F13	Archibald logistic	$\frac{a}{b+c^{x}}$	3 (yes)	sigmoid
F14	Logistic with location parameter	$\frac{a}{1+Exp(-b(x-c))}$	3 (yes)	sigmoid
F15	Gompertz	$a(Exp(-Exp(-bx+c)))$	3 (yes)	sigmoid
F16	Gompertz with location parameter	$a(Exp(-Exp(-b(x-c))))$	3 (yes)	sigmoid
F17	Morgan–Mercer–Flodin	$\frac{ax^{c}}{b+x^{c}}$	3 (yes)	sigmoid
F18	Lomolino	$\frac{a}{1+(b)^{\log\frac{c}{x}}}$	3 (yes)	sigmoid
F19	EVF with location parameter	$a(1-Exp(-Exp(b(x+c))))$	3 (yes)	sigmoid
F20	Rational	$\frac{a+bx}{1+cx}$	3 (yes)	convex
F21	Asymptotic regression	$a-bc^{-x}$	3 (yes)	convex
F22	Michaelis–Menten (Monod)	$\frac{ax}{b+x}$	2 (yes)	convex
F23	He–Legendre	$\frac{a}{b+x^{-c}}$	3(yes)	sigmoid
F24	Generalized cumulative Weibull distribution	$a(1-Exp(-(b(x-c))^{d}))$	4 (yes)	sigmoid
F25	Power (quadratic)	$10^{(a+b\log(x)+c(\log(x))^{2})}$	3 (no)	convex
F26	Logarithmic (quadratic)	$(a+b\log(x))^{2}$	2 (no)	convex
F27	Logarithmic (general power)	$(a+b\log(x))^{c}$	3 (no)	convex
F28	Extreme value	$a(1-Exp(bx+c))$	3 (yes)	sigmoid

6.3.1. Rule extraction

One of the important issues related to SARs is to understand what spatial information affects the shape of SARs and to investigate the likely relation between spatial configuration and function coefficients.

The second part of our method involves proposing a machine learning approach to discover spatial information associated with the functions' coefficients. For this purpose, we calculated several important factors which we gathered from the literature and which were applicable to EcoSim. Therefore, for each given time step we calculated number of patches (PatchNum); average patch size (PatchSizeAvg) where patch size is the number of cells in the patch (The average patch size is 3±3 cells in our case), area and perimeter of patches, fractal dimension (FracDim) of the spatial distribution of the individuals in the world using the box-counting method [249] and contagion. We also calculated for each time step spatial complexity, SC, a modified version of spatial temporal complexity STC [250] for a two-dimensional world. SC determines the level of patchiness of the world, where higher values correspond to very complex patterns that consist of irregular patches with different sizes (something close to a random distribution) and where lower values show patterns with large-sized regularly shaped patches consisting of fully occupied or completely empty areas). Other factors were species richness, population, and sampling scale (SScale). For each time step of the simulation we performed calculations on all of this information.

To investigate which factors are effective for predicting any of the SAR models' coefficients' values, we discretized each coefficient into two bins showing high or low values and employed these values as the class labels for the data generated for every time step of the simulation that we used for sampling. Therefore, we built a data set in which each row described the spatial factors in a specific time step, with a class label that specified the value of the coefficients (low or high). The reason why we used two bins is that the more bins considered, the more complicated and more difficult the classification is, leading to a lower accuracy. Then, using machine learning techniques, we tried to classify the different habitat conditions based on the aforementioned factors such as habitat, with larger or smaller numbers of species, larger or smaller populations, and dispersed or aggregated spatial distribution of population. In this way, we were able to discriminate between various habitat conditions and different coefficients values (low or high).

For this 2-class classification problem, the C4.5 algorithm [128] was applied because, in addition to the classification, it provides a decision tree that can easily be used to extract rules that can

explain which factors are involved in determining the value of a coefficient. In addition, several feature selection algorithms such as BestFit, GreedyStepwise, LinearForwardSelection, RandomSearch, and Ranker in Weka [172] were used to select the most important factors based on the ability to discriminate between different conditions. Then, the factors selected by most of the algorithms were retained. As a result, in addition to selecting the best factors, we avoided the overfitting problem (the learning of too specific rules of the learning set with weak predictive ability) and also decreased the number of rules obtained from the decision tree focusing on the most significant ones. The data set was split for each sampling method into a training set and a test set in such a way that 70% of the instances are in the training set and the rest are for the testing and evaluation of the classifier. We calculated the accuracy for every rule in the decision tree and selected the rules with the greatest accuracy (more than 70%).

6.4. RESULTS AND DISCUSSION

6.4.1. Effect of sampling scale

We found that F1, the power function, had a higher rank (lower delta-AICc value) in SS and IS than LS and VLS for both random and nested sampling. Likewise, F25, the quadratic power function and F2, the extended power function, had a higher rank in SS and IS than LS and VLS for nested sampling. This shows that F1, F25, and F2 (all generating convex curves) are more suitable models for small to intermediate sampling scales. On the other hand, F9, the Chapman-Richards function and F18, the Lomolino function, and F24, the generalized cumulative Weibull distribution (all of which generate sigmoid curves) had higher ranks in LS and VLS than in SS and IS for both sampling methods. The dependence of the shape of the SAR curve on sampling scale has also been observed by Triantis et al. [223]. The best performing functions did not have any asymptotes for SS and IS sampling scales, while LS and VLS had an upper asymptote (Table 6-4). However, what is also worth noting is that F4, the Plotkin persistence function, which is in the power function family, was the only function in the top six rankings for all sample scales for both random and nested sampling. This suggests that the overall best-suited SAR model is an extended power function.

Table 6-4. The six best functions ranked based on ΔAICc (the values in parenthesis) for different sampling scales and the two sampling methods: nested and random.

SS		IS		LS		VLS	
Nested	**Random**	**Nested**	**Random**	**Nested**	**Random**	**Nested**	**Random**
F4 (6.7)	F4 (5.0)	F25 (6.2)	F4 (4.1)	F17 (5.4)	F9 (4.1)	F24 (3.7)	F9 (3.3)
F25 (6.9)	F9 (6.6)	F17 (6.2)	F9 (4.5)	F4 (6.0)	F4 (4.2)	F9 (4.1)	F4 (3.6)
F23 (7.1)	F17 (6.9)	F4 (6.3)	F17 (4.6)	F18 (6.1)	F17 (4.3)	F4 (4.2)	F24 (4.9)
F2 (7.1)	F20 (7.8)	F23 (6.4)	F18 (5.7)	F25 (6.2)	F18 (4.5)	F18 (5.5)	F18 (4.9)
F17 (7.7)	F18 (8.0)	F2 (6.6)	F23 (6.5)	F2 (6.6)	F23 (6.0)	F11 (6.0)	F17 (5.9)
F24 (8.0)	F23 (8.7)	F18 (8.0)	F25 (7.6)	F9 (6.7)	F25 (6.6)	F17 (6.7)	F25 (6.8)

Fridley at al. [251] and Dengler [24] believe that the best performing SAR functions do not have upper asymptotes, which agrees with our findings regarding the best-suited functions for SS and IS but which disagrees with our findings regarding the best-suited functions for LS and VLS. One plausible explanation for finding upper asymptotic functions to be the best-suited SAR functions for large and very large sampling scales is that the kinds of species are limited by the maximum size of the world in EcoSim. In support of this explanation, He and Legendre [215] and Tjorve [23] argue that there will be upper limits to the number of species that are able to colonize a given land area, so that curves describing the relationship between species richness and land area will have upper asymptotes as the size of the land area increases. In opposition to this view, Dengler [24] argues that even if it is granted that the limits of a given land area impose real limits on the number of species that can co-exist in that area, it does not follow that the best-suited SAR curve will be asymptotic. To back up this claim, he argues that the empirical data do not support the existence of upper asymptotes with respect to the variety of species that can occupy larger land areas [24].

However, a recent study conducted by Triantis et al. [223] testing 20 SAR models using 601 empirical data sets from terrestrial islands suggests that for at least very large land areas, the best-suited SAR functions are sigmoidal even though the power function is the best-suited SAR model overall. These findings cohere with our results regarding the best SAR model as discussed below.

In Table 6-4, we present the six highest ranked functions for each sampling scale across both sampling approaches using ΔAICc rankings. What is noteworthy is that the highest (#1) ranked

SAR models for SS and IS are convex whereas the highest ranked (#1) SAR models for LS and VLS are sigmoidal. This concurs with the empirical findings of Triantis et al. [223].

We conclude that although the power function family tends to be the best ranked function, SAR models are highly dependent on sampling scale and sampling approach as we found several high-ranked SAR functions with respect to various sampling scales for the two different sampling strategies (Table 6-4).

6.4.2. Effect of Sampling Method: Nested vs. Random sampling

The fitting process, in both sampling approaches, was highly significant ($p<0.0001$ in most cases) and the residuals were normal ($p>0.05$) which indicates that we cannot reject the normality hypothesis. In general, we observed that most of the functions had a higher rank in random sampling than in nested sampling. One possible explanation is that there is higher stochasticity in the random sampling versus nested sampling method. For nested sampling (Table 6-5), we found the persistence1 function or Plotkin (F4), a modification of the power function, to be the highest ranked using the ΔAICc rank criterion. This function provides a very good fit with real data sets, such as for the Polish butterfly species [208]. The second highest ranked function for nested sampling was the quadratic power function (F25), which was followed by the Morgan-Mercer-Flodin (F17) and the extended power (F2) functions. All the functions mentioned above generate convex curves with no asymptotes except for F17 which generates a sigmoid curve. F17 was the highest ranked and the third highest ranked for LS sampling scales, consistent with what was said above. Therefore, the SAR models in the power function family were the highest ranked for nested sampling using the delta-AICc rank criterion.

For random sampling (Table 6-6), the highest ranked function overall was F4 followed by F9 (Chapman-Richards function) and F17. F4 was also most frequently the highest ranked function, followed by the simple power function (F1). Although F1 had the second highest frequency of being the best ranked function (about 14%), its rank was lower than certain other functions (functions above F1 in Table 6-6) for the remaining samples. Consistent with what was said above regarding the dependence of goodness of fit on sampling scale, the sigmoidal F9 function was the highest ranked SAR function for LS and VLS scales (see Table 6-6). As with nested sampling, a few of the functions were in no cases the highest ranked function when fitted with data samples. The lowest ranked functions were the same as those for the nested sampling method, thereby demonstrating a consistency between the two sampling methods for these functions.

Table 6-5. Average goodness-of-fit values (AICc, R^2_{adj}), using nested sampling for 28 different functions sorted based on ΔAICc rank. AICc STD is standard deviation of AICc. Frequency is the proportion of the samples for which a function is the best fitted function. Extrapolation rank shows the rank of extrapolation capability.

Function	AICc	AICc STD	Frequency	ΔAICc Rank	Extrapolation Rank	R^2_{adj}
F4	-88.2	27.9	25.5%	5.8	10.2	0.997
F25	-84.6	30.3	7.0%	6.5	9.1	0.997
F17	-83.2	32.4	7.5%	6.5	9.1	0.996
F2	-83	31.7	6.4%	7.1	9.4	0.996
F9	-84.4	32.1	5.8%	7.3	8.4	0.983
F18	-84.9	27.8	0.6%	7.5	7.8	0.996
F24	-79.3	51	14.0%	7.6	9	0.957
F11	-83.4	29.2	1.3%	9.2	8.7	0.996
F23	-68.8	61.4	0.9%	9.6	12.6	0.982
F3	-70.7	38.7	4.8%	10.2	11	0.995
F5	-69	39.3	2.3%	11.4	11.5	0.995
F20	-61.7	37.6	5.1%	11.5	10.7	0.993
F1	-60.5	45	8.0%	12.4	13.1	0.992
F10	-64.9	46	1.8%	12.5	12	0.993
F27	-58.1	49.8	3.3%	13.6	11.8	0.975
F21	-47.4	43.2	3.1%	14.2	14.7	0.99
F7	-51.1	33.8	1.4%	15.5	11.1	0.989
F28	-44.1	44.1	0.0%	16.8	16.2	0.979
F26	-43	30.2	0.3%	17.5	12.3	0.985
F22	-29.5	40.9	0.2%	19.9	16.5	0.98
F16	-19.8	43.8	0.1%	21.2	20.8	0.976
F8	-12.9	45.7	0.2%	23.2	22.1	0.967
F13	2	54.3	< 0.1%	24.2	24.3	0.883
F15	23.2	70.1	0.3%	24.3	24.4	0.787
F14	-6.1	43.5	0.0%	24.7	24	0.963
F12	32.4	60.5	0.0%	25.9	26.5	0.768
F19	6.1	44.2	< 0.1%	26.3	26.4	0.944
F6	28.6	36.7	0.0%	27.3	25.5	0.864

Table 6-6. Average goodness-of-fit values (AIC, ΔAICc, R^2_{adj}), using random sampling

Function	AICc Mean	AICc STD	Frequency	ΔAICc Rank	Extrapolation Rank	R^2_{adj}
F4	-45.6	21.5	33.4%	4.2	13.1	0.988
F9	-43.9	26.2	12.5%	4.6	13	0.971
F17	-42.9	23.2	8.3%	5.4	13.2	0.987
F18	-44.4	21.7	0.6%	5.8	12.9	0.987
F25	-42.7	22.3	2.4%	7.8	13.6	0.987
F2	-41.8	23.3	1.7%	8.8	13.7	0.986
F23	-30.1	45.6	0.0%	9.8	14.6	0.971
F20	-36.6	26.3	8.0%	10	12.7	0.986
F24	-35.9	40.5	1.5%	10.1	13.5	0.946
F11	-41.1	22.5	0.1%	11.6	13.1	0.986
F7	-35.4	25.5	6.5%	12.1	12.6	0.984
F1	-29.6	31.2	13.9%	13.2	15	0.982
F3	-34.5	26.3	0.6%	13.3	14.4	0.985
F10	-30.7	31.2	2.2%	13.4	14.3	0.983
F21	-27.5	31.3	4.4%	13.7	13.4	0.982
F27	-28.7	33.7	1.2%	13.8	14.4	0.97
F5	-33.5	26.8	0.2%	14.4	14.6	0.984
F26	-32	25.2	1.4%	15.1	12.8	0.982
F28	-26.3	32.1	0.0%	15.9	14.6	0.974
F22	-20.1	32.9	0.5%	18.7	13.9	0.977
F16	-9	35.8	0.1%	21.6	15.9	0.969
F8	-5.8	38.5	0.1%	22.5	16.7	0.964
F13	9.7	48.4	0.0%	24.6	18.1	0.876
F15	32.3	62.4	0.4%	24.7	19.9	0.776
F14	1.8	37.5	0.0%	25.1	17.6	0.956
F12	39.2	55.5	< 0.1%	26.3	21.1	0.756
F19	12.5	39.2	0.0%	26.6	19.2	0.936
F6	28.8	36.2	0.0%	27.3	18.8	0.874

For both nested and random sampling, we know that F1 and F4 are both in the power function family, which strongly supports the idea that the power function family is the best fitting candidate for SARs, as argued in [210], [24]. This idea is strengthened when we examine the

ranks of F25 and F2 in both sampling methods. F17 was one of the functions that demonstrated a good fit for both sampling approaches with a rank very close to F9 as these models behave similarly [23]. F9 was also tested in [214] and reported it as having a good performance. F18 also had a relatively good rank as reported in several studies in [24], [210].

Regarding extrapolation capability for nested sampling (Table 6-5), the Lomolino function (F18) had the highest rank, followed by the Chapman-Richards (F9) and the cumulative beta-P distribution (F11) functions. The high extrapolation capability of F18 makes sense given that it was consistently ranked in the top 6 SAR functions (except for SS nested sampling) indicating its high degree of fit with the data across sampling methods and sampling scales. Moreover, our results regarding the high extrapolation capability of the F18 function agrees with the findings of [252]. F25, F17, and F2 had the next highest ranks. F6, F19, and F12 had the poorest extrapolation capability, similar to their ΔAICc rank. In random sampling (Table 6-6), the Kobayashi logarithmic (F7) and the rational (F20) functions were the highest ranked functions with respect to suitability for extrapolation followed by the quadratic logarithmic (F26) and F18 functions. The quadratic and simple power functions (F25, F1) performed well and moderately well, respectively. F4 also performed well in terms of extrapolation toward smaller scales. F4 was the best model in this regard amongst the 28 functions studied for both nested and random sampling. (We removed 20% of smallest areas and fitted the rest of the data set to all models, and then calculated ESSE and the rank of all functions for the smallest area that were removed).

6.4.3. Beta diversity analysis and the explanation of slope *z*

As mentioned above, beta diversity is a measure of species turnover between habitats that can be used to estimate and explain the slope z of the power function and its variants [253]. We calculated 4 indices of beta diversity across the 4 sampling scales as described in section 6.3. We found that z decreased in value from smaller sampling scales to larger sampling scales (see Figure 6-3 and Table 6-7), which coheres with the empirical findings in [236], [237]. This result suggests that z can be regarded as a measure of the rate of increase of species [236]; where for larger scales, the rate of increase is lower than it is for smaller scales given that there are already many species for the larger areas. Similarly, we found that the Whittaker and N* beta diversities decreased as the sampling scale increased (see Figure 6-3). The inverse relationship that we observed in our simulation study between turnover and sampling scale has in fact been observed in a number of empirical studies [254], [255], [256]. The above results suggest that there is a direct linear relationship between z and indices of beta diversity. That is, as species turnover

increases so does slope *z*. In particular, we found significant linear regressions for *z* vs. N* for all the sampling scales except for SS (see Figure 6-4), thereby making possible the estimation of *z* from N* consistent with [253] for all but the smallest sampling scales. This result provides an explanation of the slope *z*: A possible reason for the decrease of the value of *z* is that there will presumably be a less pronounced difference from habitat to habitat in the kinds of species found (less turnover) given that there are more kinds of species found in larger land areas and hence more continuity between habitats [245]. An additional result relating to beta diversity is that there is a significant linear regression between N* vs. Sorenson turnover ($p = 0.001$) where N* and turnover vary directly, and that there is a moderately significant linear regression between N* vs. Sorenson nestedness ($p = 0.05$) where N* and nestedness vary inversely (see Figure 6-5). These results make sense for two reasons. First, N* is an extension of the Whittaker index which itself is a measure of species turnover and second, as nestedness increases, presumably there would be less species gain and loss (lower N*) from one habitat to the next given that they share some species in common.

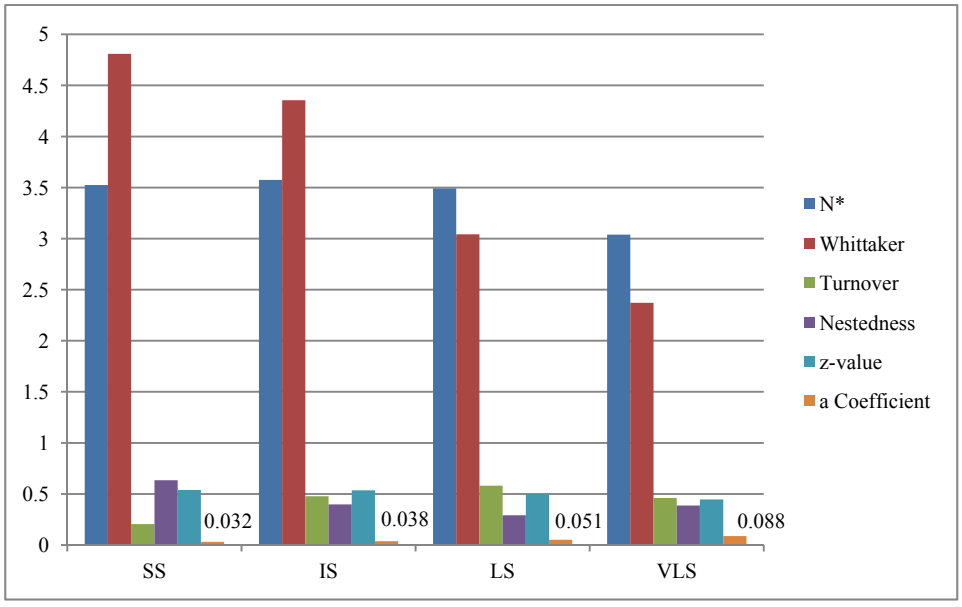


Figure 6-3. Comparison of the average beta diversity measures for four different scales along with the power function coefficients

Table 6-7. Average z-value along with standard deviation for different sampling scale size. The results show larger z-value in smaller sampling scales

Sampling scale/ Sampling method	SS	IS	LS	VLS
Nested	0.54(0.1)	0.51(0.08)	0.46(0.08)	0.4(0.07)
Random	0.48(0.05)	0.49(0.06)	0.45(0.07)	0.4(0.07)

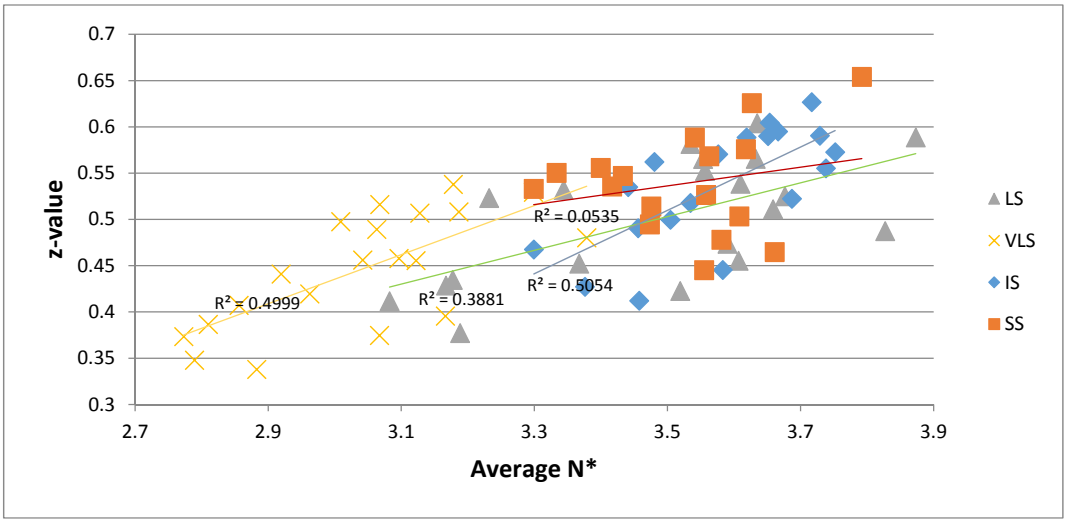


Figure 6-4. Regression analysis of the average N* and z-value for four different scales

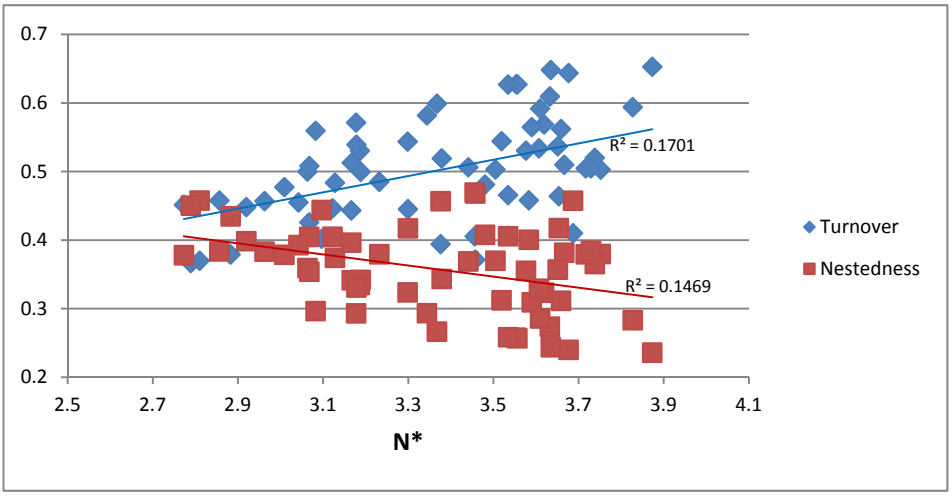


Figure 6-5. N* declines with nestedness in community structure, while it increases with species turnover

6.4.4. Coefficients interpretation based on rule extractions

As Gould [235] pointed out, many biologists have thrown up their arms adopting the view that SAR coefficients have no biologically real interpretation. An important result of our rule extraction technique is that it offers a relevant interpretation of the SAR coefficients in terms of spatial configuration along with spatial scales, ranging from moderate to very high levels of accuracy. For this purpose, we chose two functions: the Plotkin function F4, the highest ranked function for both sampling approaches and the simple power function, F1, the best ranked function for nested sampling amongst models with two coefficients.

Table 6-8 provides a list of extracted rules for the Plotkin function (F4) that paves the way for an interpretation of the coefficients 'a', 'b' and 'c' both for nested sampling (except for the constant 'c') and for random sampling. Each rule listed in Table 6-8 shows how these coefficients behave under a variety of conditions for both nested and random sampling. With respect to the coefficient 'a' for nested sampling, we see that 'a' is low when spatial complexity (SC) is less than or equal to its threshold (t_{sc}) value and when the average patch size is greater than its threshold value (accuracy of 0.84). On the other hand, 'a' is high when SC is higher than its threshold value (accuracy of 0.79). Similar rules with even higher levels of accuracy (0.92, 0.85) were extracted for random sampling (see Table 6-8 for further details).

Table 6-8. Extracted rules for F4 coefficients. Spatial complexity, patch size average, sampling scale size, and fractal dimension are the main factors to determine the coefficients values

	Condition	Coefficient	Hit Ratio	Accuracy
Nested Sampling	($SC \leq t_{sc}$) and ($PatchSizeAvg > t_{ps}$)	Low *a*	0.38	0.84
	$SC > t_{sc}$	High *a*	0.48	0.79
	($PatchNum > t_{pn}$) and (SScale > IS)	Low *b*	0.26	0.75
	($PatchNum \leq t_{pn}$) and (SScale > SS)	High *b*	0.35	0.72
Random Sampling	($SC < t_{sc}$) and ($PatchNum < t_{pn}$)	Low *a*	0.36	0.92
	$SC > t_{sc}$	High *a*	0.48	0.85
	($FracDim > t_{fd}$) and ($SC < t_{sc}$)	Low *b*	0.29	0.74
	$FracDim \leq t_{fd}$	High *b*	0.48	0.71
	SScale > IS	Low *c*	0.50	0.99
	SScale ≤ IS	High *c*	0.50	0.88

Intuitively, these rules indicate that the coefficient 'a' varies with respect to SC and either average patch size (nested sampling) or number of patches (random sampling). As Connor and McCoy [234] observe, the coefficient ‘a’ had been virtually ignored by biologists up until that time with the exception of Heatwole [257] who suggested that ‘a’ is related to the minimum area necessary to sustain a given species. Connor and McCoy [234] ultimately conclude that ‘a’ should be regarded simply as an uninterpreted constant given that some of its values for empirical data are negative and hence they have no real biological meaning. In subsequent articles, the coefficient ‘a’ has simply been taken to denote the average number of species per unit area [208], [258]. What our results suggest is that for the Plotkin function, the magnitude of the coefficient ‘a’ (that is, the average number of species per unit area) is directly proportional to the spatial complexity of a given area so that ‘a’ can also be interpreted as an indicator of spatial complexity. In a world with high spatial complexity, there are less empty spaces so the probability of finding more individuals increases, which therefore increases the probability of finding new species. As a result 'a' will have a higher value. On the other hand, with lower spatial complexity, there are more empty spaces, which leads to a lower probability of new species’ occurrence and a lower value of 'a' [250].

Although we were unable to extract rules leading to an interpretation of the coefficient 'c' for nested sampling, we were able to extract rules with very high degrees of accuracy for random sampling. For random sampling, the coefficient 'c' varies with respect to the size of the sampling scale. When the sampling scale is greater than an intermediate size (IS), the coefficient 'c' is low (accuracy of 0.99), although when the sampling scale is smaller (less than or equal to IS), the coefficient 'c' is high (accuracy of 0.88). Ulrich and Buszko [258] suggest that ‘c’ might be a corrective for deviations of the power function at smaller sampling scales. If this interpretation is correct, then higher values of ‘c’ for smaller scales may correct for errors in the power function at these smaller scales. The inability to find rules leading to the interpretation of 'c' for F4 for nested sampling along with relatively moderate accuracy levels in some cases reveals the complexity of the problem. In classification problems, finding a linear or non-linear function which is able to discriminate between several classes is not always easily achievable. Using different classification algorithms such as support vector machine [259] may improve the results although for our experiments, these algorithms did not improve the results, therefore, we do not report those results here. It is possible that adding new factors could improve the classification accuracy although we will reserve that experiment for a future study.

Finally, 'b'(slope z) varies with respect to patch number and sampling scale size for nested sampling (accuracy of 0.75, 0.72) and it varies with respect to fractal dimension, SC and sampling scale size for simple random sampling (accuracy of 0.74, 0.71).With respect to the slope 'b' the rules in Table 6-8 indicate that in larger areas, 'b' is relatively low (SScale > IS), whereas in smaller areas, 'b' is relatively high (SScale > SS). This result agrees with experimental data cited by Martin [236]. According to Martin, the slope 'b' in the power function measures the rate of species increase so that larger values of z are associated with higher rates in the increase of new kinds of species [236]. Using three spatially neutral models, Cencini et al., [231] also found that as the rate of speciation increases so does the value of z. Martin [236] hypothesizes that in larger areas, there is a lower rate of increase in species diversity given that a high number of species already co-exist, so that the number of species that have not yet colonized decreases. In a recent empirical study, Franzen et al., [237] found that as the range size of an ecosystem increases, the value of slope z decreases (meaning a lower rate in the increase of species diversity), which agrees with the results of our study.

Further, with respect to the rules in Table 6-8 relating to the Plotkin function (F4), both for nested and random sampling, there are a number of factors in the conditions governing the relative size of 'a', 'b', and 'c' such as sampling scale (SScale), patch numbers, fractal dimension, average patch size, and spatial complexity (SC). This agrees with a suggestion made by Connor and McCoy [217] that there are possibly multiple non-mutually exclusive causes contributing to species-area relationships.

Table 6-9 provides a list of rules extracted for the simple power function, F1, with respect to nested sampling. The coefficient ‘a’ and the slope ‘b’ (z-value) both vary with respect to fractal dimension (a measure of spatial complexity), tempered by spatial scale. In particular, the value of ‘a’ is low for smaller spatial scales and its value is high for larger spatial scales (see Table 6-9). This makes sense given that in larger land areas there will tend to be more kinds of species present. The value of slope ‘b’ is low when the fractal dimension exceeds its threshold value relative to larger spatial scales (accuracy of 0.90). On the other hand, ‘b’ has a high value when the fractal dimension is below its threshold value relative to smaller spatial scales (accuracy of 0.82). These results once again agree with the experimental data in [236] and [258]. Moreover, there is a consistency between how ‘b’ behaves for F1 and how it behaves for F4 given that its magnitude varies directly with spatial scale. Finally, it is worth noting that the conditions of the extracted rules for F1 (low fractal dimension paired with smaller spatial scale; high fractal dimension paired with larger spatial scale) are realistic since they agree with the observations of

Nams and Bourgeois [260] that in natural habitats, fractal dimension tends to vary directly with spatial scale.

Table 6-9. Extracted rules for F1 Coefficients in nested sampling (t_{fd}, t_{fd*} are threshold values so that $t_{fd}<t_{fd*}$)

Condition	Coefficient	Hit Ratio	Accuracy
(FracDim≤ t_{fd}) and (SScale ≤ LS)	Low *a*	0.38	0.87
(FracDim> t_{fd}) and (SScale > IS)	High *a*	0.25	0.94
(FracDim≤ t_{fd*}) and (SScale ≤ IS)	High *b*	0.32	0.82
(FracDim> t_{fd}) and (SScale >IS)	Low *b*	0.25	0.90

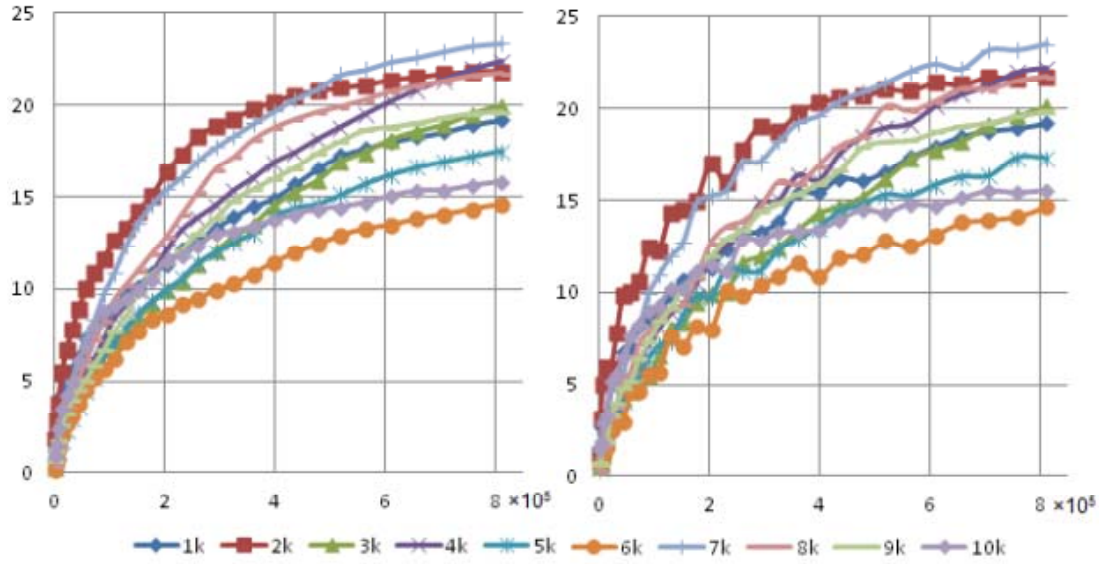
Our results indicate that for the best ranked functions and across the two sampling methods, we were able to determine the meaning of the coefficients with a reasonable degree of accuracy. These results are a significant gain given the relative paucity of articles attempting to elucidate the meaning of coefficients in high performing SAR functions.

6.4.5. Verification of the rules extracted for F1 and its extension F4

To verify the extracted rules for F1 and F4, we performed several regression analyses between spatial features versus the power function family coefficients (For all the regression analysis $p<0.00001$). We found that 'b' had an inverse relation with fractal dimension ($R^2=0.27$), patch number ($R^2=0.25$), and SC ($R^2=0.28$). With higher fractal dimension, the species accumulation rate is faster which leads to lower 'b' [245]. There is a similar scenario when patch number or SC increases. However, 'b' had a direct linear relation with contagion ($R^2=0.3$) and average patch size ($R^2=0.25$). Larger contagion which implies larger patch size, leads to less accumulation of species due to the fact that the individuals in a patch are mostly from the same species given that they tend to group together for reproductive purposes. This results in a higher 'b' value. We also observed a direct linear relationship between 'a' and fractal dimension ($R^2=0.25$). This finding is meaningful since 'a' is an estimation of the average number of species per cell and given that increasing fractal dimension increases the probability of finding new species. The reason that only sampling scale, SC and fractal dimension appeared in the rules is that other features such as patch size are covered by them and so redundant features are discarded by the applied rule extraction method.

6.4.6. Validity of our simulation approach

To investigate the main source of the SAR variation in EcoSim, we plotted the SAR for 10 different time steps of one of EcoSim's runs for two different sampling methods (Figure 6-6). Also, Figure 6-7 depicts the SAR for the nine different runs of the EcoSim at time step 25000 and for two different sampling methods. These figures show that the most important source of variation of the spatial configuration is variation in different runs with the second most important source being different sampling methods. This is another confirmation of the diversity of the world configurations generated by EcoSim even though the initial parameters are fixed.

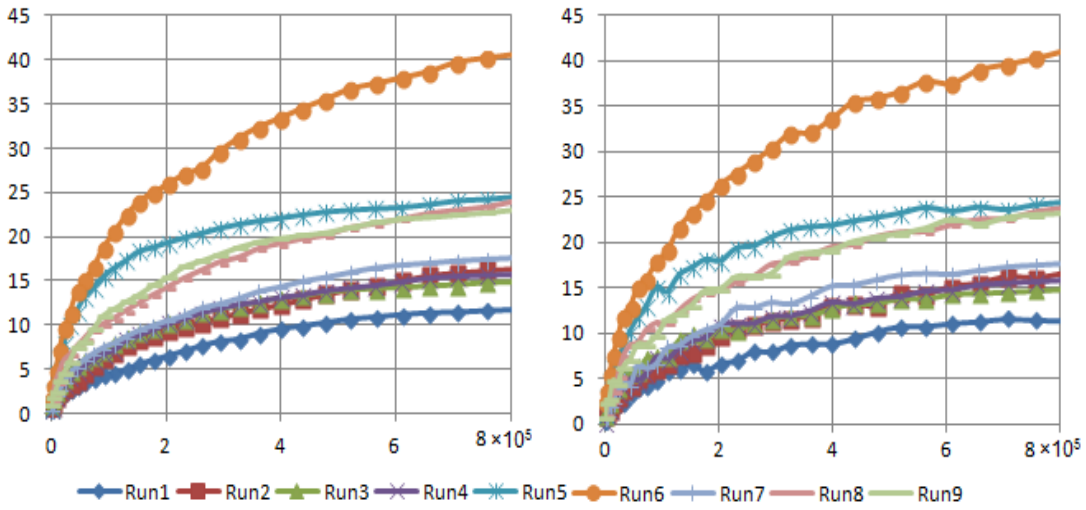


a) Nested Sampling b) Random Sampling

Figure 6-6. Species-Area curves for 10 different time steps, from time step 1000 (1k) to time step10000 (10k), of one of EcoSim's runs for nested and random sampling (The x- axis is the area base on the number of cells and the y-axis is the number of species)

Although EcoSim is a fairly unsophisticated simulation of the real world, it yielded distribution patterns with spiral shapes which have been observed in predator-prey systems [100]. The EcoSim simulation also yielded coherent patches inhabited by species along with the abiotic environment (food pattern) that approximate patterns often observed in nature (see Figure 6-1). The dominance of the power function family over the other models, which corresponds to the situation in many real world communities, is another validation of our simulation. For the simple power function (F1), we obtained a very good fit (R^2_{adj}= 0.98 in average) and z-value ranges from 0.2 to 0.78 which is in the range of z-value in real communities (Table 6-10). To the best of our

knowledge, EcoSim is the largest simulation that has been employed to study SAR models, as there is no limitation to the number of organisms and species in a vast habitat with one million cells.



a) Nested Sampling b) Random Sampling

Figure 6-7. Species-Area curves for nine different runs at time step 25000 for nested and random sampling (The x- axis is the area base on the number of cells and the y-axis is the number of species)

6.5. Conclusions

Using data extracted from EcoSim (to the best of our knowledge, the most complete ecosystem simulation in terms of number of individuals and species, behavioral model and evolution of individuals), our study demonstrated that although there is no unique function that best describes all species-area relationships, functions in the power family, and in particular the Plotkin extended power function (F4) were the best ranked functions. The power function family is suitable, as we observed them always among the six best ranked models in nested sampling. Amongst them, the power function is the simplest model with the fewest coefficients and hence from the point of view of pragmatic parsimony it may be easier to apply the simple power function to the data. However, for more accurate results, a more complicated model such as the Plotkin function may better fit the data.

Table 6-10. z-value of several studies along with z-value for the current study. z-value in EcoSim is in the range of this value in the real communities

z-value	Species	Scale	Reference
0.11 to 0.64	Plants	Continental	[261]
0.21 to 0.41	Birds	Fragmented Forests	[200]
0.12 to 0.35	Plants	Woodlands Sites	[262]
0.07 to 0.48	Plants	Grassland and Forest	[212]
0.17 to 0.64	Birds, Mammals, Amphibians	Continental	[263]
0.36 to 0.67	Trees	Forest	[221]
0.05 to 0.64	Plants	Silwood Park	[264]
0.157 to 0.485	Plants,Vertebrates,Invertebrates,Lichens, Fungi	Islands	[223]
0.2 to 0.78	Prey	EcoSim	Current study

Furthermore, we demonstrated that a number of factors, such as sampling scale and sampling strategies, should be taken into account because they affect the shape of the SAR models. We found different models to be the most suitable function for different sampling methods and sampling scales. Models generating convex curves tended to be more appropriate for small to intermediate scales whereas models generating sigmoid curves tended to be more accurate for larger scales.

We proposed, for the first time, a machine learning approach to discern the meaning of the SAR functions' coefficients by providing several rules associated with their probability of prediction. We were able to determine the meanings of the SAR coefficients from these extracted rules. However, we are not arguing that our interpretations of the coefficients are the only possible interpretations, but merely that they are plausible. We are arguing that we have designed a

method to study and discover the specific meaning of some of these parameters for some specific environmental conditions, and that this approach can be applied to other data sets as well.

Finally, we found that the slope z measuring the rate of species increase for SAR models in the power function family is directly proportional to beta diversity, which suggests that beta diversity and SAR models are to some extent both measures of species richness.

Chapter 7

7. Rule Extraction from Random Forest: the RF+HC Methods

7.1. Introduction

The main machine learning method applied in the previous chapters was rule extraction. To extract the rules, we trained C4.5 algorithm to infer knowledge from data generated by EcoSim. Although constructing a decision tree is simple and the generated rules are typically comprehensible, the accuracy of the generated rules is not good for some data sets. The main reason is that the C4.5 method has lower performance in compare to other classifiers such as SVM, NN, or random forest (RF). One approach is to use an ensemble of decision tree such as RF to improve the accuracy. RF is an ensemble learning method for both classification and regression that constructs and integrates multiple decision trees at training step using bootstrapping. Additionally, it aggregates the outputs of all trees via plurality voting in order to classify a new input. It has few parameters to tune and it is robust against overfitting. It runs efficiently on large data sets and can handle thousands of input variables. Moreover, RF has an effective method for estimating missing data, and has some mechanisms to deal with unbalanced data sets [25]. In some applications, RF outperforms well-known classifiers such as support vector machines (SVMs) and neural networks [131], [132]. Finally, it has exhibited very good performance especially when the number of features is much higher than the number of samples such as in bioinformatics and computational biology data sets [265], [266].

Despite good performance of RF in different domains, its major drawback is that, similar to neural networks and SVMs, it generates a 'black box' model in the sense that it does not have the ability to explain and interpret the model in an understandable form [147], [267] given that it generates a vast number of propositional if-then rules. As a result, ensemble predictors such as RF are very rarely used in domains where making transparent models is mandatory, such as predicting clinical outcomes [267]. In order to overcome this limitation, the hypothesis generated by RF should be transformed into a more comprehensible representation.

In previous years, a high number of rule extraction methods using trained neural networks and SVMs have been published (see [112] for a good survey). Nevertheless, in the case of the RF model, few research projects have been conducted (see the reviews in chapter 3). A procedure for

the interpretation of the RF model is proposed: the RF+HC methods (Random Forest + Hill Climbing). The main idea is that, once the RF is built, a hill climbing algorithm is used to search for a minimum set of rules that has the highest predictive accuracy. The proposed methods can be treated as a decompositional rule extraction approach given that we employed all the generated rules by RF, which are dependent on the number of trees and also the tree structures in the RF.

7.2. RF + HC Methods

As we mentioned in chapter 3, RE can be expressed as an optimization problem and one solution of this problem is to apply heuristic search methods. The main idea of our proposed RE methods is to use hill climbing method to search in a very large search space of the RF rule sets to find a good set of rules, which not only eases comprehensibility but also improves the overall accuracy. These methods overcome the complexity of finding the best rule set, which is an NP-hard problem [142], [143].

In this section, we present our algorithm (Algorithm 1) to extract comprehensible rules from a RF as follows. The algorithm consists of four parts: In the first part, RF is constructed and all the rules in the forest are extracted into the *Rs* set. The second part of the algorithm computes the score of all rules based on the *RsCoverage* matrix; a sparse matrix that shows which rules cover each sample and its corresponding class label. Afterwards, the scores are assigned to the rules in order to control the rule selection process, which can be based on different factors such as accuracy and rule coverage. We used equation (7-1) that has been shown to be a promising fitness function [268]:

$$ruleScore1 = \frac{cc - ic}{cc + ic} + \frac{cc}{ic + k} \qquad (7\text{-}1)$$

In this formula, *cc* (correct classification) is the number of training samples that are covered and correctly classified by the rule. Variable *ic* (incorrect classification) is the number of incorrectly classified training samples that are covered by the rule. Finally, *k* is a predefined positive constant value (in our case k=4, though other values can be used as it is mostly to avoid the denominator becomes zero). This scoring function ensures the retention of the rules with higher classification accuracy and higher coverage and removes the noisy rules. It also reduces the chance of the occurrence of identical values for different rules. Furthermore, it does not ignore the rules of minority classes. Obviously, other fitness measures can be used instead. One possibility would be to employ the rule score based on metrics such as number of features in the extracted rule set and number of antecedents to increase the quality of rules in terms of comprehensibility.

In the third step of the algorithm, a fitness proportionate selection method is used *iniRuleNo* times to generate an initial rule set (*iniRs*) with a probability to select a rule proportional to its score. In order to search the RF rules space, we used the random-restart stochastic hill climbing method, which gives a local optimum point of the search space based on the random start locations. Any other search methods such as simulated annealing, tabu search, genetic algorithm, or any other greedy heuristic methods can be applied instead of HeuristicSearch function in the algorithm. We repeated the search with a predefined maximum number of iterations (*MaxIteration*), each time with a new initial rule set. This can compensate some of the deficiencies in hill climbing due to the randomized and incomplete search strategy [269].

Algorithm 1 RF+HC

```
 Input: trainSet, tetsSet,iniRuleNo, treeNo
Step 1: // Construct Random Forest
RF = trainRF(trainSet, treeNo);
Rs = getAllTerminalNodes (RF);

Step 2: //compute rules coverage
m = size(trainSet);
n = size(Rs);
RsCoverage=zeros(m, n);
foreach sample in trainSet
     foreach rule in Rs
         if match(rule, sample)
      RsCoverage(sample, rule) = class;
         end if
     end for
end for
RScore = ruleScore (RsCoverage);

Step 3: // Repeat the HC method to obtain best rules
iniRs = getRuleSet(RScore, n, iniRuleNo);
impRs = iniRs; bestRs=iniRs;
for i=1 to MaxIteration
      impRs = HeuristicSearch (impRs, RScore);
      if Acc_impRs > Acc_bestRs
          bestRs = impRs;
          impRs = getRuleSet(RScore, n, iniRuleNo);
      end if
end for

Step 4: // calculate the accuracy on test set
calcPerformance (testSet, bestRs);
```

The hill climbing algorithm, searches for the best neighbor, the one that has the highest score, of the current location based on equation (7-1) in the search space and by changing (adding/removing) one rule to the current rule set. For adding/removing a rule, we used the same fitness proportionate selection procedure that was employed for producing the *iniRs*.

First, a rule is selected. If that rule has already been selected it is removed otherwise the rule is added to the current rule set. The hill climbing score function was defined based only on the overall accuracy because the scoring schema of the second step already took into account both rule coverage and rule accuracy. If the new movement in the rule set space improves the score value, that change is retained. Otherwise it is discarded and then another neighbor in the rule space is sought. This means that the proposed method moves towards the first solution that can improve the objective function. We repeat this step for a pre-defined maximum number of iterations (*MaxIteration*). Finally, in the fourth step, we apply the best extracted rule set on the test set to evaluate the generalization ability of the extracted rules.

One of the RF characteristics is that there is no pruning while it is constructed. Therefore, we expect to have long rules (with a large number of antecedents) in the rule set as well as in the extracted rule set using the proposed algorithm. Having long rules damages the interpretability of the model and thus rules' length should be considered in the applications for which the interpretation of the rules is important. Therefore, we proposed a second algorithm similar to Algorithm 1, except that a modified version of the rule score function (i.e., equation 7-2) was used, where *rl* shows rule length or number of antecedents. We called the new method RF+HC_CMPR (i.e., RF+HC method with an emphasis on comprehensibility). In the RF+HC_CMPR method more generalized rules (shorter length rules with higher accuracy) have higher priority than the more specialized rules (the longer rules with lower accuracy) based on the following equation:

$$ruleScore2 = ruleScore1 + \frac{cc}{rl} \qquad (7\text{-}2)$$

The inputs of the proposed methods are the training/test sets, initial number of rules (*iniRuleNo*) and the number of trees in the RF (see Algorithm 1). Variable *iniRuleNo* adjusts the tradeoff between accuracy and comprehensibility. In cases where prediction ability is important, higher values are used and in cases where the interpretation of the underlying model is important lower values should be used. For the implementation, in order to make fair comparison with the CRF method, we used Matlab as the source code available for the CRF method is also in Matlab.

7.3. Experiments and Discussion

To evaluate our proposed methods, we applied CRF [155], [156] and RF on 22 different data sets. Different criteria have been proposed to evaluate a RE algorithm [270]. Accuracy is defined as the ability of extracted rules to predict unseen test sets and fidelity indicates how similar the RE output is to the underlying model output. In other words, it expresses the percentage of instances classified identically to the underlying model. Another major factor is comprehensibility, which is not easy to measure due to the subjective nature of this concept. Prior domain knowledge also plays an important role in comprehensibility. There are different factors that are used to determine comprehensibility such as, the number of rules and the average number of antecedents. Another desirable characteristic of a RE method is its potential to be applicable to a wide range of applications. If a RE algorithm is applicable to data sets with a large number of samples, features, or classes then it is said to be scalable. It is obvious that both running time and algorithm complexity are inherent in the notion of scalability. Finally, consistency shows the ability of RE to produce similar rules each time it is applied to the same data set, although there are different meanings for similarity itself.

In our work, we measured the average accuracy of 10 times 3-fold cross-validation for evaluating accuracy. We used three folds given that we had some data sets with small numbers of samples, although we repeated it 10 times. This measure demonstrates the prediction and generalization ability of the extracted rules. Majority voting is used to classify a sample when more than one rule covers a sample. We assumed a default rule such that the samples not covered by any of the extracted rules are simply assigned to the high frequency class in the dataset. In the RF+HC methods, due to their stochastic nature, we repeated the whole procedure 10 times and computed the average results along with their standard deviations. For the CRF method, 10 different values for the lambda parameter (which indicates the tradeoff between the number of rules and accuracy) were used. To determine these values, we did a few pilot runs with each data set separately. To determine the best lambda, a cross-validation step is incorporated in the CRF method such that it selects the lambda value, which gives the minimum error for cross-validation. Therefore, in CRF, at the cross-validation step, the best lambda is selected and then it is used in the training step.

In order to show the comprehensibility of the methods, we considered the number of rules, maximum rule length, and total number of antecedents in the extracted rule set. For the CRF method, these values are related to the lambda parameter value, which gives the lowest cross-

validation error. On the other hand, those of RF+HC methods are related to 10 repetitions of the process. All the values are rounded to the closest integer value.

Scalability is one of the most important evaluation metrics often overlooked in most of the RE methods such as the CRF method. To be realistic, the RE method should be applicable to different problems and data sets with a variety of characteristics. To have a fair comparison, we used 10 different lambdas in the CRF method and we divided the required time to find the best lambda by 10. This means that we did not count the time required for examining all the 10 lambda values and we only considered the time for cross-validation using the best lambda plus the time for training and test steps. We considered the cross-validation time because the CRF method is an iterative method involving feature selection and optimization. At each iteration, the features in the extracted rules are kept and the rest are removed. For the new data set obtained from previous step, different lambda values can lead to the higher accuracy as some features were removed in the previous step. Therefore, finding the best lambda using cross-validation at each iteration is crucial in this method. We measured the computational time as a metric to evaluate the scalability. For RF+HC, we repeated each experiment 10 times and then divided the overall time by 10. We also considered the hill climbing repetition time (*MaxIteration*) in order to calculate the computation time. Because we used a collection of data sets with different characteristics, we were able to see the scalability of our methods in different situations.

For this study, we did not consider fidelity because it is not a suitable measure in some RE methods, especially for the ensemble methods where the accuracy of the extracted rules is in some cases higher than those of the underlying models [114]. Similarly, we did not use consistency because there is no clear definition of similarity between rules. Moreover, for the RE methods based on the heuristic methods such as genetic algorithm or hill climbing, it is very hard to guarantee that consistent rules are generated at every step [268], [271]. Therefore, we compared the accuracy for RF+HC, RF+HC_CMPR, CRF, and RF. We also reported the number of rules, maximum rule length, and total number of antecedents in the extracted rule set in addition to the computational time.

As the input of the proposed algorithm, we should specify the initial number of rules (*iniRuleNo*) for each data set. We obtained these numbers using a couple of pilot tests. We used 500 decision trees to build a RF with $m = \sqrt{n}$, which is the most frequently used default value in the literature. In the random-restart hill climbing, we repeated hill climbing from 10 initial rule sets. We took MaxIteration = 500 in all of our experiments. Higher values provide hill climbing with more

opportunities for improving the rule set score, although it did not happen in our case. For comparing the proposed methods with CRF in terms of performance, comprehensibility, and computation time complexity, we used Wilcoxon [272] and Friedman [273] tests as suggested in [274].

7.3.1. Data sets

We used 22 data sets with various characteristics in terms of the number of features, the number of samples, and the number of classes to observe how the performance of the proposed methods varies depending on the data set type. Eighteen data sets were taken from UCI machine learning repository [31] and other four data sets are gene expression microarray data sets, Golub [275], Colon [276], Nutt [277], Veer [278]. The extreme cases are Veer with 24188 features, Magic with 19020 samples, and Yeast and Cardio with 10 classes (Table 7-1).

Table 7-1. Data sets along with their characteristics

DATA SET	FEATURES	CLASS	SAMPLE
BREAST CANCER	9	2	699
MAGIC	10	2	19020
MUSK CLEAN1	166	2	476
WINE	13	2	178
WINE QUALITY	11	6	1599
IRIS	4	3	150
YEAST	8	10	1485
CARDIOGRAPHY	20	10	1726
BALANCE SCALE	4	3	625
CMC	9	3	1473
GLASS	9	6	214
HABERMAN	3	2	306
IONO	34	2	351
SEGMENTATION	19	7	210
TAE	5	3	151
ZOO	16	7	101
ECOLI	7	8	336
SPAM	57	2	4601
GOLUB	5147	2	72
COLON	2000	2	62
Glimo NUTT	12625	2	50
VEER	24188	2	77

7.3.2. Accuracy and Generalization Ability

On average, both the RF+HC and RF+HC_CMPR methods gave almost the same level of accuracy as the CRF method with marginal differences (Table 7-2). Moreover, all three methods obtained 96% of the RF accuracy for the whole data sets on average. For some datasets, they demonstrated higher accuracy than RF such as Tae, Cmc, and Golub with RF+HC and Tae and Clean with CRF method.

Table 7-2 Percentage accuracy of the RF+HC, RF+HC CMPR, CRF, and RF methods on the selected data sets

Data set	RF+HC	RF+HC_CMPR	CRF	RF
Cancer	96.18 (0.32)	96.23 (1.56)	95.71 (1.01)	**96.65 (1.75)**
Magic	85.37 (0.46)	85.6 (0.28)	83.65 (1.3)	**88.12 (0.3)**
Clean	81.34 (3.25)	83.17 (4.3)	88.45 (1.55)	**88.68 (2.18)**
Wine	92.07 (3.29)	95.93 (1.8)	91.93 (5.91)	**98.99 (0.9)**
Wineqlty	65.13 (1.93)	62 (1.8)	62.79 (0.57)	**68.59 (3.47)**
Iris	93.36 (2.4)	94.12 (3.25)	94.4 (2.61)	**96.40 (1.67)**
Yeast	59.98 (1.5)	61.3 (0.7)	55.02 (2.75)	**62.02 (1)**
Cardio	81.74 (0.82)	82 (0.6)	84.01 (0.84)	**85.67 (2.19)**
BalancS	84.48 (0.52)	83.75 (2.36)	82.87 (2.86)	**87.24 (1.6)**
Cmc	**52.87 (0.99)**	52.6 (2.5)	49.42 (3.65)	52.46 (2.57)
Glass	74.33 (2.7)	73.75 (7.3)	72.77 (2.15)	**78.02 (7.51)**
Haber	67.69 (2.1)	69.14 (1.7)	70.2 (4.42)	**73.92 (4.2)**
Iono	90.14 (3.53)	91.9 (3.3)	91.45 (1.6)	**93.16 (1.9)**
Segment	87.54 (1.86)	89.97 (2.4)	88.86 (3.7)	**93.14 (2.1)**
Tae	**57.60 (3.46)**	53.45 (4.1)	62.29 (4.8)	55.60 (1)
Zoo	91.33 (9.6)	92.96 (5.87)	93.94 (8.2)	**97.02 (2)**
Ecoli	84.2 (3.11)	79.9 (4)	86.67 (11.54)	**86.96 (1.74)**
Spam	94.04 (0.71)	94.33 (0.5)	94.2 (1.05)	**95.24 (0.3)**
Golub	**93.00 (6.7)**	87.25 (7.6)	86.11 (9.62)	92.5 (4.5)
Colon	74.76 (5.26)	76.1 (3.9)	**82.46 (17.94)**	75.00 (11.85)
Glimo	64.11 (4.26)	66.3 (7.36)	54.9 (8.99)	**71.69 (14.47)**
Veer	58.27 (7.88)	63.11 (8)	60.97 (8.99)	**66.43 (13.76)**

Table 7-3 Each cell shows 'Number of extracted rules / Maximum length of rule / Total number of antecedents' in each method. The values in bold show the best results.

Data set	RF+HC	RF+HC_CMPR	CRF	RF
Cancer	36 / 8 / 159	**33** / **6**/ **129**	463 / 9 / 1940	12075 / 13 / 65869
Magic	2604 / 8 / 8186	**2597** / **3** / **5697**	3182 / 37 / 50668	608155 / 58 / 8514170
Clean	83 / 15 / 586	**78** / **10** / **473**	104 / 18 / 947	18392 / 20 / 150309
Wine	16 / 8 / 64	**14** / **5** / **55**	176 / 7 / 619	7590 / 10 / 26784
Wineqlty	**1258** / 21 / 12301	1259 / **12** / **10526**	2282 / 24 / 22256	138889 / 30 / 1757860
Iris	13 / 6 / 39	**11** / **5** / **28**	43 / 5 / 145	4202 / 9 / 13222
Yeast	**1037** / 25 / 13460	1303 / **13** / **11621**	1836 / 27 / 18430	126936 / 32 / 1469328
Cardio	1609 / 20 / 15720	**1606** / **11** / **12951**	2121 / 20 / 19003	126412 / 22 / 1150839
BalancS	88 / 9 / 471	**83** / **5** / **339**	360 / 11 / 1768	19764 / 13 / 124447
Cmc	332 / 16 / 2390	**322** / **10** / **1818**	2025 / 19 / 14695	74257 / 22 / 754197
Glass	88 / 13 / 398	**59** / **8** / **335**	10050 / 12 / 30662	16530 / 16 / 115932
Haber	28 / 13 / 165	**25** / **8** / **140**	410 / 16 / 2417	19697 / 18 / 142512
Iono	41 / 11 / 193	**36** / **7** / **145**	155 / 12 / 784	10641 / 14 / 57312
Segment	**42** / 10 / 267	54 / **6** / **175**	11134 / 13 / 24065	9905 / 12 / 59837
Tae	91 / 13/ 495	**76** / **8** / **359**	177 / 13 / 997	14437 / 16 / 93530
Zoo	16 / 6 / 66	**15** / **4** / **51**	185 / 7 / 608	4954 / 9 / 17615
Ecoli	**138** / 11 / 762	141 / **7** / **649**	8900 / 14 / 29421	16761 / 16 / 105260
Spam	476 / 34 / 5076	**473** / **21** / **4228**	1154 / 41 / 14852	118878 / 44 / 1455859
Golub	9 / 3 / 18	6 / **2** / 10	**1** / 3 / **3**	2322 / 4 / 4939
Colon	**17** / 4 / 46	19 / **3** / **39**	27 / 5 / 85	2620 / 6 / 8154
Glimo	**9** / 3 / 23	12 / **2** / **20**	17 / 4 / 47	1953 / 4 / 4716
Veer	18 / 4 / 45	**17** / **3** / **33**	39 / 5 / 128	3254 / 6 / 8513

A similar result was observed in [114] when the authors used a neural network ensemble to extract the rules, observing higher accuracy for extracted rules than for the underlying model. The generalization ability of RF+HC depends on the selection of the high score rules in RF and depends on some probabilistic selection of rules with low scores in the training set, but which may be important for unseen data.

Comparing the accuracy of CRF method with the proposed methods revealed that the null hypothesis with $\alpha = 0.05$ cannot be rejected with $z = 0.41$ (CRF vs. RF+HC) and $z = 0.42$ (CRF vs. RF+HC_CMPR), while the critical z value is -1.96 in Wilcoxon test. Therefore, the difference is not significant, which proves that all methods are equivalent in terms of accuracy.

7.3.3. Comprehensibility

Although a feature selection phase was incorporated in the CRF method, our methods were superior in the number of extracted rules in all the data sets except the Golub data set (Table 7-3). The number of rules extracted by RF+HC or RF+HC_CMPR on average are 0.6% of the total number of rules in RF while that of CRF is 11.66%, which demonstrates an impressive improvement in compare to RF and CRF. The proposed methods significantly reduced the number of rules compared to CRF (z=-4.06$) and consequently improved the comprehensibility. However, the difference in terms of rule numbers for the two proposed methods was not significant (z=-1.89). There is one dataset (Golub) for which CRF extracted only one rule. In such cases, the extracted rule is related to one class and it can only explain that class. However, there is no information and interpretation regarding the other class(es). Therefore, we believe that this type of rule is not fully comprehensible as they cannot describe the underlying model completely. We found an issue in the implementation of the CRF method, which will affect the results. When the number of rules is reported, only the rules with the weights greater than a threshold (in this case 10e-6) are considered. However, all the extracted rules are used to do prediction for the test set. It means that the reported number of extracted rules is not correct. The CRF results in Table 7-3 correspond to the correct number of rules.

We used the modified version of the rule score function (i.e., equation 7-2) in order to give higher priority to the more generalized rules. Table 7-3 shows the comparison between the original algorithm and RF+HC_CMPR. The results showed that RF+HC_CMPR have a stronger impact on the maximum rule length and also on the total number of antecedents (42% and 18% decrease respectively) in the rule set in comparison with RF+HC. In addition, we observed no significant change in the accuracy. These results indicate that RF+HC_CMPR improves the comprehensibility significantly (z=-4.16).

Comparing the CRF method with the two proposed methods using Wilcoxon test (critical z=-1.96) indicates that RF+HC had a significantly lower maximum rule length (z=-3.13) and also number of antecedents (z=-4.07) compared to CRF. RF+HC_CMPR was superior in all data sets in terms of maximum rule length (z=-4.09) and number of antecedents (z=-4.07) except for the maximum rule length for Golub.

One important aspect of comprehensibility is the number of rules extracted from an underlying model. However, we have to consider the importance of the tradeoff between accuracy and comprehensibility. The extracted rules should not only be concise but also have good

performance on unseen samples. This is, in fact, the main objective of rule extraction. Therefore, a good rule extraction method should consider two facts simultaneously: comprehensibility and generalization ability, although it should be adjustable based on the application. More complex datasets will decrease one of them. For example, for the Magic dataset, RF generates 608155 rules with approximately 88% accuracy. This number of rules shows the complexity of the model for this dataset. RF+HC methods extract only about 0.4% of the RF rules and give about 85% accuracy for this data set. We still can generate fewer rules by decreasing *iniRuleNo*, although it will reduce accuracy. Therefore, what needs to be considered in order to have a fair judgment is the combination of the number of rules and accuracy. The results we have presented here correspond to the smallest number of rules in order to achieve a level of accuracy as close as possible to the level of accuracy for RF.

7.3.4. Complexity and Scalability

We found a significant difference in terms of computational time between our methods and CRF (z=-4.07). For all data sets, the RF+HC methods were faster than CRF with the exception of the Iris data set, which had only a one-second difference (Table 7-4).

More specifically, in some cases with large numbers of classes such as Yeast, Glass, Ecoli, and Segment, our methods were 136, 310, 518, and 842 times faster than CRF respectively. We observed the same phenomenon for data sets with a large number of samples such as Magic, Spam, and CMC where RF+HC and RF+HC_CMPR were 13, 18, and 130 times faster than CRF. On average, the overhead time for the proposed methods and CRF method was 1.12, and 11.8 times respectively relative to RF time.

Moreover, we observed more computational time for CRF when there was a larger number of classes (Table 7-4) because the CRF method considers c classifiers (c being the number of classes) and finds a weight vector for each class. When there are a relatively large number of samples and a large number of classes simultaneously, the CRF method has an even worse performance. In addition, a large number of features can increase the computational time as CRF has a repeating feature selection step and in each step a new RF is built, the best lambda is found and then a new optimization problem is solved. However, RF+HC methods are based on a unique RF where there is a fixed amount of time for building the RF. The overhead time on top of RF in RF+HC methods has a strong linear correlation with the number of samples in the data sets (R2=0.994). Therefore, it can be said that the complexity of the proposed algorithm is O(N), where N is the number of samples in the data set.

Table 7-4 Computational time for RF+HC, RF+HC_CMPR, CRF, and RF in seconds

Data set	RF+HC	RF+HC_CMPR	CRF	RF
Cancer	16	16	36	5
Magic	1409	1425	19338	1050
Clean	34	34	118	26
Wine	4	5	13	1
Wineqlty	52	56	5317	17
Iris	4	9	3	1
Yeast	46	49	6276	15
Cardio	80	83	6410	31
BalancS	17	17	233	4
Cmc	36	36	4696	14
Glass	5	5	1551	1
Haber	10	10	15	2
Iono	9	9	24	3
Segment	7	7	5900	2
Tae	5	5	14	1
Zoo	3	3	14	1
Ecoli	9	9	4669	2
Spam	236	239	4479	166
Golub	230	230	253	228
Colon	56	56	62	54
Glimo	633	633	720	631
Veer	3165	3165	3558	3162

7.4. Overall Comparison and Major Contributions

The major contributions of both proposed methods in comparison to RF are that they refine RF in selecting the most valuable rules, which leads to a huge decrement in the number of rules i.e. 0.6% of the random forest rules, while at the same time attaining 96% of the RF accuracy with a reasonable overhead time on top of RF time. In addition, both methods improved the comprehensibility in comparison with CRF while retaining the same accuracy. RF+HC decreased the number of rules, the maximum rule length, and the total number of antecedents by 27%, 16%,

and 49% respectively in average. RF+HC_CMPR also reduced them by 25%, 50%, and 59%. The RF+HC methods decreased the computational time in 21 of the 22 data sets. Moreover, for the data sets with a large number of samples and/or a large number of classes, they were much faster (up to about 800 times).

Table 7-5 summarizes the overall comparisons of RF+HC and RF+HC_CMPR with the CRF method. The numbers in the table specify the average rank of each method for Friedman test computed for the mentioned criteria in the table, where lower value demonstrates the better method. The Friedman test showed significant difference between the average ranks and the mean rank for each criterion. However, the difference was marginal for the accuracy as it was also confirmed by the Wilcoxon test. These results show that our proposed methods are better than CRF in terms of number of rules, computational time, maximum rule length, and also number of antecedents while they keep level of accuracy as the same as CRF method.

Table 7-5. Comparison summary for different methods. The values are the average rank with the standard deviation in the parenthesis

	RF+HC	**RF+HC_CMPR**	**CRF**
Accuracy	2.23 (0.81)	1.73 (0.7)	2.05 (0.9)
Rule#	1.77 (0.53)	1.32 (0.48)	2.91 (0.43)
Time	1.34 (0.24)	1.7 (0.37)	2.95 (0.21)
MaxCond	2.11 (0.26)	1.02 (0.11)	2.86 (0.35)
Cond#	2 (0)	1 (0)	3 (0)

7.5. Conclusions

We introduced new rule extraction methods derived from a RF: RF+HC. Once the RF is built, a hill climbing algorithm is used to search for a rule set that has high predictive accuracy with a drastic reduction in the number of rules compared to RF. In addition, our methods are much more scalable than the state-of-the-art method, CRF. Experimental results showed that these methods are superior to the CRF method in terms of comprehensibility as they generate fewer and shorter rules while keeping the same level of accuracy. Finally, both are much more scalable than the CRF method and it can be applied more generally and on data sets with various characteristics.

Chapter 8

8. Conclusion and Future Works

Understanding and investigating natural systems in the real world is always challenging for scientists due to the complexity involved in such systems. Among them, those related to biology and ecological phenomena are absolutely fascinating even though very difficult to understand. However, during recent decades new theoretical approaches, such as artificial life systems and artificial intelligence methods, have emerged that bring promising capabilities to investigate them. With increases in computational power, it is possible to make complex artificial life systems to simulate natural phenomena. More specifically, individual-based modeling is one approach for understanding the behavior of complex ecosystems. It is a bottom-up approach allows considering the traits and behavior of individual organisms and resulting in the emergence of some high level phenomena, outcome of the whole interactions. Simulating the simple and general interaction rules of real ecosystems creates an artificial ecosystem with patterns similar to what are observed in nature.

However, due to the multiple interactions between individuals, such artificial systems have strongly emerging non-linear behaviors. Understanding of such systems is still challenging as they can generate vast amount of data related to every single components of the simulation. Therefore, data analysis plays an important role in order to turn the generated raw data into insight. Artificial intelligent methods, and more specifically machine learning, is one of the most popular methods in data analysis. They are able to extract useful knowledge, suggesting conclusions, and helping decision-making by learning from the input raw data. Regression, classification, feature selection, rule extraction are examples of machine learning methods that can be used for this purpose. Applying these types of methods helps to achieve two important aims. First, knowledge can be inferred from the generated raw data which can result in new insights about a phenomenon. Second, the conformity of the inferred knowledge can be verified by the real ecosystems.

Biologists and ecologists can barely study many difficult evolutionary or ecological questions only by studying real ecosystems because, most of the time, there is not enough data available, or it is a very time consuming and expensive task to perform an experiment. We employed EcoSim, a generic complex simulation platform, to investigate several ecological questions, as well as long-term evolutionary patterns and processes such as speciation and extinction of species. The major difference between EcoSim and the classic modeling approaches is that classic ecological

modeling is based on a pre-defined fitness functions. This causes a bias as the decisions made by individuals with distinct behavioral models rely on an external evaluation (pre-defined fitness function) and is therefore not an emerging property. To avoid such bias a complex system in which fitness emerges from the multiple interactions between numerous individuals is needed.

During my PhD study, three major phenomena related to the species have been investigated. The main reasons behind speciation and extinction of species are among the challenging problems for biologists. In addition, there is debate regarding the best function to describe the species-area relationship and also its coefficients interpretation. The first study was to investigate the ability of spatial and spatiotemporal information about species in an artificial ecosystem for the prediction of speciation events. We used various measures to extract this type of features and we use them to predict speciation. We obtained good prediction results showing that spatial distribution information of species effectively predict speciation events. Our results are confirmed with the real field studies showing that the geographical and spatial distribution of individuals in one species is a leading phenomenon for speciation. Our results also indicate that some generic traits exist in our simulation that characterizes the speciation events. In another experiment, we investigated how various demographics, genetics, environment and spatial distribution features can predict speciation. We obtained very good accuracy for the prediction that show the calculated features are effective in prediction of the speciation and can help for better understanding of the speciation. We extracted several simple rules from the constructed decision tree. These rules are semantically clear and sound reasonable based on biological evidences. This is an important result as the proposed approach has proven to have the capability of generating realistic rules when compared with real biological data.

In third study, we investigated three broad categories of genetic, environmental, and demographic features associated with species extinction in EcoSim. We obtained a rule set for each category and showed that these rules can predict extinction in the next 100 time steps with a very high level of accuracy. We also demonstrated that these rules are generic by applying the constructed predictive model on completely different simulation runs. The acquired results suggest how accurate our proposed machine learning approach can be from several different perspectives. First, the proposed approach is able to extract important features related to extinction effectively, especially when there is a plethora of features and there is no exact knowledge about them. Second, the categorization idea helps to study the effect of features in a more fine-grained way and to extract rules associated with them accompanied by an evaluation of their accuracy. This may prove to be beneficial for conservation biologists for being able to detect early signals of

extinction. For example, we found that population size of the species and also average genetic distance of parents at breeding time in one species are really important features as we were able to predict extinction using those features alone with a high accuracy comparable to the accuracy level obtained when we used all the features in each categories such as genetic, environmental, and demographic. This is particularly useful for obtaining a high level of predictive accuracy based on a minimum amount of information from the environment. Further, this approach can be applied to test new hypotheses regarding new factors involved in extinction. While our results are not directly valid for real situations given that our model involves a high level of abstraction as well as being a simplification of the real world, our results provide interesting insights that could be of aid to biologists in formulating and testing new hypotheses relating to species extinction. Finally, the approach we have employed has the potential to be useful for more dedicated studies focusing on species extinction. Also to be noticed is the general innovation of providing a methodology for ecological data analysis based on machine learning techniques.

In other study, using data generated by EcoSim, we showed that although there is no unique function that best describes all species-area relationships, functions in the power family, and in particular the Plotkin extended power function were the best ranked functions. The power function family seems to be the most suitable set of functions, as we observed them always among the six best ranked models in nested sampling. Amongst them, the power function is the simplest model with the fewest coefficients and hence, from the point of view of pragmatic parsimony, it may be easier to apply the simple power function to the data. However, for more accurate results, a more complicated model such as the Plotkin function may better fit the data. Furthermore, we demonstrated that a number of factors, such as sampling scale and sampling strategies, should be considered because they affect the shape of the SAR models. We found different models to be the most suitable function for different sampling methods and sampling scales. Models generating convex curves tended to be more appropriate for small to intermediate scales whereas models generating sigmoid curves tended to be more accurate for larger scales. We proposed, for the first time, a machine learning approach to discern the meaning of the SAR functions' coefficients by providing several rules associated with their probability of prediction. We also were able to determine the meanings of the SAR coefficients from these extracted rules. However, we are not arguing that our interpretations of the coefficients are the only possible interpretations, but merely that they are plausible. We also are arguing that we have designed a method to study and discover the specific meaning of some of these parameters for some specific environmental conditions, and that this approach can be similarly applied to other data sets.

Finally, we found that the slope z measuring the rate of species increase for SAR models in the power function family is directly proportional to beta diversity, which suggests that beta diversity and SAR models both are, to some extent, measures of species richness.

Complex systems, such as EcoSim, generate a huge amount of data. To be able to answer theoretical question using such systems, efficient methods for data analysis and knowledge extraction are also inevitable. We used various machine learning techniques to analyze the data generated by the EcoSim experiments. Our objective was to conduct a robust test to prove the effectiveness of our framework for identifying reasons behind different theoretical ecological phenomena such as speciation, extinction, and SAR. By interpreting the obtained models we were able to extract meaningful rules to enrich our knowledge about such phenomena. We also showed that machine learning techniques are particularly efficient to analyze such data bringing semantically interpretable rules with high predictive accuracy and therefore these techniques should be extended and considered as important tools for future theoretical or empirical studies. More specifically, the role of rule extraction was prominent when some explanations for the prediction result of the predictive model are needed. For our case we used decision tree to be able to extract the rules explaining the prediction results. The problem with decision tree is that in general it has lower performance compared to other predictive models such as SVMs and ensemble methods. However, those methods are not interpretable and cannot easily explain their prediction results. As a result, we introduced new rule extraction methods from random forest. The proposed methods search for a rule set that has high predictive accuracy with a drastic reduction in the number of rules compared to RF. In addition, our methods are much more scalable than the state-of-the-art method, CRF. Experimental results showed that the proposed methods are superior to the CRF method in terms of comprehensibility as they generate fewer and shorter rules while keeping the same level of accuracy. Finally, the proposed methods are much more scalable than the CRF method and it can be applied more generally and on data sets with various characteristics.

All the research studies on EcoSim showed promising results, that they brought new insights in the ecology field and also the obtained results are plausible compared to the real nature. We showed that the speciation and extinction mechanisms cohere with the empirical studies based on the real ecosystems. In addition, we showed that the best SAR model in EcoSim is power function family as so in many of the research studies based on natural ecosystems. We also demonstrated that coefficients of the SAR model have meaning associated to some ecological factors. Therefore, our studies confirm the usefulness of EcoSim as a generic platform to study

broad evolutionary ecological phenomena. EcoSim provides opportunity for ecologist that besides the field studies, they can also benefit from EcoSim by conducting more in depth experiments related to the species specially when either not enough information is available or it is very time consuming and expensive to collect data. Different types of speciation mechanism can be implemented and tested. More in depth studies on species extinction to investigate the effect of factors such as initial population size, Allee effect, predator pressure, resource depletion, and catastrophe events can be performed, most of the times with few modifications in the simulation. It is also possible to construct a more specialized version of EcoSim to study a specific type of species by providing real data parameters. It is possible to easily conduct studies in different granularity levels such as individual level, species level, or the whole population in the ecosystem. In all such types of studies, data analysis has a prominent role. One of the effective analysis methods is machine learning that we showed some of their important aspects in this thesis. These types of analysis help biologists and ecologists to discern new insights and facts about different phenomena, especially when there are a lot of factors involved and there is not enough knowledge about their impact on such phenomena, which in turn can bring about new hypothesis and ideas about them. Therefore, a mixture of field studies, artificial life, and artificial intelligence is a very powerful research portfolio for ecologists and biologists.

In terms of data analysis, only having the prediction results are not enough in some domains, especially when we need to know the reasons that lead to such results. For example, even though prediction of, for example, breast cancer is very useful, knowing why it happens is much more important. Because having the clear reasons makes it possible to prevent the diseases instead of just curing them. Therefore, in parallel with devising new improved prediction methods, more advanced rule extraction methods are also necessary. More specifically, rule extraction methods that are able to extract comprehensible yet accurate rules from big data are desired.

As a result and for future works, different ecological studies can be conducted using EcoSim such as studying the relationship between learning and evolution. For this purpose, new features and concepts should be added to the platform, for example learning capability should be implemented for the individuals. Another possibility is to implement speciation based on the various definitions existing in the literature and then study their impact on the results. We found rule extraction as a very useful tool to interpret the results of a predictive model, which is essential in many fields of study such as health, finance, etc. We proposed new rule extraction methods using a simple heuristic search method and we obtained promising results. Those methods can be extended in several ways. For instance, other advanced search methods such as

genetic algorithm or tabu search can be applied to find the best set of rules. In addition, different score functions can be tested for rule selection procedure. Investigating dimensionality reduction methods such as factorization methods to reduce the dimension of the *RsCoverage* matrix, described in chapter 7, is another direction for the future work.

Appendix A

Copyright Permissions

1- Robin Gras

I give permission to include materials for the papers presented in chapters 4 to 7 of Morteza Mashayekhi's dissertation.

2- Brian McPherson, Abbas Ghadri, Marwa Khater, Yasaman Farahani, Meisam Hosseini

I do give permission to include materials for the papers presented in chapters 2, 5, and 6 of Morteza Mashayekhi's dissertation.

REFERENCES / BIBLIOGRAPHY

[1] M. A. Bedau, "Artificial life: organization, adaptation and complexity from the bottom up," *Trends Cogn. Sci.*, vol. 7, no. 11, pp. 505–512, 2003.

[2] C. Vidal, "The Beginning and the End: The Meaning of Life in a Cosmological Perspective," *arXiv Prepr. arXiv1301.1648*, 2013.

[3] J. R. Krebs and N. B. Davies, *Behavioural ecology: an evolutionary approach*. John Wiley & Sons, 2009.

[4] D. W. Stephens, *Foraging theory*. Princeton University Press, 1986.

[5] A. K. Seth, "The ecology of action selection: Insights from artificial life," *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 362, no. 1485, pp. 1545–1558, 2007.

[6] M. Mashayekhi, A. Golestani, Y. M. F. Farahani, and R. Gras, "An enhanced artificial ecosystem: Investigating emergence of ecological niches," in *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, vol. 14, pp. 693–700.

[7] C. Ricotta, "From theoretical ecology to statistical physics and back: self-similar landscape metrics as a synthesis of ecological diversity and geometrical complexity," *Ecol. Modell.*, vol. 125, no. 2, pp. 245–253, 2000.

[8] D. Devaurs and R. Gras, "Species abundance patterns in an ecosystem simulation studied through Fisher's logseries," *Simul. Model. Pract. Theory*, vol. 18, no. 1, pp. 100–123, Jan. 2010.

[9] G. Keppel, *Data analysis for research designs*. Macmillan, 1989.

[10] C. O'Neil and R. Schutt, *Doing Data Science: Straight Talk from the Frontline*. " O'Reilly Media, Inc.," 2013.

[11] C. M. Bishop and others, *Pattern recognition and machine learning*, vol. 1. springer New York, 2006.

[12] R. Gras, D. Devaurs, A. Wozniak, and A. Aspinall, "An Individual-Based Evolving Predator-Prey Ecosystem Simulation Using Fuzzy Cognitive Map as Behavior Model," *Artif. Life*, vol. 15, no. 4, pp. 423–463, 2009.

[13] R. Gras, A. Golestani, M. Hosseini, M. Khater, Y. M. Farahani, M. Mashayekhi, S. M. Ibne, A. Sajadi, E. Salehi, and R. Scott, "EcoSim: an Individual-Based Platform for Studying Evolution," 2011, pp. 284–286.

[14] R. J. Safran and P. Nosil, "Speciation: The origin of new species," *Nat. Educ. Knowl.*, vol. 3, no. 10, p. 17, 2012.

[15] E. Mayr and others, "Animal species and evolution.," *Anim. species their Evol.*, 1963.

[16] M. Mashayekhi and R. Gras, “Investigating the Effect of Spatial Distribution and Spatiotemporal Information on Speciation using Individual-Based Ecosystem Simulation.,” *GSTF J. Comput.*, vol. 2, no. 1, 2012.

[17] M. G. Ritchie, “Sexual selection and speciation,” *Annu. Rev. Ecol. Evol. Syst.*, vol. 38, pp. 79–102, 2007.

[18] B. D. Griffen and J. M. Drake, “A review of extinction in experimental populations.,” *J. Anim. Ecol.*, vol. 77, no. 6, pp. 1274–87, Nov. 2008.

[19] M. H. Sedehi, R. Gras, and M. Sina, “Prediction of Imminent Species’ Extinction in EcoSim.,” in *ICAART (1)*, 2012, pp. 318–323.

[20] U. K. Rai, “Minimum sizes for viable population and conservation biology,” *Our Nat.*, vol. 1, no. 1, pp. 3–9, 2003.

[21] D. H. Reed, E. H. Lowe, D. A. Briscoe, and R. Frankham, “Inbreeding and extinction : Effects of rate of inbreeding,” *Conserv. Genet.*, vol. 4, no. 3, pp. 405–410, 2003.

[22] M. V. Lomolino, “The species-area relationship: new challenges for an old pattern,” *Prog. Phys. Geogr.*, vol. 25, no. 1, pp. 1–21, Jan. 2001.

[23] E. Tjørve, “Shapes and functions of species-area curves: a review of possible models,” *J. Biogeogr.*, vol. 30, no. 6, pp. 827–835, Jun. 2003.

[24] J. Dengler, “Which function describes the species-area relationship best? A review and empirical evaluation,” *J. Biogeogr.*, vol. 36, no. 4, pp. 728–744, Apr. 2009.

[25] L. Breiman, “Random forests,” *Mach. Learn.*, pp. 1–33, 2001.

[26] M. Mashayekhi and R. Gras, “Speciation Prediction based on Spatial Distribution and Spatiotemporal Information from an Individual-Based Ecosystem Simulation,” in *2rd Annual International Conference on Advanced Topics in Artificial Intelligencerd Annual International Conference on Advanced Topics in Artificial Intelligence*, 2011, pp. 56–62.

[27] M. Mashayekhi, M. H. Sedehi, and R. Gras, “Can We Predict Speciation and Species Extinction Using an Individual-Based Ecosystem Simulation?,” in *International Conference on Artificial Intelligence (ICAI’13)*, 2013, pp. 301–307.

[28] M. Mashayekhi, B. MacPherson, and R. Gras, “A machine learning approach to investigate the reasons behind species extinction,” *Ecol. Inform.*, vol. 20, pp. 58–66, 2014.

[29] M. Mashayekhi, B. MacPherson, and R. Gras, “Species–area relationship and a tentative interpretation of the function coefficients in an ecosystem simulation,” *Ecol. Complex.*, vol. 19, pp. 84–95, Sep. 2014.

[30] M. Mashayekhi and R. Gras, “Rule extraction from random forest: RF+HC methods,” in *Advances in Artificial Intelligence*, 2015, pp. 223–237.

[31] C. Blake, E. Keogh, and C. J. Merz, "UCI repository of machine learning data bases (1998) www. ics. uci. edu/mlearn," *MLRepository. html*.

[32] C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, "Artificial Life II: Proceedings of the Workshop on Artificial Life, February 1990, Sante Fe, New Mexico. Redwood City, Calif." Addison-Wesley, 1992.

[33] J. Von Neumann, A. W. Burks, and others, "Theory of self-reproducing automata," 1966.

[34] N. Wiener, "Cybernetics; or control and communication in the animal and the machine.," 1948.

[35] R. A. Wilson and F. C. Keil, *The MIT encyclopedia of the cognitive sciences*. MIT press, 2001.

[36] C. G. Langton, *Artificial life: An overview*. Mit Press, 1997.

[37] J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.

[38] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[39] J. H. Holland, "Emergence: from chaos to order." Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1998.

[40] C. G. Langton and others, *Artificial life*. Addison-Wesley Publishing Company Redwood City, CA, 1989.

[41] M. A. Bedau, E. Snyder, and N. H. Packard, "A Classification of Longterm Evolutionary Dynamics," in *Proc. of Art. Life VI*, 1998, pp. 228–237.

[42] D. L. DeAngelis and W. M. Mooij, "Individual-based modeling of ecological and evolutionary processes," *Annu. Rev. Ecol. Evol. Syst.*, pp. 147–168, 2005.

[43] M. Niazi and A. Hussain, "Agent-based computing from multi-agent systems to agent-based models: a visual survey," *Scientometrics*, vol. 89, no. 2, pp. 479–499, 2011.

[44] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. Suppl 3, pp. 7280–7287, 2002.

[45] D. Hiebeler, "The swarm simulation system and individual-based modeling," 1994.

[46] V. Grimm, "Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future?," *Ecol. Modell.*, vol. 115, no. 2–3, pp. 129–148, Feb. 1999.

[47] H. H. Shugart and others, *A theory of forest dynamics. The ecological implications of forest succession models.* Springer-Verlag, 1984.

[48] K. A. Rose, E. S. Rutherford, D. S. McDermot, J. L. Forney, and E. L. Mills, "Individual-based model of yellow perch and walleye populations in Oneida Lake," *Ecol. Monogr.*, vol. 69, no. 2, pp. 127–154, 1999.

[49] B. Letcher, J. Priddy, and J. Walters, "An individual-based, spatially-explicit simulation model of the population dynamics of the endangered red-cockaded woodpecker, Picoides borealis," *Biol. Conserv.*, vol. 86, pp. 1–14, 1998.

[50] X. Thibert-Plante and a. P. Hendry, "Five questions on ecological speciation addressed with individual-based simulations," *J. Evol. Biol.*, vol. 22, no. 1, pp. 109–123, Jan. 2009.

[51] C. López-Alfaro, C. F. Estades, D. K. Aldridge, and R. Gill, "Individual-based modeling as a decision tool for the conservation of the endangered huemul deer (Hippocamelus bisulcus) in southern Chile," *Ecol. Modell.*, vol. 244, pp. 104–116, 2012.

[52] J. Zhang, M. T. Khasawneh, and S. R. Bowling, "Gender change in certain species - an agent-based modeling study," in *2010 IEEE Systems and Information Engineering Design Symposium*, 2010, pp. 225–228.

[53] C. A. Abbott, M. W. Berry, E. J. Comiskey, L. J. Gross, and H. K. Luh, "Computational Models of White-Tailed Deer in the Florida Everglades. UTK-CS." 1995.

[54] D. H. Deutschman, S. A. Levin, C. Devine, and L. A. Buttel, "Scaling from trees to forests: analysis of a complex simulation model," *Sci. Pap. Ed.*, vol. 277, no. 5332, pp. 1688–1696, 1997.

[55] S. Rasmussen, C. Knudsen, R. Feldberg, and M. Hindsholm, "The coreworld: Emergence and evolution of cooperative structures in a computational chemistry," *Phys. D Nonlinear Phenom.*, vol. 42, no. 1, pp. 111–134, 1990.

[56] T. Ray, "An Approach to the Synthesis of Life," *Artif. Life II*, pp. 371–408, 1991.

[57] T. S. Ray, "Evolution, ecology and optimization of digital organisms," *St. Fe*, 1992.

[58] T. Taylor and J. Hallam, "Replaying the tape: An investigation into the role of contingency in evolution," *Proc. Artif. Life VI, Los Angeles*, pp. 256–265, 1998.

[59] A. N. Pargellis, "Digital life behavior in the amoeba world," *Artif. Life*, vol. 7, no. 1, pp. 63–75, 2001.

[60] C. Adami and C. T. Brown, "Evolutionary learning in the 2D artificial life system Avida," in *Artificial life IV*, 1994, vol. 1194, pp. 377–381.

[61] C. Ofria and C. O. Wilke, "Avida: a Software Platform for Research in Computational Evolutionary Biology," *Artif. Life*, vol. 10, pp. 191–229, 2004.

[62] M. A. Fortuna, L. Zaman, A. P. Wagner, and C. Ofria, “Evolving digital ecological networks,” *PLoS Comput. Biol.*, vol. 9, no. 3, p. e1002928, 2013.

[63] C. Adami, C. Ofria, and T. C. Collier, “Evolution of biological complexity,” *Proc. Natl. Acad. Sci.*, vol. 97, no. 9, pp. 4463–4468, 2000.

[64] C. Adami, “Sequence complexity in Darwinian evolution,” *Complexity*, vol. 8, no. 2, pp. 49–56, 2002.

[65] R. Lenski, C. Ofria, R. Pennock, and C. Adami, “The Evolutionary Origin of Complex Features,” *Nature*, vol. 423, pp. 139–144, 2003.

[66] C. H. Chandler, C. Ofria, and I. Dworkin, “Runaway sexual selection leads to good genes,” *Evolution (N. Y).*, vol. 67, no. 1, pp. 110–119, 2013.

[67] J. H. Holland, “The echo model,” *Propos. a Res. Progr. Adapt. Comput. St. Fe Inst.*, 1992.

[68] O. J. Schmitz and G. Booth, “Modelling food web complexity: the consequences of individual-based, spatially explicit behavioural ecology on trophic interactions,” *Evol. Ecol.*, vol. 11, no. 4, pp. 379–398, 1997.

[69] L. Yaeger, “Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or PolyWorld: Life,” *Artif. Life III*, pp. 263–298, 1992.

[70] D. O. Hebb, *The organization of behavior: Aneuropsychological theory*. Wiley, 1949.

[71] L. S. Yaeger, “How evolution guides complexity,” *HFSP J.*, vol. 3, no. 5, pp. 328–339, 2009.

[72] L. Yaeger, V. Griffith, and O. Sporns, “Passive and driven trends in the evolution of complexity,” *arXiv Prepr. arXiv1112.4906*, 2011.

[73] J. Murdock and L. S. Yaeger, “Genetic clustering for the identification of species,” in *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, 2011, pp. 29–30.

[74] A. Channon and R. Damper, “Evolving novel behaviors via natural selection,” *Proc. Artif. Life VI, Los Angeles*, pp. 384–388, 1998.

[75] A. Channon and R. Damper, “Towards the Evolutionary Emergence of Increasingly Complex Advantageous Behaviours,” *Internaltion J. Syst. Sci.*, vol. 7, no. 31, pp. 843–860, 2000.

[76] A. Channon, “Passing the ALife Test: activity statistics classify evolution in Geb as unbounded,” in *6th European Conference on Advances in Artificial Life*, 2001, pp. 417–426.

[77] M. Komosi'nski and S. Ulatowski, "Framsticks: Towards a simulation of a nature-like world, creatures and evolution," in *Advances in Artificial Life*, Springer, 1999, pp. 261–265.

[78] J. M. Epstein and R. Axtell, *Growing artificial societies: social science from the bottom up*. Brookings Institution Press, 1996.

[79] M. Smith, "Using massively-parallel supercomputers to model stochastic spatial predator-prey systems," *Ecol. Modell.*, vol. 58, no. 1, pp. 347–367, 1991.

[80] V. Volterra, "Variations and Fluctuations of the Number of Individulas in Animal Species living together," *Anim. Ecol.*, vol. 3, pp. 409–448, 1931.

[81] G. Bell, "The evolution of trophic structure," *Heredity (Edinb).*, vol. 99, no. 5, pp. 494–505, 2007.

[82] W. Yamaguchi, M. Kondoh, and M. Kawata, "Effects of evolutionary changes in prey use on the relationship between food web complexity and stability," *Popul. Ecol.*, vol. 53, no. 1, pp. 59–72, 2011.

[83] C. J. Scogings, K. A. Hawick, and H. A. James, "Tools and techniques for optimisation of microscopic artificial life simulation models," in *Proceedings of the Sixth IASTED International Conference on Modelling, Simulation, and Optimization, Gabarone, Botswana*, 2006, pp. 90–95.

[84] C. J. Scogings and K. A. Hawick, "Modelling Predator Camouflage Behaviour and Tradeoffs in an Agent-Based Animat Model," in *Proc. IASTED International Conference on Modelling and Simulation (MS2013)*, 2013, no. CSTN-184, pp. 32–802.

[85] C. J. Scogings and K. A. Hawick, "Introducing a Gestation Period of Time-Delayed Benefit into an Animat-based Artificial Life Model," in *Proc. 12th IASTED Int. Conf. on Artificial Intelligence and Applications(AIA13)*, 2013, pp. 43–50.

[86] V. Grimm, U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. K. Heinz, G. Huse, A. Huth, J. U. Jepsen, C. Jørgensen, W. M. Mooij, B. Müller, G. Pe□er, C. Piou, S. F. Railsback, A. M. Robbins, M. M. Robbins, E. Rossmanith, N. Rüger, E. Strand, S. Souissi, R. A. Stillman, R. Vabø, U. Visser, and D. L. DeAngelis, "A standard protocol for describing individual-based and agent-based models," *Ecol. Modell.*, vol. 198, no. 1–2, pp. 115–126, Sep. 2006.

[87] D. Devaurs and R. Gras, "Species Abundance Patterns in an Ecosystem Simulation Studied through Fisher ' s Logseries," *Simul. Model. Pract. Theory*, vol. 18, no. 1, pp. 100–123, 2009.

[88] Y. M. Farahani and A. Golestani, "Complexity and Chaos Analysis of a Predator-Prey Ecosystem Simulation," *Cogn. 2010,* 2010.

[89] a. Golestani, R. Gras, and M. Cristescu, “Speciation with gene flow in a heterogeneous virtual world: can physical obstacles accelerate speciation?,” *Proc. R. Soc. B Biol. Sci.*, no. April, Apr. 2012.

[90] M. Hosseini and R. Gras, “Prediction of Imminent Species’ Extinction in EcoSim,” *Int. Conf. Agents Artif. Intell.*, pp. 318–323, 2012.

[91] G. Robin, A. Golestani, C. Melania, and A. P. Hendry, “Speciation without pre-defined fitness functions,” *PLoS One*, 2014.

[92] A. Aspinall and R. Gras, “K-means clustering as a speciation mechanism within an individual-based evolving predator-prey ecosystem simulation,” *Act. Media Technol.*, pp. 318–329, 2010.

[93] B. Kosko, “Fuzzy cognitive maps,” *Int. J. Man. Mach. Stud.*, vol. 24, no. 1, pp. 65–75, 1986.

[94] J. Tisseau, “Réalité virtuelle: autonomie in virtuo,” *Habilit. Dir. des Rech. Univ. Rennes*, vol. 1, 2001.

[95] M. Khater, D. Murariu, and R. Gras, “Contemporary evolution and genetic change of prey as a response to predator removal,” *Ecol. Inform.*, vol. 22, pp. 13–22, 2014.

[96] A. Golestani and R. Gras, “Multifractal phenomena in EcoSim, a large scale individual-based ecosystem simulation,” in *Int. Conf. Artificial Intelligence, Las Vegas, 2011*, 2011, pp. 991–999.

[97] L. Seuront, F. Schmitt, Y. Lagadeuc, D. Schertzer, S. Lovejoy, and S. Frontier, “Multifractal analysis of phytoplankton biomass and temperature in the ocean,” *Geophys. Res. Lett.*, vol. 23, no. 24, pp. 3591–3594, 1996.

[98] A. Golestani and R. Gras, “Identifying Origin of Self-Similarity in EcoSim, an Individual-Based Ecosystem Simulation, using Wavelet-based Multifractal Analysis,” in *Proceedings of the World Congress on Engineering and Computer Science*, 2012, vol. 2.

[99] V. N. Biktashev, J. Brindley, A. V Holden, and M. A. Tsyganov, “Pursuit-evasion predator-prey waves in two spatial dimensions,” *Chaos An Interdiscip. J. Nonlinear Sci.*, vol. 14, no. 4, pp. 988–994, 2004.

[100] N. F. Otani, A. Mo, S. Mannava, F. H. Fenton, E. M. Cherry, S. Luther, and R. F. Gilmour Jr, “Characterization of multiple spiral wave dynamics as a stochastic predator-prey system,” *Phys. Rev. E*, vol. 78, no. 2, p. 21913, 2008.

[101] M. Khater, E. Salehi, and R. Gras, “Correlation between Genetic Diversity and Fitness in a Predator-Prey Ecosystem Simulation,” *AI 2011 Adv. Artif. Intell.*, pp. 422–431, 2011.

[102] A. Golestani and R. Gras, “Regularity analysis of an individual-based ecosystem simulation,” *Chaos*, vol. 20, no. 4, p. 3120, 2010.

[103] J. Mallet, “A species definition for the modern synthesis,” *Trends Ecol. Evol.*, vol. 10, no. 7, pp. 294–299, 1995.

[104] D. Devaurs and R. Gras, “Species abundance patterns in an ecosystem simulation studied through Fisher’s logseries,” *Simul. Model. Pract. Theory*, vol. 18, no. 1, pp. 100–123, 2010.

[105] R. A. Fisher, A. S. Corbet, and C. B. Williams, “The relation between the number of species and the number of individuals in a random sample of an animal population,” *J. Anim. Ecol.*, pp. 42–58, 1943.

[106] B. Baesens, R. Setiono, C. Mues, and J. Vanthienen, “Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation,” *Manage. Sci.*, vol. 49, no. 3, pp. 312–329, Mar. 2003.

[107] D. Martens, B. Baesens, T. Van Gestel, and J. Vanthienen, “Comprehensible Credit Scoring Models Using Rule Extraction From Support Vector Machines Credit Risk Modelling , Group Risk Management , Dexia Group,” pp. 1–21.

[108] J. Chorowski and J. M. Zurada, “Extracting rules from neural networks as decision diagrams.,” *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2435–46, Dec. 2011.

[109] G. Fung, S. Sandilya, and R. B. Rao, “Rule extraction from linear support vector machines,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 32–40.

[110] R. Jiang, H. Yang, F. Sun, and T. Chen, “Searching for interpretable rules for disease mutations: a simulated annealing bump hunting strategy.,” *BMC Bioinformatics*, vol. 7, p. 417, Jan. 2006.

[111] R. Andrews, J. Diederich, and A. B. Tickle, “Survey and critique of techniques for extracting rules from trained artificial neural networks,” *Knowledge-based Syst.*, vol. 8, no. 6, pp. 373–389, 1995.

[112] J. Huysmans, B. Baesens, and J. Vanthienen, “Using rule extraction to improve the comprehensibility of predictive models,” *DTEW-KBI_0612*, pp. 1–55, 2006.

[113] J. Huysmans, R. Setiono, B. Baesens, and J. Vanthienen, “Minerva: Sequential covering for rule extraction,” *Syst. Man, Cybern. Part B Cybern. IEEE Trans.*, vol. 38, no. 2, pp. 299–309, 2008.

[114] Z. Zhou, Y. Jiang, and S. Chen, “Extracting symbolic rules from trained neural network ensembles,” *Ai Commun.*, 2003.

[115] N. Barakat and J. Diederich, “Eclectic rule-extraction from support vector machines,” *Int. J. Comput. Intell.*, vol. 2, no. 1, pp. 59–62, 2005.

[116] G. P. J. Schmitz, C. Aldrich, and F. S. Gouws, “ANN-DT: an algorithm for extraction of decision trees from artificial neural networks,” *Neural Networks, IEEE Trans.*, vol. 10, no. 6, pp. 1392–1401, 1999.

[117] R. Setiono, W. K. Leow, and J. M. Zurada, “Extraction of rules from artificial neural networks for nonlinear regression,” *Neural Networks, IEEE Trans.*, vol. 13, no. 3, pp. 564–577, 2002.

[118] U. Johansson, R. König, and L. Niklasson, “Rule extraction from trained neural networks using genetic programming,” in *13th International Conference on Artificial Neural Networks*, 2003, pp. 13–16.

[119] P. Clark, “The CN2 Induction Algorithm 1 Introduction,” vol. 3, no. 4, pp. 261–283, 1989.

[120] W. Cohen, “Fast Effective Rule Induction,” in *12th International Conference on Machine Learning*, 1995, pp. 115–123.

[121] J. H. Friedman and N. I. Fisher, “Bump hunting in high-dimensional data,” *Stat. Comput.*, vol. 9, no. 2, pp. 123–143, 1999.

[122] U. Johansson, R. König, and L. Niklasson, “Genetic rule extraction optimizing brier score,” *Proc. 12th Annu. Conf. Genet. Evol. Comput. - GECCO ’10*, p. 1007, 2010.

[123] G. W. Brier, “Verification of forecasts expressed in terms of probability,” *Mon. Weather Rev.*, vol. 78, no. 1, pp. 1–3, 1950.

[124] E. R. Hruschka and N. F. F. Ebecken, “Extracting rules from multilayer perceptrons in classification problems: a clustering-based approach,” *Neurocomputing*, vol. 70, no. 1, pp. 384–397, 2006.

[125] T. Q. Huynh and J. A. Reggia, “Guiding hidden layer representations for improved rule extraction from neural networks,” *Neural Networks, IEEE Trans.*, vol. 22, no. 2, pp. 264–275, 2011.

[126] P. Zhu and Q. Hu, “Rule extraction from support vector machines based on consistent region covering reduction,” *Knowledge-Based Syst.*, vol. 42, pp. 1–8, 2013.

[127] A. Li and G. Chen, “A new approach for rule extraction of expert system based on SVM,” *Measurement*, vol. 47, pp. 715–723, 2014.

[128] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[129] E. Frank and I. H. Witten, “Generating accurate rule sets without global optimization,” 1998.

[130] N. Barakat and A. P. Bradley, “Rule extraction from support vector machines: a review,” *Neurocomputing*, vol. 74, no. 1, pp. 178–190, 2010.

[131] J. J. Näppi, D. Regge, and H. Yoshida, “Comparative Performance of Random Forest and Support Vector Machine Classifiers for Detection of Colorectal Lesions in CT Colonography,” in *Proceedings of the Third International Conference on Abdominal Imaging: Computational and Clinical Applications*, 2012, pp. 27–34.

[132] R. Caruana and A. Niculescu-Mizil, “An Empirical Comparison of Supervised Learning Algorithms,” in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 161–168.

[133] B. Panda, J. S. Herbach, S. Basu, and R. J. Bayardo, “Planet: massively parallel learning of tree ensembles with mapreduce,” *Proc. VLDB Endow.*, vol. 2, no. 2, pp. 1426–1437, 2009.

[134] C. Chen, A. Liaw, and L. Breiman, “Using random forest to learn imbalanced data,” *Univ. California, Berkeley*, 2004.

[135] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[136] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[137] Y. Freund, R. Schapire, and N. Abe, “A short introduction to boosting,” *Journal-Japanese Soc. Artif. Intell.*, vol. 14, no. 771–780, p. 1612, 1999.

[138] R. E. Schapire, “The strength of weak learnability,” *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.

[139] J. Shotton, T. Sharp, P. Kohli, S. Nowozin, J. Winn, and A. Criminisi, “Decision jungles: Compact and rich models for classification,” in *Advances in Neural Information Processing Systems*, 2013, pp. 234–242.

[140] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, “How many trees in a random forest?,” in *Machine Learning and Data Mining in Pattern Recognition*, Springer, 2012, pp. 154–168.

[141] U. Johansson, C. Sonstrod, and T. Lofstrom, “One tree to explain them all,” in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, 2011, pp. 1444–1451.

[142] G. Martínez-Muñoz and A. Suárez, “Pruning in ordered bagging ensembles,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 609–616.

[143] G. Martinez-Muoz, D. Hernández-Lobato, and A. Suárez, “An analysis of ensemble pruning techniques based on ordered aggregation,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 31, no. 2, pp. 245–259, 2009.

[144] G. Brown, J. Wyatt, R. Harris, and X. Yao, “Diversity creation methods: A survey and categorisation,” *J. Inf. Fusion*, vol. 6, pp. 5–20, 2005.

[145] P. Latinne, O. Debeir, and C. Decaestecker, “Limiting the number of trees in random forests,” *Mult. Classif. Syst.*, pp. 1–10, 2001.

[146] Q. McNemar, “Note on the sampling error of the difference between correlated proportions or percentages,” *Psychometrika*, vol. 12, no. 2, pp. 153–157, Jun. 1947.

[147] H. Zhang and M. Wang, “Search for the smallest random forest,” *Stat. Interface*, vol. 2, no. 3, p. 381, 2009.

[148] S. Bernard, L. Heutte, and S. Adam, “On the selection of decision trees in random forests,” in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, 2009, pp. 302–307.

[149] M. Gashler, C. Giraud-Carrier, and T. Martinez, “Decision tree ensemble: Small heterogeneous is better than large homogeneous,” in *Machine Learning and Applications, 2008. ICMLA’08. Seventh International Conference on*, 2008, pp. 900–905.

[150] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[151] F. Yang, W. Lu, L. Luo, and T. Li, “Margin optimization based pruning for random forest,” *Neurocomputing*, vol. 94, pp. 54–63, 2012.

[152] J. H. Friedman and B. E. Popescu, “Predictive learning via rule ensembles,” *Ann. Appl. Stat.*, pp. 916–954, 2008.

[153] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. R. Stat. Soc. Ser. B*, pp. 267–288, 1996.

[154] N. Meinshausen, “Node harvest,” *Ann. Appl. Stat.*, pp. 2049–2072, 2010.

[155] S. Liu, R. Y. Patel, P. R. Daga, H. Liu, G. Fu, R. Doerksen, Y. Chen, and D. Wilkins, “Multi-class Joint Rule Extraction and Feature Selection for Biological Data,” *2011 IEEE Int. Conf. Bioinforma. Biomed.*, pp. 476–481, Nov. 2011.

[156] S. Liu, R. Y. Patel, P. R. Daga, H. Liu, G. Fu, R. J. Doerksen, Y. Chen, and D. E. Wilkins, “Combined rule extraction and feature elimination in supervised classification,” *NanoBioscience, IEEE Trans.*, vol. 11, no. 3, pp. 228–236, 2012.

[157] A. Van Assche and H. Blockeel, “Seeing the forest through the trees: Learning a comprehensible model from an ensemble,” in *Machine Learning: ECML 2007*, Springer, 2007, pp. 418–429.

[158] B. D. Mishler and M. J. Donoghue, “Species concepts: a case for pluralism,” *Syst. Zool.*, vol. 31, no. 4, pp. 491–503, 1982.

[159] E. Mayr, “Ecological Factors in Speciation,” *Evolution (N. Y).*, vol. 1, no. 4, pp. 263–288, 1947.

[160] E. Mayr, *Systematics and the origin of species, from the viewpoint of a zoologist*. Harvard University Press, 1942.

[161] S. Gavrilets, “Perspective: models of speciation: what have we learned in 40 years?,” *Evolution*, vol. 57, no. 10, pp. 2197–215, Oct. 2003.

[162] S. Ramachandran, O. Deshpande, C. C. Roseman, N. a Rosenberg, M. W. Feldman, and L. L. Cavalli-Sforza, “Support from the relationship of genetic and geographic distance in human populations for a serial founder effect originating in Africa.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 102, no. 44, pp. 15942–7, Nov. 2005.

[163] J. B. Losos and R. E. Glor, “Phylogenetic comparative methods and the geography of speciation,” *Trends Ecol. Evol.*, vol. 18, no. 5, pp. 220–227, May 2003.

[164] S. R. Jammalamadaka and A. Sengupta, *Topics in circular statistics*, vol. 5. World Scientific Pub Co Inc, 2001.

[165] L. Parrott, R. Proulx, and X. Thibert-Plante, “Three-dimensional metrics for the analysis of spatiotemporal data in ecology,” *Ecol. Inform.*, vol. 3, no. 6, pp. 343–353, 2008.

[166] H. He and E. A. Garcia, “Learning from Imbalanced Data,” *Knowl. Creat. Diffus. Util.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[167] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *arXiv Prepr. arXiv1106.1813*, 2011.

[168] N. V. Chawla, “Data Mining For Imbalanced Datasets: An Overview, Data Mining and Knowledge Discovery Handbook,” Springer, 2010, pp. 875–886.

[169] L. Parrott, R. Proulx, and X. Thibert-Plante, “Three-dimensional metrics for the analysis of spatiotemporal data in ecology,” *Ecol. Inform.*, vol. 3, no. 6, pp. 343–353, Dec. 2008.

[170] H. Li and J. F. Reynolds, “A new contagion index to quantify spatial patterns of landscapes,” *Landsc. Ecol.*, vol. 8, no. 3, pp. 155–162, Sep. 1993.

[171] W. B. Sherwin, “Entropy and Information Approaches to Genetic Diversity and its Expression: Genomic Geography,” *Entropy*, vol. 12, no. 7, pp. 1765–1798, Jul. 2010.

[172] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: an update,” *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[173] R. Hilborn, A. M. Schueller, and D. B. Hayes, “Minimum viable population size for lake sturgeon (Acipenser fulvescens) using an individual-based model of demographics and genetics,” *Can. J. Fish. Aquat. Sci.*, vol. 68, no. 1, pp. 62–73, Jan. 2011.

[174] J. M. Drake and B. D. Griffen, “Early warning signals of extinction in deteriorating environments.,” *Nature*, vol. 467, no. 7314, pp. 456–9, Sep. 2010.

[175] D. F. Sax and S. D. Gaines, “Species invasions and extinction : The future of native biodiversity on islands,” 2008.

[176] J. M. Drake and D. M. Lodge, “Effects of environmental variation on extinction and establishment,” *Ecol. Lett.*, vol. 7, no. 1, pp. 26–30, Jan. 2004.

[177] J. Joshi, P. Stoll, H.-P. Rusterholz, B. Schmid, C. Dolt, and B. Baur, “Small-scale experimental habitat fragmentation reduces colonization rates in species-rich grasslands.,” *Oecologia*, vol. 148, no. 1, pp. 144–52, May 2006.

[178] B. D. Griffen, J. M. Drake, and P. R. S. B, “Effects of habitat quality and size on extinction in experimental populations,” *Proc. R. Soc. B*, vol. 275, pp. 2251–2256, 2008.

[179] B. Dennis, “Allee effects: population growth, critical density, and the chance of extinction,” *Nat. Resour. Model.*, vol. 3, no. 4, pp. 481–538, 1989.

[180] D. Reed, E. Lowe, and D. Briscoe, “Inbreeding and extinction: effects of rate of inbreeding,” *Conserv. Genet.*, vol. 4, no. 3, pp. 405–410, 2003.

[181] J. a Markert, D. M. Champlin, R. Gutjahr-Gobell, J. S. Grear, A. Kuhn, T. J. McGreevy, A. Roth, M. J. Bagley, and D. E. Nacci, “Population genetic diversity and fitness in multiple environments.,” *BMC Evol. Biol.*, vol. 10, p. 205, Jan. 2010.

[182] J. M. Drake, J. Shapiro, and B. D. Griffen, “Experimental demonstration of a two-phase population extinction hazard.,” *J. R. Soc. Interface*, vol. 8, no. 63, pp. 1472–9, Oct. 2011.

[183] C. D. Collins, R. D. Holt, and B. L. Foster, “Patch size effects on plant species decline in an experimentally fragmented landscape.,” *Ecology*, vol. 90, no. 9, pp. 2577–88, Sep. 2009.

[184] N. a. Doran, a. J. Arnold, W. C. Parker, and F. W. Huffer, “Is Extinction Age Dependent?,” *Palaios*, vol. 21, no. 6, pp. 571–579, Dec. 2006.

[185] K. L. Evans, J. J. D. Greenwood, and K. J. Gaston, “The roles of extinction and colonization in generating species – energy relationships,” pp. 498–507, 2005.

[186] B. D. Griffen and J. M. Drake, “A review of extinction in experimental populations,” *J. Anim. Ecol.*, vol. 77, no. 6, pp. 1274–1287, 2008.

[187] U. K. Rai, “Minimum Sizes for Viable Population and Conservation Biology,” pp. 3–9, 2003.

[188] S. D. Gregory and F. Courchamp, “Safety in numbers: extinction arising from predator-driven Allee effects,” *J. Anim. Ecol.*, vol. 79, no. 3, pp. 511–514, 2010.

[189] H. Pälike, “Impact and extinction,” *Science (80-.).*, vol. 339, no. 6120, pp. 655–656, 2013.

[190] O. Ovaskainen and B. Meerson, “Stochastic models of population extinction.,” *Trends Ecol. Evol.*, vol. 25, no. 11, pp. 643–52, Nov. 2010.

[191] P. Perner, “Improving the Accuracy of Decision Tree Induction by Feature Pre-Selection,” vol. 15, no. 8, pp. 747–760, 2001.

[192] a Purvis, J. L. Gittleman, G. Cowlishaw, and G. M. Mace, “Predicting extinction risk in declining species.,” *Proc. Biol. Sci.*, vol. 267, no. 1456, pp. 1947–52, Oct. 2000.

[193] T. M. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[194] S. B. . Lande R. Engen S, *Stochastic Population Dynamics in Ecology and Conservation*. New York: Oxford University Press, 2003.

[195] R. Lande, “Anthropogenic, ecological and genetic factors in extinction and conservation,” *Res. Popul. Ecol. (Kyoto).*, vol. 40, no. 3, pp. 259–269, 1998.

[196] B. W. Brook, D. W. Tonkyn, J. J. O. Grady, and R. Frankham, “Contribution of Inbreeding to Extinction Risk in Threatened Species,” vol. 6, no. 1, 2002.

[197] D. Newman and D. Pilson, “Increased probability of extinction due to decreased genetic effective population size: Experimental populations of Clarkia pulchella,” *Evolution (N. Y).*, vol. 51, pp. 354–362, 1997.

[198] Saccheri I, M. Kuussaari, M. Kankare, P. Vikman, W. Fortelius, and I. Hanski, “Inbreeding and extinction in a butterfly metapopulation,” *Nature*, pp. 491–494, 1998.

[199] R. Frankham, “Genetics and extinction,” *Biol. Conserv.*, vol. 126, no. 2, pp. 131–140, Nov. 2005.

[200] G. Rompré, W. D. Robinson, A. Desrochers, and G. Angehr, “Predicting declines in avian species richness under nonrandom patterns of habitat loss in a neotropical landscape.,” *Ecol. Appl.*, vol. 19, no. 6, pp. 1614–1627, Sep. 2009.

[201] J. Rybicki and I. Hanski, “Species--area relationships and extinctions caused by habitat loss and fragmentation,” *Ecol. Lett.*, vol. 16, no. s1, pp. 27–38, 2013.

[202] P. G. Desmet and R. M. Cowling, “Using the Species-Area Relationship to Set Baseline Targets for Conservation,” *Ecol. Soc.*, vol. 9, no. 2, pp. 1–39, 2004.

[203] S. Fattorini, “To Fit or Not to Fit? A Poorly Fitting Procedure Produces Inconsistent Results When the Species–Area Relationship is used to Locate Hotspots,” *Biodivers. Conserv.*, vol. 16, no. 9, pp. 2531–2538, May 2006.

[204] D. P. Tittensor, F. Micheli, M. Nyström, and B. Worm, “Human impacts on the species-area relationship in reef fish assemblages.,” *Ecol. Lett.*, vol. 10, no. 9, pp. 760–72, Sep. 2007.

[205] M. Murakami and T. Hirao, “Lizard predation alters the effect of habitat area on the species richness of insect assemblages on Bahamian isles,” *Divers. Distrib.*, vol. 16, no. 6, pp. 952–958, Nov. 2010.

[206] J. B. Plotkin, M. D. Potts, D. W. Yu, S. Bunyavejchewin, R. Condit, R. Foster, S. Hubbell, J. LaFrankie, N. Manokaran, L. H. Seng, R. Sukumar, M. a Nowak, and P. S. Ashton, “Predicting species diversity in tropical forests.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 97, no. 20, pp. 10850–10854, Sep. 2000.

[207] M. Lomolino, “Ecology’s most general, yet protean 1 pattern: the species□area relationship,” *J. Biogeogr.*, vol. 27, no. 1, pp. 17–26, 2000.

[208] W. Ulrich and J. Buszko, “Basic and Applied Ecology Self-similarity and the species – area relation of Polish butterflies,” *Basic Appl. Ecol.*, vol. 270, pp. 263–270, 2003.

[209] R. J. Whittaker and J. M. Fernández-Palacios, *Island biogeography: ecology, evolution, and conservation*. Oxford University Press, 2007.

[210] J. Dengler and S. Boch, “Sampling-Design Effects on Properties of Species-Area Relationships – A Case Study from Estonian Dry Grassland Communities,” *Folia Geobot.*, vol. 43, no. 3, pp. 289–304, Nov. 2008.

[211] S. Fattorini, “On the general dynamic model of oceanic island biogeography,” *J. Biogeogr.*, vol. 36, no. 6, pp. 1100–1110, 2009.

[212] C. Dolnik and M. Breuer, “Scale Dependency in the Species-Area Relationship of Plant Communities,” *Folia Geobot.*, vol. 43, no. 3, pp. 305–318, Nov. 2008.

[213] A. I. Azovsky, “Species-area and species-sampling effort relationships: disentangling the effects,” *Ecography (Cop.).*, vol. 34, no. 1, pp. 18–30, Feb. 2011.

[214] C. H. Flather and R. Mountain, “Fitting Species-Accumulation Functions and Assessing Regional Land Use Impacts on Avian Diversity,” *Biogeography*, vol. 23, no. 2, pp. 155–168, 1996.

[215] F. He and P. Legendre, “On species-area relations,” *Am. Nat.*, vol. 148, no. 4, pp. 719–737, 1996.

[216] E. Tjørve, “Shapes and functions of species-area curves (II): a review of new models and parameterizations,” *J. Biogeogr.*, vol. 36, no. 8, pp. 1435–1445, Aug. 2009.

[217] E. F. Connor and E. D. McCoy, “Species–area relationships,” *Encycl. Biodivers.*, vol. 5, pp. 397–411, 2001.

[218] J. E. Keeley and C. J. Fotheringham, “Species-area relationships in Mediterranean-climate plant communities,” *J. Biogeogr.*, vol. 30, no. 11, pp. 1629–1657, Oct. 2003.

[219] H. García Martín and N. Goldenfeld, “On the origin and robustness of power-law species-area relationships in ecology.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 27, pp. 10310–5, Jul. 2006.

[220] S. Drakare, J. J. Lennon, and H. Hillebrand, “The imprint of the geographical, evolutionary and ecological context on species-area relationships.,” *Ecol. Lett.*, vol. 9, no. 2, pp. 215–27, Feb. 2006.

[221] A. Surendra, “Prediction of plant species diversity by log-linear and power function models along the east coast of India,” *Trop. Ecol.*, vol. 50, no. 1, pp. 103–109, 2009.

[222] H. Merwe and M. W. Rooyen, “Species–area relationships in the Hantam-Tanqua-Roggeveld, Succulent Karoo, South Africa,” *Biodivers. Conserv.*, vol. 20, no. 6, pp. 1183–1201, Mar. 2011.

[223] K. a. Triantis, F. Guilhaumon, and R. J. Whittaker, “The island species-area relationship: biology and statistics,” *J. Biogeogr.*, vol. 39, no. 2, pp. 215–231, Feb. 2012.

[224] F. He and P. Legendre, “Species diversity patterns derived from species-area models,” *Ecology*, vol. 83, no. 5, pp. 1185–1198, 2002.

[225] E. Tjørve and K. M. C. Tjørve, “The species-area relationship, self-similarity, and the true meaning of the z-value.,” *Ecology*, vol. 89, no. 12, pp. 3528–33, Dec. 2008.

[226] S. M. Scheiner, “Six types of species-area curves,” *Glob. Ecol. Biogeogr.*, vol. 12, no. 6, pp. 441–447, Oct. 2003.

[227] T. D. Olszewski, “A unified mathematical framework for the measurement of richness and evenness within and among multiple communities,” *Oikos*, vol. 104, no. 2, pp. 377–387, Feb. 2004.

[228] E. Tjørve, W. E. Kunin, C. Polce, and K. M. Calf Tjørve, “Species-area relationship: separating the effects of species abundance and spatial distribution,” *J. Ecol.*, vol. 96, no. 6, pp. 1141–1151, Nov. 2008.

[229] L. Jost, “The Relation between Evenness and Diversity,” *Diversity*, vol. 2, no. 2, pp. 207–232, Feb. 2010.

[230] H. M. Pereira, L. Borda-de-Água, and I. S. Martins, “Geometry and scale in species-area relationships.,” *Nature*, vol. 482, no. 7386, pp. E3–4; author reply E5–6, Feb. 2012.

[231] M. Cencini, S. Pigolotti, and M. A. Muñoz, “What ecological factors shape species-area curves in neutral models?,” *PLoS One*, vol. 7, no. 6, p. e38232, 2012.

[232] W. R. Turner and E. Tjørve, “Scale-dependence in species-area relationships,” *Ecography (Cop.).*, vol. 28, no. 6, pp. 721–730, Dec. 2005.

[233] F. He and S. P. Hubbell, “Species-area relationships always overestimate extinction rates from habitat loss.,” *Nature*, vol. 473, no. 7347, pp. 368–71, May 2011.

[234] E. Connor and E. McCoy, “The statistics and biology of the species-area relationship,” *Am. Nat.*, vol. 113, no. 6, pp. 791–833, 1979.

[235] S. Gould, “An allometric interpretation of species-area curves: the meaning of the coefficient,” *Am. Nat.*, vol. 114, no. 3, pp. 335–343, 1979.

[236] T. Martin, “Species-area slopes and coefficients: a caution on their interpretation,” *Am. Nat.*, vol. 118, no. 6, pp. 823–837, 1981.

[237] M. Franzén, O. Schweiger, and P.-E. Betzholtz, “Species-area relationships are controlled by species traits,” *PLoS One*, vol. 7, no. 5, p. e37359, 2012.

[238] D. Lawson and H. J. Jensen, “The species--area relationship and evolution,” *J. Theor. Biol.*, vol. 241, no. 3, pp. 590–600, 2006.

[239] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[240] C. Crisci, B. Ghattas, and G. Perera, “A review of supervised machine learning algorithms and their applications to ecological data,” *Ecol. Modell.*, vol. 240, pp. 113–122, Aug. 2012.

[241] A. Golestani and R. Gras, “Regularity analysis of an individual-based ecosystem simulation,” *Chaos An Interdiscip. J. Nonlinear Sci.*, vol. 20, no. 4, p. 43120, 2010.

[242] J. Harte, “Self-Similarity in the Distribution and Abundance of Species,” *Science (80-.).*, vol. 284, no. 5412, pp. 334–336, Apr. 1999.

[243] A. Baselga and C. D. L. Orme, “betapart: an R package for the study of beta diversity,” *Methods Ecol. Evol.*, vol. 3, no. 5, pp. 808–812, 2012.

[244] M. Lazarina, V. Sgardeli, A. S. Kallimanis, and S. P. Sgardelis, “An effort-based index of beta diversity,” *Methods Ecol. Evol.*, vol. 4, no. 3, pp. 217–225, 2013.

[245] D. Storch, P. A. Marquet, J. H. Brown, and others, *Scaling biodiversity*. Cambridge University Press Cambridge, UK, 2007.

[246] H. Gitay, S. H. Roxburgh, and J. B. Wilson, “Species-area relations in a New Zealand tussock grassland, with implications for nature reserve design and for community structure,” *J. Veg. Sci.*, vol. 2, no. 1, pp. 113–118, 1991.

[247] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[248] K. P. Burnham and D. R. Anderson, *Model selection and multimodel inference: a practical information-theoretic approach*, vol. 45, no. 2. Springer Verlag, 2002.

[249] B. B. Mandelbrot, “The fractal geometry of nature/Revised and enlarged edition,” *New York, WH Free. Co., 1983, 495 p.*, vol. 1, 1983.

[250] L. Parrott, “Quantifying the complexity of simulated spatiotemporal population dynamics,” *Ecol. Complex.*, vol. 2, no. 2, pp. 175–184, 2005.

[251] J. D. Fridley, R. K. Peet, T. R. Wentworth, and P. S. White, “Connecting fine-and broad-scale species-area relationships of southeastern US flora,” *Ecology*, vol. 86, no. 5, pp. 1172–1177, 2005.

[252] J. Dengler and J. Oldeland, “Effects of sampling protocol on the shapes of species richness curves,” *J. Biogeogr.*, vol. 37, no. 9, pp. 1698–1705, Sep. 2010.

[253] J. Harte, S. McCarthy, K. Taylor, A. Kinzig, and M. L. Fischer, “Estimating species-area relationships from plot to landscape scale using species spatial-turnover data,” *Oikos*, pp. 45–54, 1999.

[254] J. M. Calderón-Patrón, C. E. Moreno, R. Pineda-López, G. Sánchez-Rojas, and I. Zuria, “Vertebrate Dissimilarity Due to Turnover and Richness Differences in a Highly Beta-Diverse Region: The Role of Spatial Grain Size, Dispersal Ability and Distance,” *PLoS One*, vol. 8, no. 12, p. e82905, 2013.

[255] H. Qian, “Global comparisons of beta diversity among mammals, birds, reptiles, and amphibians across spatial scales and taxonomic ranks,” *J. Syst. Evol.*, vol. 47, no. 5, pp. 509–514, 2009.

[256] M. Vellend, “Do commonly used indices of β-diversity measure species turnover?,” *J. Veg. Sci.*, vol. 12, no. 4, pp. 545–552, 2001.

[257] H. Heatwole, *Biogeography of reptiles on some of the islands and cays of eastern Papua-New Guinea*. Smithsonian Institution, 1975.

[258] W. Ulrich and J. Buszko, “Habitat reduction and patterns of species loss,” *Basic Appl. Ecol.*, vol. 5, no. 3, pp. 231–240, 2004.

[259] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[260] V. O. Nams and M. Bourgeois, “Fractal analysis measures habitat use at different spatial scales: an example with American marten,” *Can. J. Zool.*, vol. 82, no. 11, pp. 1738–1747, 2004.

[261] C. Lead, O. E. Sala, D. van Vuuren, and A. S. Zaitsev, “Biodiversity across scenarios,” *Ecosyst. Hum. Well-Being Scenar. Find. Scenar. Work. Gr.*, vol. 2, p. 375, 2005.

[262] J. W. Morgan, N. K. Wong, and S. C. Cutler, “Life-form species–area relationships in a temperate eucalypt woodland community,” *Plant Ecol.*, vol. 212, no. 6, pp. 1047–1055, Jan. 2011.

[263] D. Storch, P. Keil, and W. Jetz, “Universal species-area and endemics-area relationships at continental scales.,” *Nature*, vol. 488, no. 7409, pp. 78–81, Aug. 2012.

[264] M. J. Crawley and J. E. Harral, “Scale dependence in plant biodiversity.,” *Science*, vol. 291, no. 5505, pp. 864–8, Feb. 2001.

[265] A.-L. Boulesteix, S. Janitza, J. Kruppa, and I. R. König, “Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 2, no. 6, pp. 493–507, 2012.

[266] W. G. Touw, J. R. Bayjanov, L. Overmars, L. Backus, J. Boekhorst, M. Wels, and S. A. F. T. van Hijum, “Data mining in the Life Sciences with Random Forest: a walk in the park or lost in the jungle?,” *Brief. Bioinform.*, pp. 315–326, 2012.

[267] L. Song, P. Langfelder, and S. Horvath, “Random generalized linear model: a highly accurate and interpretable ensemble predictor,” *BMC Bioinformatics*, vol. 14, no. 1, p. 5, 2013.

[268] B. K. Sarkar, S. S. Sana, and K. Chaudhuri, “A genetic algorithm-based rule extraction system,” *Appl. Soft Comput.*, vol. 12, no. 1, pp. 238–254, 2012.

[269] B. Selman and C. P. Gomes, “Hill-climbing Search,” *Encycl. Cogn. Sci.*, 2006.

[270] J. Huysmans, B. Baesens, and J. Vanthienen, “Using rule extraction to improve the comprehensibility of predictive models,” *DTEW-KBI_0612*, 2006.

[271] J. H. Ang, K. C. Tan, and A. A. Mamun, “An evolutionary memetic algorithm for rule extraction,” *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1302–1315, 2010.

[272] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bull.*, pp. 80–83, 1945.

[273] M. Friedman, “The use of ranks to avoid the assumption of normality implicit in the analysis of variance,” *J. Am. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.

[274] J. Demšar, “Statistical Comparisons of Classifiers over Multiple Data Sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.

[275] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and others, “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring,” *Science (80-.).*, vol. 286, no. 5439, pp. 531–537, 1999.

[276] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, “Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays,” *Proc. Natl. Acad. Sci.*, vol. 96, no. 12, pp. 6745–6750, 1999.

[277] C. L. Nutt, D. R. Mani, R. A. Betensky, P. Tamayo, J. G. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. E. McLaughlin, T. T. Batchelor, and others, “Gene expression-based

classification of malignant gliomas correlates better with survival than histological classification," *Cancer Res.*, vol. 63, no. 7, pp. 1602–1607, 2003.

[278] L. J. van't Veer, H. Dai, M. J. Van De Vijver, Y. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, and others, "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, no. 6871, pp. 530–536, 2002.

VITA AUCTORIS

Morteza Mashayekhi received his Bachelor's and Master's Degrees in Computer Engineering from University of Isfahan and Isfahan University of Technology, Iran, in 2001, and 2004, respectively. He studied in the School of Computer Science, University of Windsor, Canada, from 2010 to 2015 for a Degree of Doctoral Philosophy. His research interests include Machine learning, Artificial Life, Ecological modeling.