Machine Learning Methods Without Tears: A Primer for Ecologists

Author(s): Julian D. Olden, Joshua J. Lawler and N. LeRoy Poff

Source: *The Quarterly Review of Biology*, Vol. 83, No. 2 (June 2008), pp. 171-193

Published by: The University of Chicago Press

Stable URL: https://www.jstor.org/stable/10.1086/587826

MACHINE LEARNING METHODS WITHOUT TEARS: A PRIMER FOR ECOLOGISTS

JULIAN D. OLDEN
School of Aquatic and Fishery Sciences, University of Washington
Seattle, Washington 98195 USA
E-MAIL: olden@u.washington.edu

JOSHUA J. LAWLER
College of Forest Resources, University of Washington
Seattle, Washington 98195 USA
E-MAIL: jlawler@u.washington.edu

N. LEROY POFF
Department of Biology, Colorado State University
Fort Collins, Colorado 80523 USA
E-MAIL: poff@lamar.colostate.edu

KEYWORDS
ecological informatics, classification and regression trees, artificial neural networks, evolutionary algorithms, genetic algorithms, GARP, inductive modeling

ABSTRACT

Machine learning methods, a family of statistical techniques with origins in the field of artificial intelligence, are recognized as holding great promise for the advancement of understanding and prediction about ecological phenomena. These modeling techniques are flexible enough to handle complex problems with multiple interacting elements and typically outcompete traditional approaches (e.g., generalized linear models), making them ideal for modeling ecological systems. Despite their inherent advantages, a review of the literature reveals only a modest use of these approaches in ecology as compared to other disciplines. One potential explanation for this lack of interest is that machine learning techniques do not fall neatly into the class of statistical modeling approaches with which most ecologists are familiar. In this paper, we provide an introduction to three machine learning approaches that can be broadly used by ecologists: classification and regression trees, artificial neural networks, and evolutionary computation. For each approach, we provide a brief background to the methodology, give examples of its application in ecology, describe model development and implementation, discuss strengths and weaknesses, explore the availability of statistical software, and provide an illustrative

The Quarterly Review of Biology, June 2008, Vol. 83, No. 2

0033-5770/2008/8302-0002$15.00

example. Although the ecological application of machine learning approaches has increased, there remains considerable skepticism with respect to the role of these techniques in ecology. Our review encourages a greater understanding of machine learning approaches and promotes their future application and utilization, while also providing a basis from which ecologists can make informed decisions about whether to select or avoid these approaches in their future modeling endeavors.

INTRODUCTION

PREDICTIVE ABILITY is considered by many to be the ultimate goal in ecology (Peters 1991). Recent decades have witnessed an increasing role of prediction in applied ecology, in large part because of the mounting threats to biological diversity from global environmental change and the resulting need for ecological forecasting (Clark et al. 2001). Such efforts, however, are hindered by the many complexities of ecosystems, including historical legacies, time lags, nonlinearities, interactions, and feedback loops that vary in both time and space (Levin 1998). Accordingly, ecologists are challenged by the need to understand and predict complex ecological processes and patterns.

One promising set of quantitative tools that can help solve such environmental challenges (e.g., global climate change, emerging diseases, biodiversity loss) is currently being researched and developed under the rubric of ecological informatics (Green et al. 2005). Ecological informatics, or eco-informatics, is an interdisciplinary framework that promotes the use of advanced computational technology to reveal ecological processes and patterns across levels of ecosystem complexity (Recknagel 2003). Machine learning (ML) is a rapidly growing area of eco-informatics that is concerned with identifying structure in complex, often nonlinear data and generating accurate predictive models. Recent advances in data collection technology, such as remote-sensing and data network centers and archives, have produced large, high-resolution datasets spanning spatial and temporal extents that were, until recently, unattainable. As a result, ecologists have the exciting opportunity to take advantage of ML approaches to model the complex relationships inherent in these large datasets. Applications of ML methods in ecology are diverse, and range from testing biogeographical, ecological, and evolutionary hypotheses to modeling species distributions for conservation and management planning (e.g., Fielding 1999; Recknagel 2001, 2003; Cushing and Wilson 2005; Ferrier and Guisan 2006; Park and Chon 2007).

ML algorithms can be organized according to a diverse taxonomy that reflects the desired outcome of the modeling process. A number of ML techniques have been promoted in ecology as powerful alternatives to traditional modeling approaches. These include supervised learning approaches that attempt to model the relationship between a set of inputs and known outputs, such as artificial neural networks (Lek et al. 1996), cellular automata (Hogeweg 1988), classification and regression trees (De'ath and Fabricius 2000), fuzzy logic (Salski and Sperlbaum 1991), genetic algorithms and programming (Stockwell and Noble 1992), maximum entropy (Phillips et al. 2006), support vector machines (Drake et al. 2006), and wavelet analysis (Cho and Chon 2006). In addition, unsupervised learning approaches are used to reveal patterns in ecological data, including Hopfield neural networks (Hopfield 1982) and self-organizing maps (Kohonen 2001). The growing use of these methods in recent years is the direct result of their ability to model complex, nonlinear relationships in ecological data without having to satisfy the restrictive assumptions required by conventional, parametric approaches (Guisan and Zimmermann 2000; Peterson and Vieglais 2001; Olden and Jackson 2002a; Elith et al. 2006). As a result, ML approaches often exhibit greater power for explaining and predicting ecological patterns. The recent formation of The International Society for Ecological Informatics, as well as the birth of the scientific journal *Ecological Informatics,*

supports the observation that ML has evolved from a field of theoretical demonstrations to one of significant and applied value in ecology.

Perhaps not surprisingly, ML approaches are predominantly used by ecologists with strong computational skills, and they have seen only limited use within the broader scientific community. Why have ML approaches not been widely embraced by ecologists? One reason is that ecologists may lack the fundamental background needed to understand and implement these methods, and they may be unsure about how to select approaches that best suit their needs. At the same time, researchers in the field of ecological informatics continue to advance better and more complex ML algorithms, arguing that more powerful computers and increased availability of large ecological data sets will move them further into the mainstream. Unfortunately, as the technology in ML grows, so does the inaccessibility of these techniques to the majority of ecologists who still require a basic understanding of why, when, where, and how such approaches should be applied.

We argue that there are relatively few examples in the ecological literature that encourage the exploration and promote the application of ML methods. In this paper, we will address this concern by providing a comprehensive review of three ML methods that have recently gained popularity among ecologists: classification and regression trees, artificial neural networks, and evolutionary computation (genetic algorithms and programming); however, we recognize that other statistical approaches, including generalized additive models and multivariate adaptive regression splines, have also illustrated utility in ecology (e.g., Austin 2007; Elith and Leathwick 2007). For each approach, we will provide a brief background to the methodology, give examples of its application in ecology, describe model development and implementation, discuss strengths and weaknesses, and explore the availability of statistical software. In order to more clearly illustrate the basic principles of the ML methodologies, we will apply each method to a common ecological question, namely modeling species richness (dependent variable) as a function of environmental descriptors (independent variables). We stress that our review is not meant to replace previously published texts on ML (e.g., Fielding 1999; Lek and Guégan 2000), rather it is intended to provide a gentle introduction to ML methods that is more readily accessible to the broad community of ecologists. We accomplish this by favoring written explanation over mathematical formulas and by avoiding statistical jargon that often serves to limit the readership and comprehension of ML methodologies by ecologists. Put simply, our hope is that this paper will encourage a greater understanding and the future application of ML approaches in the ecological sciences.

An Illustrative Example of Machine Learning Methods

In order to illustrate ML methodologies, we will use an empirical example relating fish species richness to environmental characteristics of 8236 north-temperate lakes in Ontario, Canada. Identifying both patterns and drivers of species richness is a long-standing problem in ecology because environmental factors typically interact in nonlinear ways to influence the number of species at any particular site. This example is used solely to demonstrate a common statistical problem in which a researcher is interested in modeling a single dependent variable as a function of multiple independent variables and, thus, is not meant as a comparative analysis of approaches. We chose this relatively simple dataset and straightforward ecological problem for illustrative purposes. Although ML approaches are well-suited for addressing even seemingly simple problems, many of the advantages of these approaches can be brought to bear on much more complex problems as well.

We selected 8 whole-lake descriptors that are related to habitat requirements of temperate fish species of the Ontario region (Minns 1989). Regional climate was represented by the mean monthly air temperature (TEMP, measured in °C) and

mean monthly precipitation (PPT, cm) for each lake based on data collected from 1836 recording stations between 1960 and 1989 by the Atmospheric Environment Service of Environment Canada (see Vander Zanden et al. 2004). Whole-lake measures of habitat included lake surface area (AREA, km^2), total shoreline perimeter (SHP, km), maximum depth (MAXD, m), elevation (ELEV, m), secchi disc depth (SDD, m), and pH. The primary source of the fish distribution data was the Fish Species Distribution Data System of the Ontario Ministry of Natural Resources. To assess predictive performance of the models, we used 10-fold cross-validation. In this procedure, the original data is partitioned into 10 subsamples each containing $n/10$ observations; a single subsample is retained as the validation data for testing the model, and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times (hence, folds), with each of the subsamples used exactly once as the validation data. The 10 results are then combined to produce a single set of predictions for all n observations. For general information regarding model selection, model validation, and the assessment of predictive performance (i.e., topics that are not the focus of our study), we refer the reader to Fielding and Bell (1997).

Classification and Regression Trees (CARTs)

background and ecological applications

Classification and Regression Trees (CARTs), collectively called decision trees, date from the pioneering work of Morgan and Sonquist (1963) in the social sciences, and their use in statistical literature was rekindled by the seminal monograph of Breiman et al. (1984). Since this time, decision trees have been widely used in a number of applied sciences including medicine, computer science, and psychology (Ripley 1996). Recent years have seen CARTs emerge as powerful statistical tools for analyzing complex ecological datasets because they offer a useful alternative when modeling nonlinear data containing independent variables that are suspected of interacting in a hierarchical fashion (De'ath and Fabricius 2000).

There have been numerous ecological applications of CARTs across a wide range of topics. Decision trees have been used to develop habitat models for threatened birds (O'Connor et al. 1996), tortoise species (Anderson et al. 2000), and endangered crayfishes (Usio 2007). Iverson and Prasad (1998) forecasted potential shifts in tree species distributions resulting from climatic warming, Rollins et al. (2004) quantified the relationship between the frequency and severity of forest fires and landscape structure, and Mercado-Silva et al. (2006) predicted patterns of fish species invasions in the Laurentian Great Lakes. Other applications have involved modeling patterns of variability in PCB concentrations of salmonid species (Lamon and Stow 1999), predicting days postpartum from fatty acids measured in harbor seal milk (Smith et al. 1997), delineating geographic patterns of bottlenose dolphin ecotypes (Torres et al. 2003), and developing models that assessed the vulnerability of the landscape to tsunami damage (Iverson and Prasad 2007).

methodology

CART analysis is a form of binary recursive partitioning where classification and regression trees refer to the modeling of categorical and continuous response variables, respectively (Bell 1999). The general anatomy of a decision tree is presented in Figure 1. The term "binary" implies that each group of observations, represented by a node in a decision tree, is split into two child nodes, a process through which the original node becomes a parent node. The term "recursive" refers to the fact that the binary partitioning process can be applied repetitively. Thus, each parent node can give rise to two child nodes and, in turn, each of these child nodes may themselves be split, forming additional children. The term "partitioning" refers to the fact that the dataset is split into sections or partitioned. Although there are many different versions of binary recursive partitioning available, each with its own unique details,

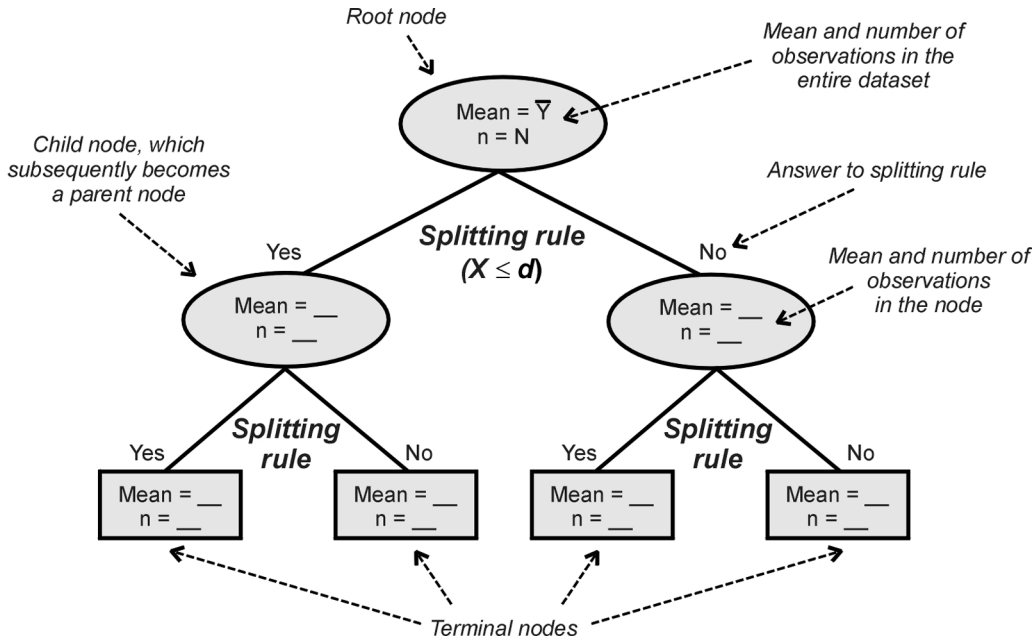

Figure 1. The General Anatomy of a Classification or Regression Tree

the overall methodology is consistent regardless of the exact implementation.

CART analysis consists of three basic steps. The first step involves *tree building*, during which a decision tree is built by repeatedly partitioning the data set into a nested series of mutually exclusive groups, each of them as homogeneous as possible with respect to the response variable. Tree building begins at the root node with the entire dataset, and the algorithm formulates split-defining conditions for each possible value of all the independent variables to create candidate—or surrogate—splits. Other splitting criteria are also available. Next, the algorithm selects the best candidate split that minimizes the average "impurity" of the two child nodes. Impurity is based on a goodness of fit measure, such as the information (entropy) index and the Gini index for classification trees and sums of squares about group means for regression trees (De'ath and Fabricius 2000). The algorithm continues recursively with each of the new children nodes until tree building is stopped.

The second step consists of *stopping the tree building process*. The process is stopped when: (1) there are only n observations in each of the child nodes (where n is set by the user), (2) all observations within each child node have the identical distribution of independent variables, making splitting impossible, or (3) an external limit on the number of splits in the tree or minimum purity threshold is achieved. A terminal node or "leaf" is a node that the algorithm cannot partition any further because one of the above criteria is met. In classification trees, each node—even the root node—is assigned a predicted probability for each class (often the class with the greatest probability is assigned to the node). For regression trees, the predicted value for each node is typically defined as the mean or median value of the response variable for the observations in the node.

The third step involves *tree pruning and optimal tree selection*. The most common method used is called "cost-complexity" pruning, which results in the creation of a sequence of progressively simpler trees through the pruning or cutting of increasingly important nodes. This method relies

on a complexity parameter, denoted α, which is gradually increased during the pruning process. Starting at the terminal nodes, the child nodes are pruned away if the resulting change in the model error is less than α multiplied by the change in tree complexity. Thus, α is a measure of how much additional accuracy a split must add to the entire tree to warrant the additional complexity. As α is increased, more and more nodes of increasing importance are pruned away, resulting in simpler and simpler trees (Bell 1999). The goal in selecting the optimal tree is to find the correct complexity parameter α so that the information in the learning dataset is fit but not overfit, the latter condition occurring when the model successfully describes the specifics of the original data but is unable to predict additional data. It is common practice to determine the optimal size of the decision tree (i.e., the number of terminal nodes) by selecting the tree size with the smallest model error based on repeated cross-validations of the data. Alternatively, the smallest tree whose model error falls within the one standard error rule is also used (see De'ath and Fabricius 2000). For further discussion on validation techniques and their use in identifying ideal tree size, see Breiman et al. (1984), Bell (1999), Hastie et al. (2001), and Sutton (2005).

Based on the branching topology of the decision tree, one can interpret the primary splits that represent the most important variables in the prediction process, as well as the best competitive surrogate splits that also show high classification power. To calculate the overall importance of the independent variables in a decision tree, the CART analysis quantifies the improvement measure attributable to each variable in its role as a surrogate to the primary split. The values of these improvements are summed over each node, totaled, and scaled relative to the best performing variable, i.e., expressed as a relative importance on a 0–100% scale (Breiman et al. 1984).

STRENGTHS AND WEAKNESSES

CART analysis has a number of advantages over traditional statistical methods that make it particularly attractive for modeling ecological data. These include the fact that CARTs are: (1) inherently nonparametric and, therefore, not affected by heteroscedasticity or distributional error structures that affect parametric procedures; (2) invariant to monotonic transformations of the data, thus eliminating the need for data transformations; (3) able to handle mixed numerical data including categorical, interval, and continuous variables; (4) able to deal with missing variables by using the surrogate splitting variables in the decision tree; (5) not affected by outliers (outliers are isolated into a node, and have only a minimal effect on splitting); (6) able to detect and reveal interactions in the data set; (7) able to effectively deal with higher dimensionality (i.e., it can identify a reduced set of important variables from a large number of submitted variables); (8) relatively simple to interpret graphically. In general, of the three ML approaches reviewed here, CART is the most flexible with respect to data requirements and the most transparent when it comes to understanding the modeling process (Table 1).

Despite their many advantages, CARTs have a number of weaknesses that are rarely discussed in the literature. First, analyses that identify "splitting" variables by employing the exhaustive search of all possibilities have the chance of increasing the complexity of the search, thereby causing computational strain, especially with large data sets (although advances in computer speed have partially eliminated this problem). This tends to select a decision tree that has more splits, thus promoting overfitting. Second, deducing rules from a decision tree can be very complicated because it is not based on a probabilistic model. Therefore, there is no probability level or confidence interval associated with predictions derived from using a classification or regression tree to classify a new set of data. Third, correlations among independent variables can complicate the identification of important interactions. Fourth, decision trees are typically unstable, i.e., sometimes small changes in the

TABLE 1
Comparison of machine learning approaches according to a number of model characteristics

Characteristic	GLM	CART	ANN	EA
Data Requirements				
Accommodate "mixed" data types	Low	High	Low	Moderate
Accommodate missing values of predictors	Low	High	Low	Low
Insensitive to monotonic transformations of predictors	Low	High	Moderate	Moderate
Robust to outliers in predictors	Low	Moderate	Moderate	Moderate
Insensitive to irrelevant predictors	Low	High	Moderate	Moderate
Modeling Process				
Automation (i.e., low degree of user involvement)	High	Moderate	Moderate	Low
Transparency of the modeling process	High	Moderate	Low	Low
Ability to model nonlinear relationships	Low	Moderate	High	High
Accommodate interactions among predictors	Low	Moderate	High	High
Model Output				
Explanatory insight and variable interpretability	High	Moderate	Moderate	Low
Predictive power	Low	Moderate	High	High
Software Availability and Ease-of-Use	High	Moderate	Low	Low

Classification and regression trees (CARTs), artificial neural networks (ANNs), and evolutionary algorithms (EAs) are compared to the family of generalized linear models (GLMs) that are traditionally used in ecology. Comparisons are generalized to include both classification and prediction problems. Values are based on Hastie et al. (2001), peer-reviewed literature, and the personal experiences of the authors.

learning sample values can lead to significant changes in the variables used in the splits. As a result, overall variable importance cannot be determined by only examining the final tree; it also requires the examination of all possible surrogate splits. Fifth, perhaps the greatest weakness of CARTs is that the final decision tree is not guaranteed to be the optimal tree. At each splitting decision in the tree growing process, the selected split is the one that results immediately in reduced impurity (for classification) or variation (for regression). One might expect that some other split, which would appear suboptimal at the time, could produce more effective future splits (Sutton 2005). A variety of approaches have been developed to address the latter two problems, including the application of bagging and boosting techniques and the creation of an ensemble tree based on random forests of multiple trees. We refer the reader to De'ath (2007) and Cutler et al. (2007) for an ecological treatment of these topics.

SOFTWARE

Many commercial packages are available to implement CART. This software varies from requiring a fair amount of user design and programming to Windows-based programs to powerful and user-friendly Graphical User Interfaces. Windows-based programs include CART (www.salford-systems.com), DTREG (www.dtreg.com), KnowledgeSEEKER (www.angoss.com), QUEST (www.stat.wisc.edu/~loh), PolyAnalyst (www.megaputer.com), Random Forests (www.stat.berkeley.edu/users/breiman), Shih Data Miner (www.shih.be), See5/C5.0 (www.rulequest.com), and XpertRule Miner (www.attar.com). Modules and libraries for statistical software packages include AnswerTree for SPSS (www.spss.com/answertree), Multivariate Exploratory Techniques (Classification Trees) for Statistica (www.statsoft.com), Enterprise Miner for SAS (www.sas.com), Tree library for S-Plus (http://lib.stat.cmu.edu/S), and Rpart for the R-package (http://cran.r-project.org).

CASE STUDY

We constructed a regression tree to gain explanatory and predictive insight into the environmental drivers of fish species richness. CART begins with the entire heterogeneous sample of 8236 north-temperate

lakes, consisting of both species-poor and species-rich lakes. The goal of CART analysis is to partition the sample according to a "splitting rule" and a "goodness of split criteria." Splitting rules are questions of the form, "Is the environmental variable less or equal to some value?" or, put more generally, "Is $X \leq d$?" where X is an independent variable and d is a constant within the range of that variable. Such questions are used to "split" the sample, and a goodness of split criteria compares different splits and determines which of these will produce the most homogeneous subsamples. With respect to our case study, we wanted to disaggregate the lakes into those with similar values of fish species richness. As an example, Figure 2 is based on output produced by CART (Salford Systems, Inc.) when we set the parent node minimum to 1000 lakes and the terminal node minimum to 200 lakes. The first step was to select the optimal size of the regression tree, defined by the number of terminal nodes. Breiman et al. (1984) suggest the 1-SE rule, whereby the best tree is taken as the smallest tree having an estimated error rate that is within one standard error of the minimum value across all trees. Using 10-fold cross-validation, we selected a final tree with 8 terminal nodes that exhibited a predictive performance of $R=0.65$ between predicted and observed species richness (Figure 2A).

The final regression tree is shown in Figure 2C. The root node at the top of the tree shows that the 8236 study lakes in the initial sample contain, on average, 6.7 fish species. The first split of the root node is based on lake surface area. For lakes less than or equal to 1.5 km^2 in area ("Yes" answer on the left-hand branch of the tree), the average richness is 5.3 ($n = 5511$ lakes). This group is split based on a shoreline perimeter greater than ("No") or less than ("Yes") 3.5 km (average richness = 6.3 vs. 3.9 species). Of the 2322 lakes with smaller shoreline perimeters, the regression tree distinguishes between two terminal nodes based on an annual air temperature threshold of 2.2 °C: node A representing 859 lakes with an average of 3.1 species, and node B representing 1463 lakes with an average of 4.5 species. The parent node containing the 3189 lakes with larger shoreline perimeters is split by a number of additional criteria, including mean monthly precipitation and elevation, to produce terminal nodes C through E, which represent the most homogeneous subgroups that can be partitioned with the given independent variables. For the right-hand split leading off from the root node the same interpretation is used. According to all surrogate splits in the regression tree, we find that surface area, shoreline perimeter, and mean monthly precipitation are the most important predictors of fish species richness (Figure 2B).

Artificial Neural Networks (ANNs)

background and ecological applications

An artificial neural network (ANN), or, more generally, a multilayer perception, is a modeling approach inspired by the way biological nervous systems process complex information. The key element of the ANN is the novel structure of the information processing system, which is composed of a large number of highly interconnected elements called neurons, working in unity to solve specific problems. The concept of ANNs was first introduced in the 1940s (McCulloch and Pitts 1943); however, it was not popularized until the development of the backpropagation training algorithm by Rumelhart et al. (1986). The flexibility of this modeling technique has led to its widespread use in many disciplines such as physics, economics, and biomedicine.

Researchers in ecology have also recognized the potential mathematical utility of neural network algorithms for addressing an array of problems. Previous applications include the modeling of species distributions (Mastrorillo et al. 1997; Özesmi and Özesmi 1999), species diversity (Guégan et al. 1998; Brosse et al. 2001; Olden et al. 2006b), community composition (Olden et al. 2006a), and aquatic primary and secondary production (Scardi and Harding

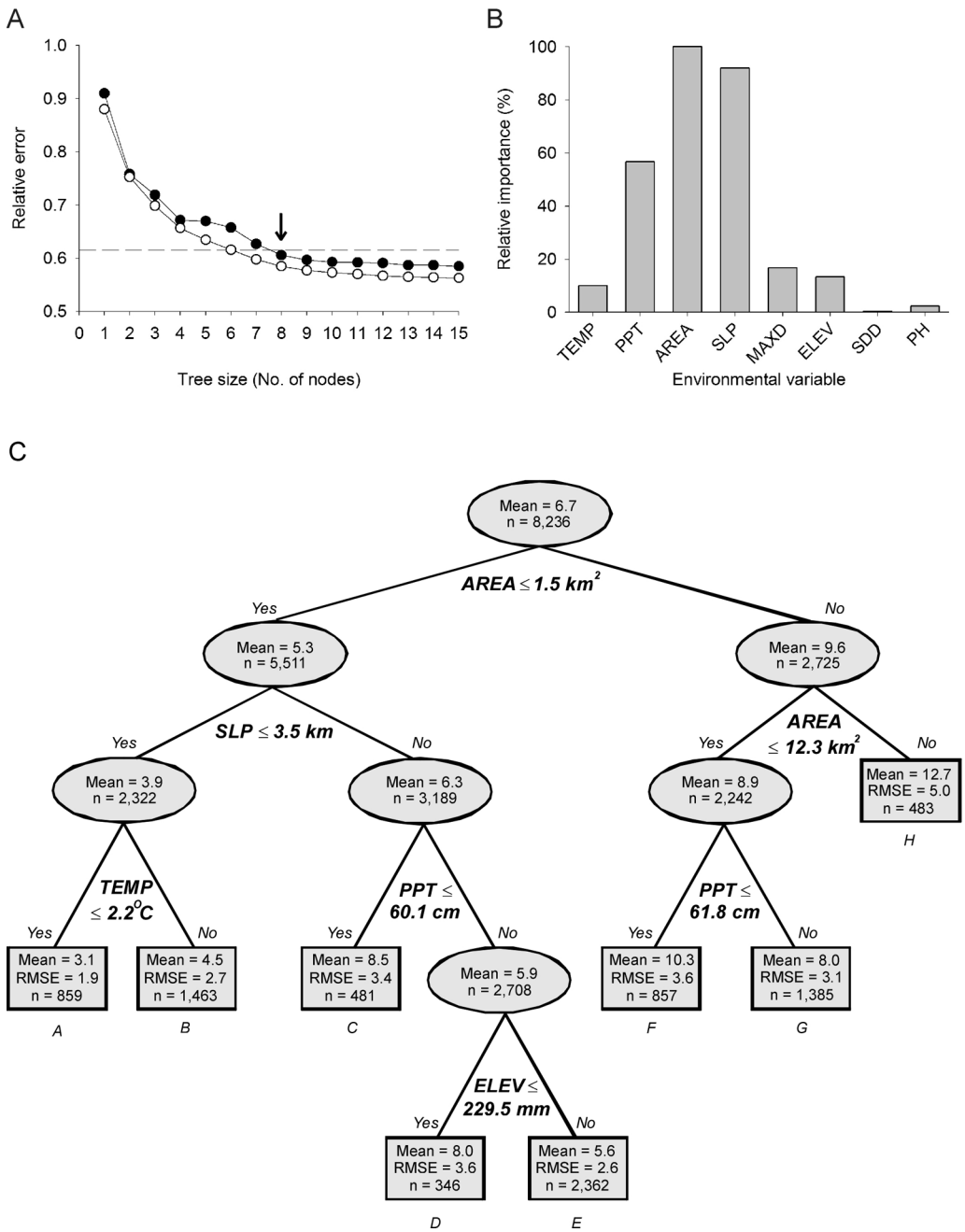

Figure 2. Regression Tree and Associated Results for Predicting Fish Species Richness

Results from the regression tree for predicting fish species richness as a function of environmental characteristics for 8236 north-temperate lakes in Ontario, Canada. (A) 10-fold cross-validation (solid circles) and resubstitution (empty circles) relative error for the regression tree. The dashed line represents + 1-SE of the relative error for the minimum regression tree (i.e., 15 nodes), and the selected tree under the 1-SE rule is indicated by the arrow. (B) Relative importance of the environmental variables for predicting fish species richness (note that values do not sum to 100). Variables include mean monthly air temperature (TEMP) and precipitation (PPT), lake surface area (AREA), total shoreline perimeter (SHP), maximum depth (MAXD), elevation (ELEV), secchi disc depth (SDD), and pH. (C) The final regression tree relating fish species richness to lake environmental characteristics. Node precision is indicated by Root-Mean-Squared-Error.

1999; McKenna 2005). Cornuet et al. (1996) used a neural network to assign individuals to appropriate taxonomic groups using multilocus genotypes. Spitz and Lek (1999) modeled wildlife damage to farmlands, and Thuiller (2003) assessed the potential impacts of climate change on the distribution of tree species in Europe. Other applications have occurred in the fields of water resource management (Maier and Dandy 2000), invasive species biology (Vander Zanden et al. 2004), and pest management (Worner and Gevrey 2006). A collection of ANN applications in ecology is presented in Lek and Guégan (2000), Recknagel (2003), and Özesmi et al. (2006), as well as in special issues of *Ecological Modelling* and *Ecological Informatics* (e.g., Recknagel 2001; Park and Chon 2007).

METHODOLOGY

There are many types of supervised and unsupervised learning methods for ANNs (Bishop 1995). Here we describe the most frequently used method in ecology: the one hidden-layer, supervised, feedforward neural network trained by the back-propagation algorithm. These neural networks are popular in the ecological literature because they are considered to be universal approximators of any continuous function (Hornik et al. 1989). In this section, we will discuss neural network architecture and the back-propagation algorithm used to parameterize the network, and we will describe the various methods available to quantify variable importance.

Network architecture refers to the number and organization of the neurons in the network (see Figure 3 for the general anatomy of a neural network). In the feedforward network, neurons are organized in an input layer, a hidden layer, and an output layer, with each layer containing one or more neurons. Each neuron is connected to all neurons in adjacent layers with an axon; however, neurons within each layer and in nonadjacent layers are not connected. The input layer typically contains p neurons, one neuron representing each of the independent variables x_1 through x_p. The number of neurons in the hidden

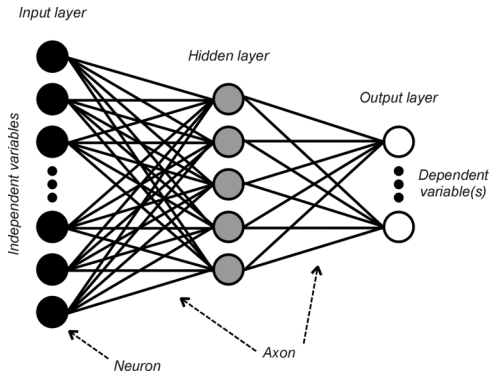

FIGURE 3. THE GENERAL ANATOMY OF AN ARTIFICIAL NEURAL NETWORK

layer can be selected arbitrarily or determined empirically by the investigator to minimize the trade-off between bias and variance (Geman et al. 1992). The addition of hidden neurons increases the ability of a network to approximate any underlying relationship among the variables, i.e., resulting in reduced bias, but also increases the variance of predictions due to overfitting the data. Although mathematical derivations exist for selecting an optimal design (see Bishop 1995), in practice it is common to train networks with different numbers of hidden neurons and to use the performance on a test data set to choose the network that performs the best. For continuous and binary response variables, the output layer commonly contains one neuron, but the number of output neurons can be greater than one if there is more than one response variable or if the response variable is categorical (i.e., a separate neuron for classifying observations into each category). Additional bias neurons with a constant output are also added to the hidden and output layers, although this is not mandatory, as these neurons play a similar role to the intercept term in general linear regression.

Each neuron in the network has an "activity level" that is defined by the value of the incoming signals received from the other neurons connected to it. In turn, each axon in a network is assigned a "connection weight" that reflects the overall intensity of the signal it transmits (i.e., input

to hidden or hidden to output). The activity levels of the input neurons are defined by the values of the predictor variables (Figure 3). The state of each hidden neuron is evaluated locally by calculating the weighted sum of the incoming signals from the neurons of the input layer, which is then subjected to an activation function, i.e., a differentiable function of the neuron's total incoming signal from all input neurons. The same procedure described above is repeated for the axon signals from the hidden layer to the output layer.

Training the neural network typically involves an error back-propagation algorithm that searches for an optimal set of connection weights that produces an output signal with a small error relative to the observed output (i.e., minimizing the fitting criterion). For continuous output variables, the most commonly used criterion is the least-squares error function, whereas for dichotomous output variables, the most commonly used criterion is the cross entropy error function, which is similar to log-likelihood (Bishop 1995). The algorithm adjusts the connection weights in a backwards fashion, layer by layer, in the direction of steepest descent, thus minimizing the error function (this is also called gradient descent). The training of the network is a recursive process where observations from the training data are entered into the network in turn, each time modifying the input-hidden and hidden-output connection weights. This procedure is repeated with the entire training dataset (i.e., each of the n observations) for a number of iterations or epochs until a stopping rule (e.g., error rate) is achieved. Prior to training the network, the independent variables should be converted to z-scores (0 to 1) in order to standardize the measurement scales of the inputs into the network.

Recent efforts have focused on the development of methods for understanding the explanatory contributions of the independent variables in ANNs (Olden and Jackson 2002b). This was, in part, prompted by the fact that neural networks were considered a "black box" approach to modeling ecological data because of the perceived difficulty in understanding their inner workings. Recent studies in the biological sciences have provided a variety of methods for quantifying and interpreting the contributions of the independent variables in neural networks (see Olden and Jackson 2002b; Gevrey et al. 2003; Olden et al. 2004). These approaches utilize the fact that the connection weights between neurons are the linkages between the inputs and the output of the neural network, and, therefore, the relative contribution of each independent variable depends on the magnitude and direction of these connection weights. Input variables with larger connection weights represent greater intensities of signal transfer; they are more important in predicting the output compared to variables with smaller weights. Negative connection weights reduce the intensity or contribution of the incoming signal and negatively affect the output, whereas positive connection weights increase the intensity of the incoming signal and positively affect the output. One method, the connection weight approach, uses the product of the input-hidden and hidden-output connection weights to determine variable importance (Olden et al. 2004). Other approaches include Garson's algorithm (Garson 1991), partial derivatives (Dimopoulos et al. 1995), a sensitivity analysis to determine the spectrum of input variable contributions in the neural network (Lek et al. 1996), and a number of pruning algorithms (Bishop 1995), including a randomization test to remove small connection weights (Olden and Jackson 2002b). Although these approaches can determine the overall influence of each independent variable, the interpretation of interactions among the variables requires the direct examination of the network connection weights (e.g., Özesmi and Özesmi 1999; Olden and Jackson 2001).

STRENGTHS AND WEAKNESSES

There are both advantages and disadvantages to neural networks, and, to discuss this subject properly, we would have to look at each individual type of network. In reference to a back-propagation feedforward approach, ANNs have a number of

advantages over traditional parametric approaches, including: (1) the ability to model nonlinear associations; (2) no requirement of specific assumptions concerning the distributional characteristics of the independent variables (i.e., nonparametric); and (3) the accommodation of variable interactions without a priori specification (Table 1). ANNs also provide a much more flexible way of modeling ecological data. Model complexity can be varied by altering the transfer function or the inner architecture of the network through an increase in the number of hidden neurons or layers to enhance data fitting, or by increasing the number of output neurons to model multiple ecological response variables, such as multiple species (e.g., Özesmi and Özesmi 1999) or entire communities (e.g., Olden 2003; Olden et al. 2006a). It is this flexibility that has likely led to the increased popularity of neural networks in ecology.

Yet ANNs are not without limitations, and there are some specific issues potential users should be aware of. First, neural network models are more complicated to implement, mainly for the reason that the optimization of the network architecture is iterative and is, thus, time consuming. However, this process can be automated and relies on computer time rather than human time to optimize. Second, the performance of a network can be sensitive to the random initial connection weights assigned to the network prior to training. To overcome this, it is recommended that multiple networks based on different initial connection weights be constructed and that a final network that is deemed representative of the population of models be selected. Third, although ANNs are no longer simplistically viewed as a black box approach to modeling data (Olden and Jackson 2002b), the model-building process is still far less transparent than that of more traditional methods, and the ability to explore direct and interactive variable contributions with ANNs will always be more complicated (Table 1).

SOFTWARE

Until recently, the ability of researchers to use neural networks was limited to those with computer programming experience. This is no longer the case. A number of Windows-based programs, modules for commonly used software packages, and libraries for different programming languages are now available. Windows-based programs include BrainMaker (www.calsci.com), EasyNN-plus (www.easynn.com), NeuroSolutions (www.neurosolutions.com), NeuralWare (www.neuralware.com), and SNNS (http://www-ra.informatik.uni-tuebingen.de/SNNS/). Modules and libraries for statistical software packages include Neural Connection for SPSS (www.spss.com), Neural Network module for Statistica (www.statsoft.com), NeuroSolutions for Excel (www.neurosolutions.com), NeuroXL Classfier for Excel (www.neuroxl.com), Enterprise Miner for SAS (www.sas.com), NeuroSolutions for MatLab (www.neurosolutions.com), Neural Network library for S-Plus (http://lib.stat.cmu.edu/S/), and Feed-forward Neural Networks for the R-package (http://cran.r-project.org).

CASE STUDY

ANN methodology begins with a fully connected network composed of axons with completely random connection weights that link the environmental variables (represented by input neurons) to species richness (represented by the output neuron) via the hidden layer of neurons. In this case study, the back-propagation algorithm modified the connection weights in an iterative fashion to maximize the match between predicted and observed levels of species richness in the study lakes. As an example, Figure 4 is based on output produced by using the Neural Network module in MatLab (Mathsoft Inc.) when we set the initial random weights to range between -0.3 and 0.3 and the maximum number of iterations for model convergence to 1000. Optimal network configuration (i.e., optimal number of neurons in the hidden layer of the network) was determined by comparing the performances of different 10-fold cross-validated networks with 1 to 20 hidden neurons in order to choose the number that produced the greatest network performance (Figure 4A). This resulted in a network with 8 input neurons (8

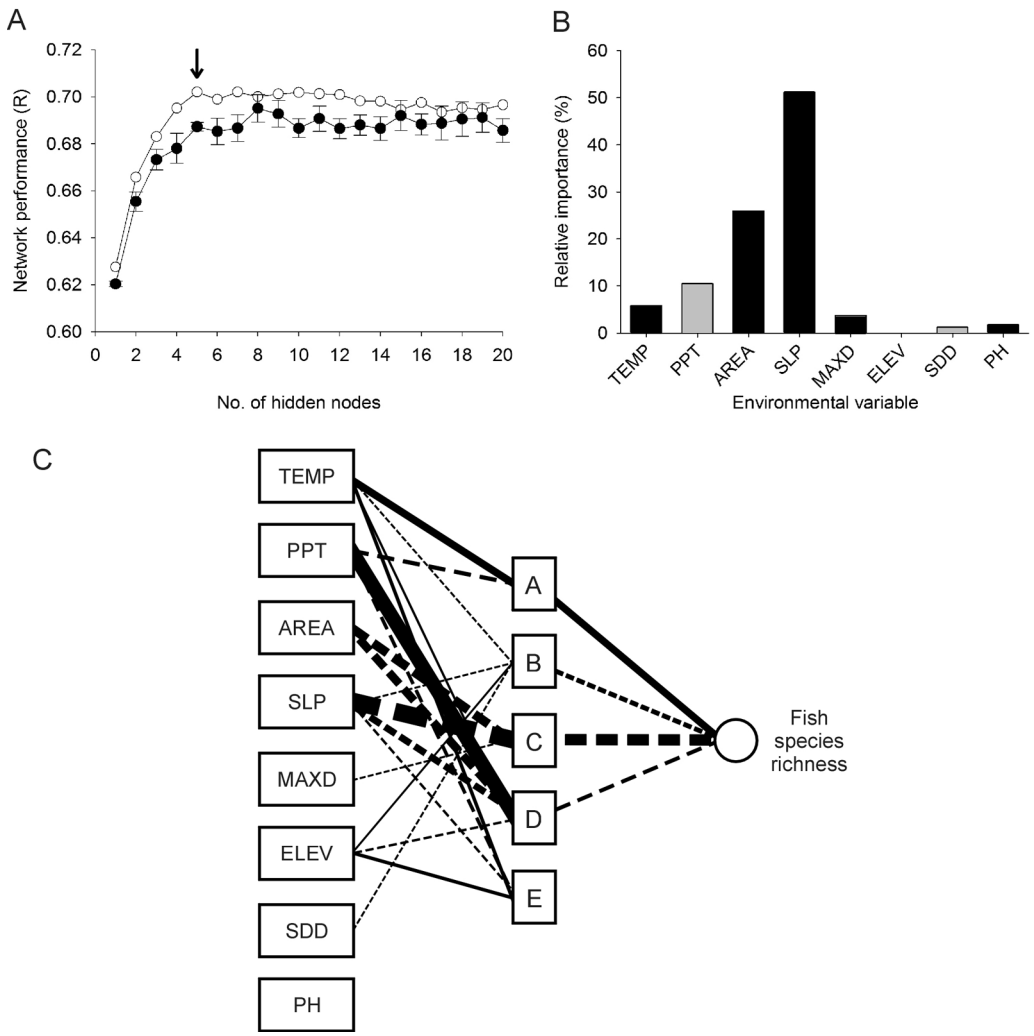

Figure 4. Artificial Neural Network and Associated Results for Predicting Fish Species Richness

Results from the artificial neural network predicting fish species richness as a function of environmental characteristics for 8236 north-temperate lakes in Ontario, Canada. (A) 10-fold cross-validation (solid circles, bars represent +/- 1-SE) and resubstitution (empty circles) relative error for the ANN. The selected network is indicated by the arrow. (B) Relative importance of the environmental variables for predicting fish species richness, where solid and empty bars indicate positive and negative relationships, respectively. (C) The artificial neural network relating fish species richness to lake environmental characteristics. Line thickness is proportional to the magnitude of the axon connection weight, and line type indicates the direction of the interaction between neurons: solid line connections are positive (excitators) and dashed line connections are negative (inhibitors). Only statistically significant connection weights are presented ($P < 0.05$). (Variable codes are presented in the Figure 2 caption.)

environmental variables), 5 hidden neurons, and 1 output neuron (i.e., species richness), and which exhibited a predictive performance of $R=0.70$ between predicted and observed species richness.

The final neural network is presented in Figure 4C. In this figure, the relative magnitudes of the connection weights are represented by line thickness (i.e., thicker lines represent greater weights) and line type rep-

resents the direction of the weights (i.e., solid lines represent positive signals and dashed lines represent negative signals). Positive effects of input variables are depicted by positive input-hidden and positive hidden-output connection weights, or negative input-hidden and negative hidden-output connection weights. Negative effects of input variables are depicted by positive input-hidden and negative hidden-output connection weights, or by negative input-hidden and positive hidden-output connection weights. Thus, the multiplication of the two connection weight directions (positive or negative) indicates the effect that each input variable has on the response variable. Interactions among predictor variables can be identified as input variables with opposing connection weights entering the same hidden neuron.

Individual and interacting influences of the environmental variables on network predictions were examined after we removed nonsignificant, small connection weights (based on $P < 0.05$) using the randomization approach of Olden and Jackson (2002b). The resulting network shows that species richness is positively correlated with mean monthly air temperature via hidden neurons A and B, and negatively correlated with mean monthly precipitation via hidden neurons A and D. The neural network also predicts fish species richness to be greatest in large and deep lakes (neuron C) with high shoreline perimeters (neurons B, C, and D). Focusing on hidden neuron B, we see that shoreline perimeter and lake elevation interact such that the positive influence of shoreline perimeter (reflecting greater littoral zone availability) on species richness (i.e., two negative connection weights) decreases with increasing lake elevation. Figure 4C illustrates the utility in eliminating those connection weights that are small and, therefore, contribute little to network predictions. In this case, we removed 23 out of the 45 possible connection weights. Summing across all connection weights illustrates that shoreline perimeter (positive), lake area (positive), and mean monthly precipitation (negative) exhibited the strongest influences on predictions of species richness from the neural network (Figure 4B).

Evolutionary Computation: Genetic Algorithms and Genetic Programming

background and ecological applications

Evolutionary computation (EC) includes a number of machine learning approaches that can be classified as stochastic optimization tools. In general, these techniques use an aspect of randomization to search for global model optima. More specifically, EC is based on the process of evolution in natural systems and was inspired by a direct analogy to sexual reproduction and Charles Darwin's principle of natural selection (Holland 1975; Goldberg 1989). EC approaches include simulated annealing, evolutionary programming, evolutionary strategies, genetic algorithms, and genetic programming. Because they have been more frequently used in ecological studies, we have confined our discussion here to genetic algorithms (GAs) and genetic programming (GP). In a strict interpretation, GAs refer to the general purpose search algorithms introduced by Holland (1975), which create population-based models that use selection and recombination operators to generate new sample points in a search space (Mitchell 1998). In contrast, GP refers to solutions to the problem in question that take the form of modular computer programs. Running each GP provides a solution to the problem (Koza 1992), whereas, in GAs, the solution is represented by fixed-length character strings (Mitchell 1998). These strings are then interpreted by one or more functions to produce solutions. We will discuss the mechanics of the two approaches in more detail in the following sections.

Ecological applications of GAs and GP have been more limited than most ML techniques, but they are growing in popularity in the natural sciences. D'Angelo et al. (1995) used GAs to model the distribu-

tion of cutthroat and rainbow trout as a function of stream habitat characteristics in the Pacific Northwest of the United States, and they were also applied by Termansen et al. (2006) to model plant species distributions as a function of both climate and land use variables. Genetic programming was used by McKay (2001) to develop spatial models for marsupial density, by Chen et al. (2000) to analyze fish stock-recruitment relationship, and by Muttil and Lee (2005) to model nuisance algal blooms in coastal ecosystems. Ecology's recent interest in EC has been driven, in large part, by the introduction of the Genetic Algorithm for Rule-Set Prediction (GARP) for predicting species distributions (Stockwell and Noble 1992). GARP uses several rule-building methods to build heterogeneous rule sets that describe the ecological niche of a species to environmental data (Stockwell and Peters 1999). The resulting niche model is then projected back onto the landscape to generate a prediction of potential distribution. To date, GARP has been used to model the spatial distribution of numerous species, including the habitat suitability of threatened species (e.g., Anderson and Martínez-Meyer 2004) and invasive species (e.g., Peterson and Vieglais 2001; Peterson 2003; Drake and Lodge 2006), as well as the geography of disease transmission (Peterson 2001). Further advances in ecological niche modeling using EC approaches continue to be developed, such as the WhyWhere algorithm advocated by Stockwell (2006) (but see Peterson 2007). Broader applications of EC methods include their use in conservation planning for biodiversity (Sarkar et al. 2006).

METHODOLOGY

The general anatomies of a genetic algorithm and a genetic program are presented in Figure 5A. In principle, GAs and GP operate on populations of competing solutions to a problem that evolve over time to converge to an optimal solution (Holland 1975). The solutions are loosely represented as "chromosomes" composed of component "genes." Both approaches involve four steps. First, random potential solutions (chromosomes) to the problem are developed. Second, the potential solutions are altered using the processes of reproduction, mutation, and crossover. Third, the new solutions are evaluated to determine their fitness (i.e., how well they solve the problem). Fourth, the most fit or best solutions are selected. Steps two through four, which can be seen as constituting a "generation of solutions," are then repeated using the solutions selected in step four until a stopping criterion is reached. In this way, solutions to a problem evolve through the multiple iterations or generations of the modeling process (Haefner 2005).

Although the principles behind GAs and GP are similar, as mentioned above, the structures of the models are fairly different. In GAs, the components of a solution (i.e., model parameters) are represented as genes on single-vector chromosomes (Figure 5B). The chromosomes change such that solutions evolve over subsequent generations in the model. In each generation, there are two opportunities for chromosomal change: recombination via crossover and random mutation. Recombination mimics sexual recombination in which genetic material from parents is combined to produce related, but different, offspring. First, chromosomes are paired up, often at random. Next, a point along the length of the chromosomes is selected, again at random, and portions of the paired chromosomes are exchanged in an event called crossover (Mitchell 1998). The resulting new chromosomes are regarded as offspring. Further change can occur through mutation events. In each generation of chromosomes, a mutation can occur changing a value from 1 to 0 or vice versa at any individual bit on a chromosome. Mutation acts as an insurance policy against the permanent loss of any simple gene, and it enhances the ability of the genetic algorithm to find the optimal solution.

Recombination and mutation act to create new generations of different solutions. Although we have provided a basic description of these two events, their implementation can vary (Mitchell 1989; Haefner 2005). For example, recombination can be a mandatory event for all chromosomes and, thus, all par-

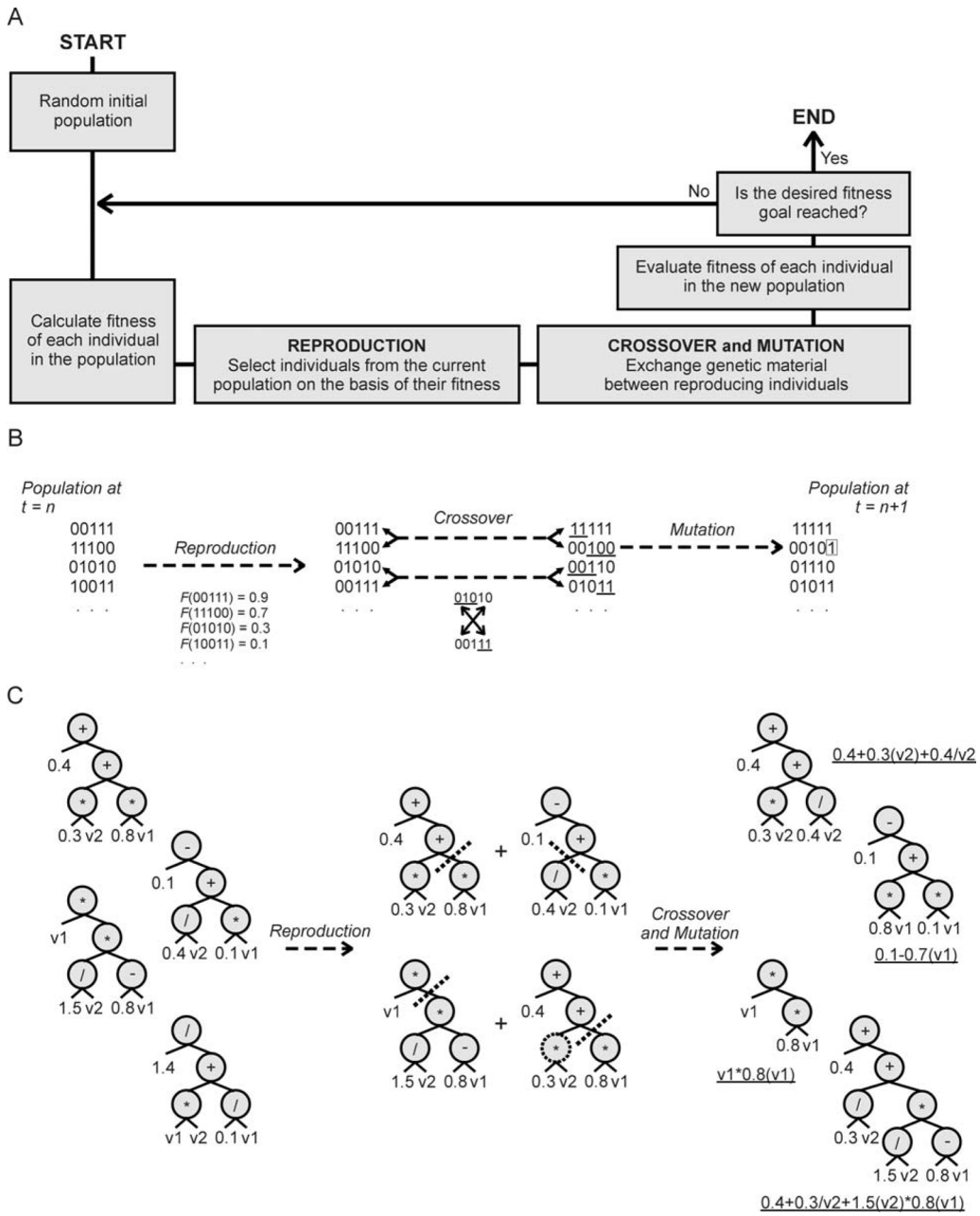

FIGURE 5. THE GENERAL ANATOMY OF A GENETIC ALGORITHM AND A GENETIC PROGRAM

(A) Schematic illustrating the process of evolutionary computation. (B) Illustration of a genetic algorithm (GA) where each individual (4 in total) is represented by a chromosome containing a binary string of 5 genes (parameter 0 or 1). *F* indicates the fitness of each individual. Crossovers occur between underlined sections of the chromosome, and the boxed value represents a mutation event. (C) Illustration of a genetic program (GP) where each individual is represented by a tree-like chromosome of genes representing values and operators. Independent variables 1 and 2 are represented by v1 and v2, respectively. Dashed lines indicate crossover points in the parents, and dashed circles represent mutation events. The underlined statements represent the corresponding regression equations for each individual in the population at t = n+1 (i.e., after crossover and mutation events).

ent chromosomes can be replaced by their offspring, or some proportion of the population can reproduce and be replaced while the rest of the chromosomes remain unchanged. In addition, other events can be used to alter the chromosomes. For example, inversions can be used to reorder portions of a chromosome (Holland 1975). The rates at which mutations and crossover events occur can often be set by the modeler.

After recombination and mutation are completed, the fitness of each chromosome is determined using a fitness function. The function takes the genes of the chromosomes as inputs and produces a value or condition that is then compared to an optimal condition (Holland 1975). The "distance" between the optimal condition and that produced by the chromosome's solution represents its fitness or, put another way, its prediction error. A selection event then takes place in which the chromosomes with higher fitness are selected to comprise the next generation.

As a simple example, consider the problem of selecting the parameters for a regression equation with five variables. The chromosomes would each have five genes corresponding to the five parameters. These five genes are represented on the chromosomes as bit strings. For each generation in the model, recombination and mutation events would produce new sets of chromosomes with different parameters for the regression equation. These new sets of parameters could be evaluated by plugging them into a predetermined regression equation and calculating the mean squared error using the data for which the regression was being developed. Chromosomes that produced lower mean squared errors would be considered more fit than chromosomes with higher mean squared errors.

The chromosomes in GP are modular computer programs in which each module can be seen as a gene (Koza 1992). These chromosomes have a tree-like structure that is in turn interpreted as an equation or a list of commands (Figure 5C). Each node in the tree has a functional value that requires additional arguments, and each terminal branch has a terminal value or command that does not require additional arguments. As an example, again consider a regression equation. The functional values can be addition, subtraction, or division, and terminal values can be numeric (parameters in the regression model) or variables (explanatory variables). Figure 5C shows the tree-like structure of such a chromosome (program) and the corresponding regression equation. As in GAs, a population of chromosomes is evolved through generations of recombination and crossover events. Mutations can occur at any functional value or any terminal value. Crossover breaks can occur at any node in the tree.

In summary, during successive generations in both GAs and GP, the initial population of chromosomes (i.e., strings of problem solutions in GAs or modular programs in GP) advances toward a fitter population by reproduction among members of the previous generation. Selection of the fittest chromosomes makes sure that only the best chromosomes can crossover or mutate, thus advancing their opportunity to find the best solution to the problem. We refer the reader to Haefner (2005) for an overview of evolutionary computation approaches, Goldberg (1989) and Mitchell (1998) for introductory texts on the subject, and Holland (1975) and Koza (1992) for a more advanced treatment of GAs and GP, respectively.

STRENGTHS AND WEAKNESSES

Evolutionary computation has a number of advantages over traditional ecological modeling approaches. First, because EC approaches perform extensive optimization searches, they are able to model nonlinear data and also may be appropriate for situations involving uneven sampling and small sample sizes. This optimal searching amounts to a bottom-up inductive analysis as opposed to the top-down deductive approach of most traditional predictive ecological modeling techniques. A second and related advantage of EC is that it provides ecologists with the ability to model complex relationships using a broad range of model structures and model fitting approaches (Ta-

ble 1). Theoretically, this means that one can have a large amount of control over the design of the models and the execution of the algorithm; however, the ability to exert this flexibility is often limited by the available software. The GARP program perfectly illustrates the flexibility of EC approaches. As mentioned above, GARP uses a set of rules to define species distribution. This rule set may be composed of many different functions or relationships (e.g, logistic functions, Boolean operators), which are assembled in the modeling process. Although the GARP algorithm has a fixed set of rules, in designing a genetic algorithm or genetic program to address a given problem, one can design one's own set of potential functions or relationships. A third advantage of EC approaches is that they were designed as stochastic optimization tools with a relatively broad application in mind.

Nonetheless, EC approaches are not the best techniques for all problems. First, many readily available statistical techniques perform as well as, if not better than, GAs at developing regression and classifications systems. GARP, in particular, has been shown to overpredict species distributions relative to other modeling approaches (Elith et al. 2006; Lawler et al. 2006). Second, there is little theory available to explain GP and little guidance for selecting model parameters. Therefore, the onus, more so than with most statistical approaches, is on the researcher to develop the potential range of model structures. Developing more complex models requires more work on the part of the modeler (Table 1).

There are also a few specific limitations of the two approaches discussed here that are worth mentioning. One specific limitation of genetic algorithms is that they use fixed-length chromosomes, which can limit the potential range of solutions. In our example of the regression equation with five parameters, we were limited in that we had to set the number of parameters in advance. Thus, we were unable to evolve a solution with, for instance, seven or eight parameters. Genetic programming does not have this limitation; the length of a solution is limited only by available computer memory or by the software used to do the modeling. Genetic programming, however, also has a distinct drawback. The solutions obtained from genetic programs are often too complex to easily interpret because they can be long strings of parameters or equations with complicated operators (we will revisit this problem in the case study to follow). Even with some of the more well-developed software, deciphering the computer code that produces the models or understanding the significance of the complex relationships that the model defines can be difficult (Table 1).

SOFTWARE

Although much of the available evolutionary computation software requires some computer programming skill, a number of more user-friendly tools have been developed. Discipulus, which is used in our case study, is available for Windows platforms (http://www.rmltech.com/). The GARP program for Windows is available at http://www.nhm.ku.edu/desktopgarp/, EJC is an "evolutionary computation research system" developed at George Mason University (http://cs.gmu.edu/~eclab/projects/ecj/), and Groovy Java Genetic Programming (JGProg) is available from http://jgprog.sourceforge.net/. At least two packages are available for R-package, including gafit and genalg (http://cran.r-project.org). And finally, there are a number of repositories for computer code for evolutionary computation (e.g., http://www.geneticprogramming.com).

CASE STUDY

We used genetic programming to model species richness in the north-temperate lakes dataset. The modeling was done with the program Discipulus, which is easy to use software designed to efficiently apply genetic programming to regression and classification problems. Discipulus uses both mutations and crossover events within a population of computer programs to evolve solutions. We used the program's default settings, consisting of a population of 500 programs, a mutation rate of 95%, and a crossover frequency of 50%. Each program consists of modules containing both vari-

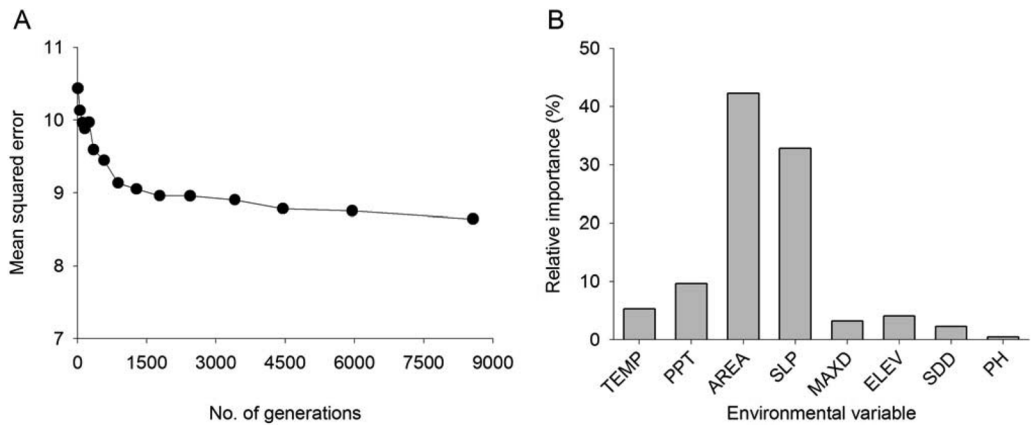

FIGURE 6. GENETIC PROGRAM AND ASSOCIATED RESULTS FOR PREDICTING FISH SPECIES RICHNESS

Results from the genetic program predicting fish species richness as a function of environmental characteristics for 8236 north-temperate lakes in Ontario, Canada. (A) Cross-validation relative error for the GP according to mean-squared-error. (B) Average relative importance of the environmental variables for predicting fish species richness based on the top 30 GPs. (Variable codes are presented in the Figure 2 caption.)

ables and operators. Discipulus allows the use of 11 different types of operators including addition, multiplication, subtraction, division, trigonometric, and Boolean operators. We used a total of 23 different potential operators to model fish species richness. To build and evaluate the models, we randomly divided the lakes dataset into three roughly equal parts corresponding to training, validation, and applied datasets. The training and validation datasets were used in model building and the applied dataset was used to provide a semi-independent assessment of model performance.

The best performing model converged after 9000 generations of the algorithm (Figure 6A) and had an $R=0.69$ between predicted and observed species richness for the semi-independent applied dataset. The two most influential variables in the top 30 models were lake area and shoreline perimeter, followed by mean monthly precipitation and air temperature (Figure 6B). As discussed previously, one of the disadvantages of using genetic programs is that the models they produce are often large and difficult to interpret. The model produced in our case study was no exception. For that reason, we have not provided any details about the structure of the GP model.

CONCLUSION

Machine learning approaches can facilitate greater understanding and prediction in the ecological sciences. The purpose of this review is to broaden the exposure of ecologists to ML by illustrating how CART, ANN, and EC can be used to address complex problems. In doing so, we hope to have introduced ecologists to a set of viable alternatives to more traditional statistical approaches. Although the application of ML approaches in ecology has increased in recent years, the growth has been relatively modest compared to other disciplines, and there remains a good degree of skepticism with respect to its role in quantitative analyses. Currently, the majority of ecologists lack the computational background needed to operate the software that implements these approaches (Fielding 1999), and, as a result, many ecologists may be hesitant to invest their time in learning extensive program code language and syntax. With the increasing popularity of these approaches, however, more user-friendly software is rapidly being developed. Such software (examples of which are listed throughout this review) will increase ML usage and awareness among ecologists and

promote the advancement of these analytical methods.

Machine learning methods are powerful tools for prediction and explanation, and they will enhance our ability to model ecological systems. They are not, however, a solution to all ecological modeling problems. No one ML approach will be best suited to addressing all problems nor will ML approaches always be preferable to traditional statistical approaches. Although ML methods are generally more flexible with respect to modeling complex relationships and messy datasets, the models they produce are often more difficult to interpret, and the modeling process itself is often far from transparent.

Although ML technologies strengthen our ability to model ecological phenomena, advances in understanding the fundamental processes underlying those phenomena are clearly critical as well. Some argue that ML methods attempt to eliminate the need for ecological intuition during the data analysis process; we strongly disagree. Human intuition cannot be entirely eliminated because the analyst must specify how the data are to be represented and what mechanisms will be used to search for a characterization of the problem. In this respect, ML should be viewed as an attempt to automate parts of the modeling process, not replace it (Olden et al. 2006b). We hope our review of machine learning methods will allow more ecologists to add these tools to their repertoire of quantitative expertise, and that it will provide them with a basis for making informed decisions about applying machine learning approaches or more traditional statistical approaches in their future modeling endeavors.

ACKNOWLEDGMENTS

We thank the students of the Ecological Informatics class at Colorado State University for motivating us to write this paper. Jodi Whittier and two anonymous referees provided insightful comments on the manuscript. This research was supported in part by David H. Smith Conservation Postdoctoral Scholarships to J. D. Olden and J. J. Lawler. J. D. Olden conceived and developed the idea for the manuscript, J. D. Olden and J. J. Lawler conducted the data analysis, and J. D. Olden, J. J. Lawler, and N. L. Poff wrote the manuscript.

REFERENCES

Anderson R. P., Martínez-Meyer E. 2004. Modeling species' geographic distributions for preliminary conservation assessments: an implementation with the spiny pocket mice (*Heteromys*) of Ecuador. *Biological Conservation* 116(2):167–179.

Anderson M. C., Watts J. M., Freilich J. E., Yool S. R., Wakefield G. I., McCauley J. F., Fahnestock P. B. 2000. Regression-tree modeling of desert tortoise habitat in the central Mojave desert. *Ecological Applications* 10(3):890–900.

Austin M. 2007. Species distribution models and ecological theory: a critical assessment and some possible new approaches. *Ecological Modelling* 200(1–2):1–19.

Bell J. F. 1999. Tree-based methods. Pages 89–105 in *Machine Learning Methods for Ecological Applications*, edited by A. H. Fielding. Boston (MA): Kluwer Academic.

Bishop C. M. 1995. *Neural Networks for Pattern Recognition*. Oxford (UK): Clarendon Press.

Breiman L., Friedman J., Olshen R. A., Stone C. J. 1984. *Classification and Regression Trees*. Belmont (CA): Wadsworth International Group.

Brosse S., Lek S., Townsend C. R. 2001. Abundance, diversity, and structure of freshwater invertebrates and fish communities: an artificial neural network approach. *New Zealand Journal of Marine and Freshwater Research* 35(1):135–145.

Chen D. G., Hargreaves N. B., Ware D. M., Liu Y. 2000. A fuzzy logic model with genetic algorithm for analyzing fish stock-recruitment relationships. *Canadian Journal of Fisheries and Aquatic Science* 57(9):1878–1887.

Cho E., Chon T.-S. 2006. Application of wavelet analysis to ecological data. *Ecological Informatics* 1(3): 229–233.

Clark J. S., Carpenter S. R., Barber M., Collins S., Dobson A., Foley J. A., Lodge D. M., Pascual M., Pielke R., Jr., Pizer W., Pringle C., Reid W. V., Rose K. A., Sala O., Schlesinger W. H., Wall D. H., Wear D. 2001. Ecological forecasts: an emerging imperative. *Science* 293:657–660.

Cornuet J. M., Aulagnier S., Lek S., Franck P., Solignac M. 1996. Classifying individuals among infraspecific taxa using microsatellite data and neural networks. *Comptes rendus de l'Académie des sciences, Série III, Sciences de la vie* 319(12):1167–1177.

Cushing J. B., Wilson T. 2005. Eco-informatics for

decision makers advancing a research agenda. Pages 325–334 in *Data Integration in the Life Sciences: Second International Workshop, DILS 2005, San Diego, CA, USA, July 20–22, 2005, Proceedings,* Lecture Notes in Computer Science, Volume 3615, edited by B. Ludäscher and L. Raschid. Berlin (Germany): Springer-Verlag.

Cutler D. R., Edwards T. C., Jr., Beard K. H., Cutler A., Hess K. T., Gibson J., Lawler J. J. 2007. Random forests for prediction in ecology. *Ecology* 88(11): 2783–2792.

D'Angelo D. J., Howard L. M., Meyer J. L., Gregory S. V., Ashkenas L. R. 1995. Ecological uses for genetic algorithms: predicting fish distributions in complex physical habitats. *Canadian Journal of Fisheries and Aquatic Sciences* 52:1893–1908.

De'ath G. 2007. Boosted trees for ecological modeling and prediction. *Ecology* 88(1):243–251.

De'ath G., Fabricius K. E. 2000. Classification and regression trees: a powerful yet simple technique for ecological data analysis. *Ecology* 81(11):3178–3192.

Dimopoulos Y., Bourret P., Lek S. 1995. Use of some sensitivity criteria for choosing networks with good generalization. *Neural Processing Letters* 2:1–4.

Drake J. M., Lodge D. M. 2006. Forecasting potential distributions of nonindigenous species with a genetic algorithm. *Fisheries* 31:9–16.

Drake J. M., Randin C., Guisan A. 2006. Modelling ecological niches with support vector machines. *Journal of Applied Ecology* 43(3):424–432.

Elith J., Graham C. H., Anderson R. P., Dudík M., Ferrier S., Guisan A., Hijmans R. J., et al. 2006. Novel methods improve prediction of species' distributions from occurrence data. *Ecography* 29(2): 129–151.

Elith J., Leathwick J. 2007. Predicting species distributions from museum and herbarium records using multiresponse models fitted with multivariate adaptive regression splines. *Diversity and Distributions* 13(3):265–275.

Ferrier S., Guisan A. 2006. Spatial modelling of biodiversity at the community level. *Journal of Applied Ecology* 43(3):393–404.

Fielding A. H., editor. 1999. *Machine Learning Methods for Ecological Applications.* Boston (MA): Kluwer Academic Publishers.

Fielding A. H., Bell J. F. 1997. A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental Conservation* 24(1):38–49.

Garson G. D. 1991. Interpreting neural-network connection weights. *Artificial Intelligence Expert* 6(4): 46–51.

Geman S., Bienenstock E., Doursat R. 1992. Neural networks and the bias/variance dilemma. *Neural Computation* 4(1):1–58.

Gevrey M., Dimopoulos I., Lek S. 2003. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological Modelling* 160(3):249–264.

Goldberg D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading (MA): Addison-Wesley.

Green J. L., Hastings A., Arzberger P., Ayala F. J., Cottingham K. L., Cuddington K., Davis F., Dunne J. A., Fortin M.-J., Gerber L., Neubert M. 2005. Complexity in ecology and conservation: mathematical, statistical, and computational challenges. *BioScience* 55(6):501–510.

Guégan J.-F., Lek S., Oberdorff T. 1998. Energy availability and habitat heterogeneity predict global riverine fish diversity. *Nature* 391:382–384.

Guisan A., Zimmermann N. E. 2000. Predictive habitat distribution models in ecology. *Ecological Modelling* 135(2–3):147–186.

Haefner J. W. 2005. *Modeling Biological Systems: Principles and Applications.* Second Edition. New York: Springer.

Hastie T., Tibshirani R., Friedman J. H. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York: Springer.

Hogeweg P. 1988. Cellular automata as a paradigm for ecological modeling. *Applied Mathematics and Computation* 27(1):81–100.

Holland J. H. 1975. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* Ann Arbor (MI): University of Michigan Press.

Hopfield J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences USA* 79(8):2554–2558.

Hornik K., Stinchcombe M., White H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2(5):359–366.

Iverson L. R., Prasad A. M. 1998. Predicting abundance of 80 tree species following climate change in the Eastern United States. *Ecological Monographs* 68(4):465–485.

Iverson L. R., Prasad A. M. 2007. Using landscape analysis to assess and model tsunami damage in Aceh province, Sumatra. *Landscape Ecology* 22(3): 323–331.

Kohonen T. 2001. *Self-Organizing Maps.* Berlin (Germany) and New York: Springer-Verlag.

Koza J. R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge (MA): MIT Press.

Lamon E. C., III, Stow C. A. 1999. Sources of variability in microcontaminant data for Lake Michigan salmonids: statistical models and implications for trend detection. *Canadian Journal of Fisheries and Aquatic Sciences* 56(Supplement):71–85.

Lawler J. J., White D., Neilson R. P., Blaustein A. R.

2006. Predicting climate-induced range shifts: model differences and model reliability. *Global Change Biology* 12(8):1568–1584.

Lek S., Delacoste M., Baran P., Dimopoulos I., Lauga J., Aulagnier S. 1996. Application of neural networks to modelling nonlinear relationships in ecology. *Ecological Modelling* 90(1):39–52.

Lek S., Guégan J.-F. 2000. *Artificial Neuronal Networks: Application to Ecology and Evolution*. Berlin (Germany): Springer-Verlag.

Levin S. A. 1998. Ecosystems and the biosphere as complex adaptive systems. *Ecosystems* 1(5):431–436.

Maier H. R., Dandy G. C. 2000. Neural networks for the prediction and forecasting of water resource variables: a review of modelling issues and applications. *Environmental Modelling and Software* 15(1):101–124.

Mastrorillo S., Lek S., Dauba F., Belaud A. 1997. The use of artificial neural networks to predict the presence of small-bodied fish in a river. *Freshwater Biology* 38(2):237–246.

McCulloch W. S., Pitts W. 1943. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5:115–133.

McKay R. I. 2001. Variants of genetic programming for species distribution modelling—fitness sharing, partial functions, population evaluation. *Ecological Modelling* 146(1–3):231–241.

McKenna J. E., Jr. 2005. Application of neural networks to prediction of fish diversity and salmonid production in the Lake Ontario basin. *Transactions of the American Fisheries Society* 134(1):28–43.

Mercado-Silva N., Olden J. D., Maxted J. T., Hrabik T. R., Vander Zanden M. J. 2006. Forecasting the spread of invasive rainbow smelt in the Laurentian Great Lakes region of North America. *Conservation Biology* 20(6):1740–1749.

Minns C. K. 1989. Factors affecting fish species richness in Ontario lakes. *Transactions of the American Fisheries Society* 118:533–545.

Mitchell M. 1998. *An Introduction to Genetic Algorithms*. Cambridge (MA): MIT Press.

Morgan J. N., Sonquist J. A. 1963. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association* 58(302):415–434.

Muttil N., Lee J. H. W. 2005. Genetic programming for analysis and real-time prediction of coastal algal blooms. *Ecological Modelling* 189(3–4):363–376.

O'Connor R. J., Jones M. T., White D., Hunsaker C., Loveland T., Jones B., Preston E. 1996. Spatial partitioning of environmental correlates of avian biodiversity in the conterminous United States. *Biodiversity Letters* 3(3):97–110.

Olden J. D. 2003. A species-specific approach to modeling biological communities and its potential for conservation. *Conservation Biology* 17(3):854–863.

Olden J. D., Jackson D. A. 2001. Fish-habitat relationships in lakes: gaining predictive and explanatory insight by using artificial neural networks. *Transactions of the American Fisheries Society* 130(5):878–897.

Olden J. D., Jackson D. A. 2002a. A comparison of statistical approaches for modelling fish species distributions. *Freshwater Biology* 47(10):1976–1995.

Olden J. D., Jackson D. A. 2002b. Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling* 154(1–2):135–150.

Olden J. D., Joy M. K., Death R. G. 2004. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling* 78(3–4):389–397.

Olden J. D., Joy M. K., Death R. G. 2006a. Rediscovering the species in community-wide modeling. *Ecological Applications* 16(4):1449–1460.

Olden J. D., Poff N. L., Bledsoe B. P. 2006b. Incorporating ecological knowledge into ecoinformatics: an example of modeling hierarchically structured aquatic communities with neural networks. *Ecological Informatics* 1(1):33–42.

Özesmi S. L., Özesmi U. 1999. An artificial neural network approach to spatial habitat modelling with interspecific interaction. *Ecological Modelling* 116(1):15–31.

Özesmi S. L., Tan C. O., Özesmi U. 2006. Methodological issues in building, training, and testing artificial neural networks in ecological applications. *Ecological Modelling* 195(1–2):83–93.

Park Y.-S., Chon T.-S. 2007. Biologically-inspired machine learning implemented to ecological informatics. *Ecological Modelling* 203(1–2):1–7.

Peters R. H. 1991. *A Critique for Ecology*. Cambridge (UK): Cambridge University Press.

Peterson A. T. 2001. Predicting species' geographic distributions based on ecological niche modeling. *Condor* 103(3):599–605.

Peterson A. T. 2003. Predicting the geography of species' invasions via ecological niche modeling. *Quarterly Review of Biology* 78(4):419–433.

Peterson A. T. 2007. Why not WhyWhere: the need for more complex models of simpler environmental spaces. *Ecological Modelling* 203(3–4):527–530.

Peterson A. T., Vieglais D. A. 2001. Predicting species invasions using ecological niche modeling: new approaches from bioinformatics attack a pressing problem. *BioScience* 51(5):363–371.

Phillips S. J., Anderson R. P., Schapire R. E. 2006. Maximum entropy modeling of species geographic distributions. *Ecological Modelling* 190(3–4):231–259.

Recknagel F. 2001. Applications of machine learning to ecological modelling. *Ecological Modelling* 146(1–3):303–310.

Recknagel F. 2003. *Ecological Informatics: Understanding*

Ecology by Biologically-Inspired Computation. Berlin (Germany) and New York: Springer-Verlag.

Ripley B. D. 1996. *Pattern Recognition and Neural Networks*. Cambridge (UK): Cambridge University Press.

Rollins M. G., Keane R. E., Parsons R. A. 2004. Mapping fuels and fire regimes using sensing, ecosystem simulation, and gradient modeling. *Ecological Applications* 14(1):75–95.

Rumelhart D. E., Hinton G. E., Williams R. J. 1986. Learning representations by back-propagating error. *Nature* 323:533–536.

Salski A., Sperlbaum C. 1991. A fuzzy logic approach to modeling in ecosystem research. Pages 520–527 in *Uncertainty in Knowledge Bases, 3rd International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU '90, Paris, France, July 2–6, 1990,* Lecture Notes in Computer Science, Volume 521, edited by B. Bouchon-Meunier et al. Berlin (Germany): Springer-Verlag.

Sarkar S., Pressey R. L., Faith D. P., Margules C. R., Fuller T., Stoms D. M., Moffett A., Wilson K. A., Williams K. J., Williams P. H., Andelman S. 2006. Biodiversity conservation planning tools: present status and challenges for the future. *Annual Review of Environment and Resources* 31:123–159.

Scardi M., Harding L. W., Jr. 1999. Developing an empirical model of phytoplankton primary production: a neural network case study. *Ecological Modelling* 120(2–3):213–223.

Smith S. J., Iverson S. J., Bowen W. D. 1997. Fatty acid signatures and classification trees: new tools for investigating the foraging ecology of seals. *Canadian Journal of Fisheries and Aquatic Sciences* 54(6):1377–1386.

Spitz F., Lek S. 1999. Environmental impact prediction using neural network modelling: an example in wildlife damage. *Journal of Applied Ecology* 36(2):317–326.

Stockwell D. R. B. 2006. Improving ecological niche models by data mining large environmental datasets for surrogate models. *Ecological Modelling* 192(1–2):188–196.

Stockwell D. R. B., Noble I. R. 1992. Induction of sets of rules from animal distribution data: a robust and informative method of analysis. *Mathematics and Computers in Simulation* 33(5–6):385–390.

Stockwell D. R. B., Peters D. 1999. The GARP modelling system: problems and solutions to automated spatial prediction. *International Journal of Geographical Information Science* 13(2):143–158.

Sutton C. D. 2005. Classification and regression trees, bagging, and boosting. Pages 303–329 in *Handbook of Statistics: Data Mining and Data Visualization*, Volume 24, edited by C. R. Rao et al. Amsterdam (the Netherlands): Elsevier Publishing.

Termansen M., McClean C. J., Preston C. D. 2006. The use of genetic algorithms and Bayesian classification to model species distributions. *Ecological Modelling* 192(3–4):410–424.

Thuiller W. 2003. BIOMOD—optimizing predictions of species distributions and projecting potential future shifts under global change. *Global Change Biology* 9(10):1353–1362.

Torres L. G., Rosel P. E., D'Agrosa C., Read A. J. 2003. Improving management of overlapping bottlenose dolphin ecotypes through spatial analysis and genetics. *Marine Mammal Science* 19(3):502–514.

Usio N. 2007. Endangered crayfish in northern Japan: distribution, abundance and microhabitat specificity in relation to stream and riparian environment. *Biological Conservation* 134(4):517–526.

Vander Zanden M. J., Olden J. D., Thorne J. H., Mandrak N. E. 2004. Predicting occurrences and impacts of smallmouth bass introductions in north temperate lakes. *Ecological Applications* 14(1):132–148.

Worner S. P., Gevrey M. 2006. Modelling global insect pest species assemblages to determine risk of invasion. *Journal of Applied Ecology* 43(5):858–867.