2017

Temporal Feature Selection with Symbolic Regression

Christopher Winter Fusting
University of Vermont

Temporal Feature Selection with Symbolic Regression

A Thesis Presented

by

Christopher W. Fusting

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Master of Science
Specializing in Mathematics

October, 2017

Defense Date: July 31, 2017
Thesis Examination Committee:

Josh Bongard, Ph.D., Advisor
Christian Skalka, Ph.D., Chairperson
James Bagrow, Ph.D.
Cynthia J. Forehand, Ph.D., Dean of Graduate College

Abstract

Building and discovering useful features when constructing machine learning models is the central task for the machine learning practitioner. Good features are useful not only in increasing the predictive power of a model but also in illuminating the underlying drivers of a target variable. In this research we propose a novel feature learning technique in which Symbolic regression is endowed with a "Range Terminal" that allows it to explore functions of the aggregate of variables over time. We test the Range Terminal on a synthetic data set and a real world data in which we predict seasonal greenness using satellite derived temperature and snow data over a portion of the Arctic. On the synthetic data set we find Symbolic regression with the Range Terminal outperforms standard Symbolic regression and Lasso regression. On the Arctic data set we find it outperforms standard Symbolic regression, fails to beat the Lasso regression, but finds useful features describing the interaction between Land Surface Temperature, Snow, and seasonal vegetative growth in the Arctic.

For every curious child.

Acknowledgements

First I'd like to thank Chris Danforth for bringing me into the Mathematics program, funding me, and helping me in any way he could along the way. I'd also like to thank my fiance Brittany Lippard for being an ardent supporter of my endeavors, creating a warm home full of delicious meals, and listening to me babble on about Analysis for two years. I'd like to thank Ryan Gallagher with whom I attended nearly every class for helping me fill in the many mathemetical holes in my formal education. I'd also like to thank Sam Kriegman for constantly arguing with me late into the night, even though he didn't like to talk on the phone, and joking around as much as I did. Speaking of joking around I'd like to thank Devan Gokhale, Collin Cappelle, Andy Reagan, Justin Foster, Tyler Gray, and everyone else in my labs for being good friends. I'd like to thank Peter Dodds, Richard Foote, and Jeff Buzas for providing a very enjoying learning environment and spending time with me during office hours. Finally I'd like to thank my research team: Chris Skalka who took a chance and funded this research at the last minute, Josh Bongard who took time to offer expert advice on evolutionary methods and welcomed me into his lab, Marcin Szubert who provided me with an invaluable code base and gave me real intuition and resources to help me learn Genetic Programming, and Tim Stevens who endured my relentless code reviews while helping us fetch the Snow data. There are many others to thank, you know who you are.

Table of Contents

Chapter 1
Literature Review

Introduction

Many machine learning (ML) problems have periodic observations, each considered a variable, of a sensor over time that are predictive of the value of a target variable. In addition to the variables themselves having predictive power, sometimes a function of the aggregate of a range of variables over time also have predictive power. We encountered this situation when trying to predict primary productivity in the Arctic using satellite derived Land Surface Temperature (LST) and Snow data in the Arctic. To see why a function of aggregated variables may be predictive we provide some background: The 2015 Arctic Report Card [21] showed an unusual downward trend in the amount of greenness across the Arctic circle and the northern reaches of Eurasia. The cause of the recent browning trend is not known, although field studies suggest extreme temperature and other events have had a significant impact [32]. Bjerke et al. [5] found fourteen weather events leading to plant stress from October of 2011 to September 2012 in the Nordic Arctic Region. These include: severe cold spells when plants had little insulating snow to protect them, unusually warm temperatures during Winter months causing Spring like development and subsequent death of plants, flooding which destroyed plants and shrubs in low lying areas, and others. Bokhorst et al. [7] simulated warming events in mid-Winter using heating lamps and underground cables and found that these conditions caused snow melt, leaving plants exposed to extreme cold. This caused vegetation death and hence less primary productivity the following Summer. These events are in contrast to the overall trend which shows increased primary productivity in the Arctic.

To capture extreme events we needed to be able to search over a range of time during which they may occur. We hypothesized that functions such as the minimum, maximum, mean, sum, variance, and skew, when taken over time, would capture the occurrence of extreme events. As an an example take the previously described Winter

warming event. This event could be described by the maximum of temperature variables during the Winter. In this paper we develop a novel feature learning technique that develops features representing the function of the aggregate of variables over time. To motivate our development choices we first discuss some background on how features are commonly constructed.

Feature Engineering and Selection

The process of manually creating features is known as "feature engineering" or "feature construction" and depending on the number of variables in a data set can be quite laborious. Feature engineering is often done in tandem with, but should not be confused with, feature selection in which the dimensionality of a data set is reduced. Often this is in an effort to reduce computational complexity or if the goal is to infer meaning from the model, identify highly predictive features. There are many approaches to feature engineering and selection, some of which we discuss here. For a good introduction see Guyon [17].

The process of feature engineering largely depends on the domain expertise of an analyst and the breadth of scientific knowledge available in the field. If little or nothing is known about the interaction of the predictor variables (predictors) with the target variable, features may be randomly constructed by taking the cross product of the predictors and (or) applying functions such as $\sqrt{x}, \exp(x), x^p$ to the predictors and then evaluating their performance in models. Selecting performant features can be achieved through scoring. A simple method is to compute the Pearson correlation coefficient between features and the target (which is performed implicitly in linear regression). However naive applications of this idea such as in Stepwise regression have been widely criticized [46] for being biased, unreliable, and badly implemented in software. More sophisticated approaches to variance decomposition have been show to be fruitful but can be computationally expensive [16].

If domain knowledge is available this process can be expedited by developing features using scientific insight. Building features in this manner, however, depends on the depth of scientific knowledge available in the model's domain and as such many useful features may not be discovered, or may be discovered only occasionally through random chance.

Once constructed the M features $\boldsymbol{\phi} = (\phi_0, \cdots, \phi_{M-1})^T$ called basis functions can be input into a linear basis function model:

$$y(\boldsymbol{w}, \boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{1.1}$$

where $w = (w_0, \cdots, w_{M-1})$ is a vector of weights and $\phi_0(\boldsymbol{x}) = 1$ is the bias term. Classification models are built similarly. From this starting point one can explore

various linear methods such as Ridge, Lasso, ElasticNet, Bayesian, and other sophisticated regression and classification techniques [4, 12]. These techniques employ a regularization parameter such that large coefficients are penalized. In all cases this leads to a more general model but in the case of Lasso [47] regression some coefficients are in fact zeroed out which leads to a more parsimonious model and performs additional feature selection implicitly.

Feature Learning

An alternative to feature engineering is "feature learning", a process in which features are learned automatically using the available variables. There has been recognition in the last decade that the amount of data and number of potential predictors is increasing dramatically, necessitating the need and development of feature learning techniques [10].

Unsupervised methods such as Principle Component Analysis (PCA), kernel PCA [39] (for nonlinear interactions), Principle Component Regression (PCR), and others can be useful in both feature selection and engineering but the resulting features are difficult to interpret. Other unsupervised methods such as k-means clustering [18] and Latent Dirichlet Allocation [6] produce features that are more interpretable, although the later is usually used only for text mining.

Many machine learning techniques perform feature learning as a side effect. One powerful approach is to define the features implicitly using a kernel as in Gaussian Process regression (GPR) where a kernel is defined in terms of the dot product of the feature vectors

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x'})$$

The basis functions $\boldsymbol{\phi}$ can be constructed by the choice of the kernel. For example GPR is equivalent to Bayesian regression when the kernel is linear $k(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{x}^T \boldsymbol{x'}$. Another example is the polynomial kernel $k(\boldsymbol{x}, \boldsymbol{x'}) = (\boldsymbol{x^T x'} + c)^n$ which expands out to an nth degree polynomial. This choice of kernel is equivalent to Polynomial regression (although the computation is not bound by the degree n). Other kernels such as the radial basis function (RBF) $\exp(-||\boldsymbol{x} - \boldsymbol{x'}||^2/2\sigma^2)$ are commonly used to express the idea that variables close to each other in value behave similarly. Expansion of the RBF leads to the dot product of feature vectors of infinite length, a remarkable ability of Gaussian Processes. Despite their power it is difficult to pinpoint exactly what features are playing the most important role beyond the expansion and analysis of the kernel. See Bishop [4] for a good introduction to GPR and Rasmussen [34] for a complete exposition.

Another approach that has gained popularity and quickly become the modern state of the art is Neural Networks (NN) [15]. Neural Networks were originally made

to mimic the brain of biological systems although today much of their implementations differ from what we know of living brains. The formulation of a neural network does not include the idea of basis functions as features in the same way as in linear and kernel methods. Even a simple fully connected feed forward single layer classification network takes a form difficult to evaluate for feature meaning

$$y_k(\boldsymbol{x}, \boldsymbol{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i\right)\right)$$

where the superscript indicates the layer of the network, σ is the sigmoid function, h is a nonlinear activation function (such as a sigmoid, logistic, or tanh), D is the number of variables, and $x_0 = 1$ such that the bias parameter is absorbed into the weights. Research in the area of interpreting neural networks is growing but still nascent [26, 54].

Another machine learning technique with implicit feature learning is Symbolic regression (SR) [23]. Symbolic regression has been used extensively to discover mathematical expressions from data, including finding physical laws [37] and reverse engineering dynamical systems [8]. Rather than defining a model and fitting parameters to data, SR builds a mathematical expression using data as a guide. Although SR models are not always fit using Genetic Programming (GP) [29], it is the most common method and we will describe it here. Programmed Darwinian evolution guides the construction of a population of mathematical expressions made up of operators and functions such as $+, -, *, \div, \ln, \exp$, etc. These operators act on terminals which are usually the variables and other features. Expressions that perform well given training data and a fitness metric mate and pass on their characteristics to their children. The fitness metric usually takes the form of mean absolute or squared error although other metrics such as the Pearson Correlation have been used successfully [43]. After some number of generations the remaining expressions are selected as the winning models.

Because SR makes no assumptions about the form of the model a priori it is capable of capturing nonlinear interactions between predictors and the target without explicitly (in the case of linear regression) or implicitly (in the case of GP regression) specifying a basis $\boldsymbol{\phi}$. However in contrast to NN it is possible to explicitly optimize SR to keep the mathematical expressions it evolves simple which makes interpretation easier. Because of this SR is an ideal feature learning tool.

In this work we introduce a terminal for the SR expression tree, the Range Terminal, which calculates a function of an aggregate of variables over time. This terminal allows mathematical expressions to capture signals over a range of temporal predictor variables. In addition it provides a simplifying symbol that makes the resulting mathematical expressions easily interpretable. We also propose a simple feature learning method:

1. SR with the Range Terminal is used to evolve mathematical expressions.
2. Expression trees are broken up into features.
3. Features are selected and weighted using Lasso regression.

Related Research

Variable and feature selection using Symbolic Regression has been explored in many studies [30, 41, 45]. Perhaps the first example of SR as a feature learning tool is the automatically defined function (ADF) [23]. Krawiec explores a feature construction technique in which each individual in the population is made up of isolated features that are evolved separately [24]. In an extension to the method highly successful features are protected from further evolution. He finds that the approach outperforms the popular decision tree algorithm C4.5. There are numerous other examples of feature learning using SR and GP. In "Coevolutionary Genetic Programming" (CGP) [27] domain specific primitive features are combined with domain independent primitive operators to create composite features for use in object recognition. In "Simultaneous Generation of Prototypes and Features through Genetic Programming" (SGPFGP) [13] GP is used to create Prototypes [48] which reduce the dimensionality and number of training examples required for the Nearest Neighbors classification technique.

This research builds on the idea of creating a parametrized terminal (of which a Range Terminal is an instance) first developed by Kriegman et al. [25]. In his research Kriegman predicted the yearly total of regional snow water equivalent (SWE) using satellite derived daily snow and SWE data. The parametrized terminal was used to aggregate pixels in high mountain Asia into shapes and apply an aggregation function such as the mean to the resulting distribution. The terminals were then used as leaves in the SR expression tree as in this research.

Two studies stand out as being similar to ours in that they employ aggregates of temporal variables over time. Stanislawska et al. [43] predict global temperature change from 1900-1999 using (among other features) the mean over a randomly initialized range of historic temporal variables. However unlike in our work this range cannot evolve and does not explore other moments of the distribution or the minimum and maximum aggregation functions. Our technique is tangentially related to "Symbolic Aggregate approXimation - Evolutionary Feature Generation" (SAX-EFG) [22] in that it looks for signals over time. In SAX-EFG the authors present a time series classification technique in which a time series is discretized by breaking it into recurring subsequences called "motifs". These motifs are used as building blocks in the construction of more complex features representing different portions of the time series. The motifs and the generated features are then optimized using GP.

In this research we use Lasso regression at the end of the feature learning process. There are a variety of SR methods that incorporate linear ML methods directly into the process of building the SR models. Many of the methods exploit the speed at which the global optimum of a model linear in basis functions can be found. "Fast Function Extraction" (FFX) is a deterministic method that creates a huge number of basis functions which are combined and optimized using ElasticNet [56]. Ick et al. [20] modify FFX to use an evolutionary process and demonstrate the power of GP used in tandem with ElasticNet. The method is later applied to resting state fMRI data to understand nonlinear interactions of different regions in the brain [1, 19]. In "Multi-gene genetic programming" (MGGP) evolved trees are linearly combined before being optimized using least squares [14]. Other methods [2] are similar. In a method explicitly tasked to find features, "Evolutionary Feature Synthesis" (EFS), features rather than individuals are evolved and iteratively weighted and removed by combining them into a linear model and applying Lasso regression [3]. The authors find the method to be extremely fast and competitive with NN on benchmark problems. The resulting linear models are selected from a Pareto front based on error and complexity. Some of these hybrid methods are quickly closing in on the current state of the art in ML while being easily interpretable. For a review see [55].

SR has also been used to model numerous environmental systems including modeling SWE [9, 25], global temperature change [44], algae blooms [31], heat flux [43], hydrology [40, 52], vegetation cover in the context of soil erosion [33], riparian zones [28], and others. In fact Genetic Algorithms in general have been praised along with NN in their usefulness at providing insight into ecology [35]. Often these models are optimized not just for error but for interpretability. This is usually achieved via a multiobjective Pareto optimization scheme in which at least one of the objectives tries to minimize model complexity [42]. A simple approach is to add an objective that minimizes tree size although this is not always indicative of the semantic complexity of a mathematical expression. A more robust complexity measure is to estimate the best fit polynomial of an expression and take the degree as a measure of complexity [50]. We will employ complexity objectives in this research to ensure simple models are available on the Pareto front.

Chapter 2

Finding Temporal Features with Symbolic Regression

Finding Temporal Features with Symbolic Regression

Chris Fusting
University of Vermont
chris@chrisfusting.com

Marcin Szubert
University of Vermont

Josh Bongard
University of Vermont

Christian Skalka
University of Vermont

ABSTRACT

Building and discovering useful features when constructing machine learning models is the central task for the machine learning practitioner. Good features are useful not only in increasing the predictive power of a model but also in illuminating the underlying drivers of a target variable. In this research we propose a novel feature learning technique in which Symbolic regression is endowed with a "Range Terminal" that allows it to explore functions of the aggregate of variables over time. We test the Range Terminal on a synthetic data set and a real world data in which we predict seasonal greenness using satellite derived temperature and snow data over a portion of the Arctic. On the synthetic data set we find Symbolic regression with the Range Terminal outperforms standard Symbolic regression and Lasso regression. On the Arctic data set we find it outperforms standard Symbolic regression, fails to beat the Lasso regression, but finds useful features describing the interaction between Land Surface Temperature, Snow, and seasonal vegetative growth in the Arctic.

KEYWORDS

machine learning, artificial intelligence, spatial, temporal, genetic programming

ACM Reference format:
Chris Fusting, Marcin Szubert, Josh Bongard, and Christian Skalka. 2018. Finding Temporal Features with Symbolic Regression. In *Proceedings of GECCO, Kyoto Japan, July 2018 (GECCO 2018),* 10 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Many machine learning (ML) problems have periodic observations, each considered a variable, of a sensor over time that are predictive of the value of a target variable. In addition to the variables themselves having predictive power, sometimes a function of the aggregate of a range of variables over time also have predictive power. We encountered this situation when trying to predict primary productivity in the Arctic using satellite derived Land Surface Temperature (LST) and Snow data in the Arctic. To see why a function of aggregated variables may be predictive we provide some background: The 2015 Arctic Report Card [21] showed an unusual downward trend in the amount of greenness across the Arctic circle and the northern reaches of Eurasia. The cause of the recent browning trend is not known, although field studies suggest extreme temperature and other events have had a significant impact [32]. Bjerke et al. [5] found fourteen weather events leading to plant stress from October of 2011 to September 2012 in the Nordic Arctic Region. These include: severe cold spells when plants had little insulating snow to protect them, unusually warm temperatures during Winter months causing Spring like development and subsequent death of plants, flooding which destroyed plants and shrubs in low lying areas, and others. Bokhorst et al. [7] simulated warming events in mid-Winter using heating lamps and underground cables and found that these conditions caused snow melt, leaving plants exposed to extreme cold. This caused vegetation death and hence less primary productivity the following Summer. These events are in contrast to the overall trend which shows increased primary productivity in the Arctic.

To capture extreme events we needed to be able to search over a range of time during which they may occur. We hypothesized that functions such as the minimum, maximum, mean, sum, variance, and skew, when taken over time, would capture the occurrence of extreme events. As an an example take the previously described Winter warming event. This event could be described by the maximum of temperature variables during the Winter. In this paper we develop a novel feature learning technique that develops features representing the function of the aggregate of variables over time. To motivate our development choices we first discuss some background on how features are commonly constructed.

1.1 Feature Engineering and Selection

The process of manually creating features is known as "feature engineering" or "feature construction" and depending on the number of variables in a data set can be quite laborious. Feature engineering is often done in tandem with, but should not be confused with, feature selection in which the dimensionality of a data set is reduced. Often this is in an effort to reduce computational complexity or if the goal is to infer meaning from the model, identify highly predictive features. There are many approaches to feature engineering and selection, some of which we discuss here. For a good introduction see Guyon [17].

The process of feature engineering largely depends on the domain expertise of an analyst and the breadth of scientific knowledge available in the field. If little or nothing is known about the interaction of the predictor variables (predictors) with the target variable, features may be randomly constructed by taking the cross product of the predictors and (or) applying functions such as $\sqrt{x}, \exp(x), x^p$ to the predictors and then evaluating their performance in models.

GECCO 2018, Kyoto Japan

DOI: 10.1145/nnnnnnn.nnnnnnn

Selecting performant features can be achieved through scoring. A simple method is to compute the Pearson correlation coefficient between features and the target (which is performed implicitly in linear regression). However naive applications of this idea such as in Stepwise regression have been widely criticized [46] for being biased, unreliable, and badly implemented in software. More sophisticated approaches to variance decomposition have been show to be fruitful but can be computationally expensive [16].

If domain knowledge is available this process can be expedited by developing features using scientific insight. Building features in this manner, however, depends on the depth of scientific knowledge available in the model's domain and as such many useful features may not be discovered, or may be discovered only occasionally through random chance.

Once constructed the M features $\boldsymbol{\phi} = (\phi_0, \cdots, \phi_{M-1})^T$ called basis functions can be input into a linear basis function model:

$$y(\boldsymbol{w}, \boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{1}$$

where $w = (w_0, \cdots, w_{M-1})$ is a vector of weights and $\phi_0(\boldsymbol{x}) = 1$ is the bias term. Classification models are built similarly. From this starting point one can explore various linear methods such as Ridge, Lasso, ElasticNet, Bayesian, and other sophisticated regression and classification techniques[4, 12]. These techniques employ a regularization parameter such that large coefficients are penalized. In all cases this leads to a more general model but in the case of Lasso [47] regression some coefficients are in fact zeroed out which leads to a more parsimonious model and performs additional feature selection implicitly.

1.2 Feature Learning

An alternative to feature engineering is "feature learning", a process in which features are learned automatically using the available variables. There has been recognition in the last decade that the amount of data and number of potential predictors is increasing dramatically, necessitating the need and development of feature learning techniques [10].

Unsupervised methods such as Principle Component Analysis (PCA), kernel PCA [39] (for nonlinear interactions), Principle Component Regression (PCR), and others can be useful in both feature selection and engineering but the resulting features are difficult to interpret. Other unsupervised methods such as k-means clustering [18] and Latent Dirichlet Allocation [6] produce features that are more interpretable, although the later is usually used only for text mining.

Many machine learning techniques perform feature learning as a side effect. One powerful approach is to define the features implicitly using a kernel as in Gaussian Process regression (GPR) where a kernel is defined in terms of the dot product of the feature vectors

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x'})$$

The basis functions $\boldsymbol{\phi}$ can be constructed by the choice of the kernel. For example GPR is equivalent to Bayesian regression when the kernel is linear $k(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{x}^T \boldsymbol{x'}$. Another example is the polynomial kernel $k(\boldsymbol{x}, \boldsymbol{x'}) = (\boldsymbol{x}^T \boldsymbol{x'} + c)^n$ which expands out to an nth degree polynomial. This choice of kernel is equivalent to Polynomial regression (although the computation is not bound by the degree n). Other kernels such as the radial basis function (RBF) $\exp(-||\boldsymbol{x} - \boldsymbol{x'}||^2/2\sigma^2)$ are commonly used to express the idea that variables close to each other in value behave similarly. Expansion of the RBF leads to the dot product of feature vectors of infinite length, a remarkable ability of Gaussian Processes. Despite their power it is difficult to pinpoint exactly what features are playing the most important role beyond the expansion and analysis of the kernel. See Bishop [4] for a good introduction to GPR and Rasmussen[34] for a complete exposition.

Another approach that has gained popularity and quickly become the modern state of the art is Neural Networks (NN)[15]. Neural Networks were originally made to mimic the brain of biological systems although today much of their implementations differ from what we know of living brains. The formulation of a neural network does not include the idea of basis functions as features in the same way as in linear and kernel methods. Even a simple fully connected feed forward single layer classification network takes a form difficult to evaluate for feature meaning

$$y_k(\boldsymbol{x}, \boldsymbol{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i\right)\right)$$

where the superscript indicates the layer of the network, σ is the sigmoid function, h is a nonlinear activation function (such as a sigmoid, logistic, or tanh), D is the number of variables, and $x_0 = 1$ such that the bias parameter is absorbed into the weights. Research in the area of interpreting neural networks is growing but still nascent[26, 54].

Another machine learning technique with implicit feature learning is Symbolic regression (SR)[23]. Symbolic regression has been used extensively to discover mathematical expressions from data, including finding physical laws[37] and reverse engineering dynamical systems[8]. Rather than defining a model and fitting parameters to data, SR builds a mathematical expression using data as a guide. Although SR models are not always fit using Genetic Programming (GP)[29], it is the most common method and we will describe it here. Programmed Darwinian evolution guides the construction of a population of mathematical expressions made up of operators and functions such as $+, -, *, \div, \ln, \exp$, etc. These operators act on terminals which are usually the variables and other features. Expressions that perform well given training data and a fitness metric mate and pass on their characteristics to their children. The fitness metric usually takes the form of mean absolute or squared error although other metrics such as the Pearson Correlation have been used successfully[43]. After some number of generations the remaining expressions are selected as the winning models.

Because SR makes no assumptions about the form of the model a priori it is capable of capturing nonlinear interactions between predictors and the target without explicitly (in the case of linear regression) or implicitly (in the case of GP regression) specifying a basis $\boldsymbol{\phi}$. However in contrast to NN it is possible to explicitly optimize SR to keep the mathematical expressions it evolves simple which makes interpretation easier. Because of this SR is an ideal feature learning tool.

In this work we introduce a terminal for the SR expression tree, the Range Terminal, which calculates a function of an aggregate of variables over time. This terminal allows mathematical expressions

to capture signals over a range of temporal predictor variables. In addition it provides a simplifying symbol that makes the resulting mathematical expressions easily interpretable. We also propose a simple feature learning method:

(1) SR with the Range Terminal is used to evolve mathematical expressions.
(2) Expression trees are broken up into features.
(3) Features are selected and weighted using Lasso regression.

1.3 Related Research

Variable and feature selection using Symbolic Regression has been explored in many studies [30, 41, 45]. Perhaps the first example of SR as a feature learning tool is the automatically defined function (ADF) [23]. Krawiec explores a feature construction technique in which each individual in the population is made up of isolated features that are evolved separately [24]. In an extension to the method highly successful features are protected from further evolution. He finds that the approach outperforms the popular decision tree algorithm C4.5. There are numerous other examples of feature learning using SR and GP. In "Coevolutionary Genetic Programming" (CGP) [27] domain specific primitive features are combined with domain independent primitive operators to create composite features for use in object recognition. In "Simultaneous Generation of Prototypes and Features through Genetic Programming" (SGPFGP) [13] GP is used to create Prototypes [48] which reduce the dimensionality and number of training examples required for the Nearest Neighbors classification technique.

This research builds on the idea of creating a parametrized terminal (of which a Range Terminal is an instance) first developed by Kriegman et al. [25]. In his research Kriegman predicted the yearly total of regional snow water equivalent (SWE) using satellite derived daily snow and SWE data. The parametrized terminal was used to aggregate pixels in high mountain Asia into shapes and apply an aggregation function such as the mean to the resulting distribution. The terminals were then used as leaves in the SR expression tree as in this research.

Two studies stand out as being similar to ours in that they employ aggregates of temporal variables over time. Stanislawska et al. [43] predict global temperature change from 1900-1999 using (among other features) the mean over a randomly initialized range of historic temporal variables. However unlike in our work this range cannot evolve and does not explore other moments of the distribution or the minimum and maximum aggregation functions. Our technique is tangentially related to "Symbolic Aggregate approXimation - Evolutionary Feature Generation" (SAX-EFG) [22] in that it looks for signals over time. In SAX-EFG the authors present a time series classification technique in which a time series is discretized by breaking it into recurring subsequences called "motifs". These motifs are used as building blocks in the construction of more complex features representing different portions of the time series. The motifs and the generated features are then optimized using GP.

In this research we use Lasso regression at the end of the feature learning process. There are a variety of SR methods that incorporate linear ML methods directly into the process of building the SR models. Many of the methods exploit the speed at which the global optimum of a model linear in basis functions can be found. "Fast Function Extraction" (FFX) is a deterministic method that creates a huge number of basis functions which are combined and optimized using ElasticNet [56]. Ick et al. [20] modify FFX to use an evolutionary process and demonstrate the power of GP used in tandem with ElasticNet. The method is later applied to resting state fMRI data to understand nonlinear interactions of different regions in the brain [1, 19]. In "Multi-gene genetic programming" (MGGP) evolved trees are linearly combined before being optimized using least squares [14]. Other methods [2] are similar. In a method explicitly tasked to find features, "Evolutionary Feature Synthesis" (EFS), features rather than individuals are evolved and iteratively weighted and removed by combining them into a linear model and applying Lasso regression [3]. The authors find the method to be extremely fast and competitive with NN on benchmark problems. The resulting linear models are selected from a Pareto front based on error and complexity. Some of these hybrid methods are quickly closing in on the current state of the art in ML while being easily interpretable. For a review see [55].

SR has also been used to model numerous environmental systems including modeling SWE [9, 25], global temperature change [44], algae blooms [31], heat flux [43], hydrology [40, 52], vegetation cover in the context of soil erosion [33], riparian zones [28], and others. In fact Genetic Algorithms in general have been praised along with NN in their usefulness at providing insight into ecology [35]. Often these models are optimized not just for error but for interpretability. This is usually achieved via a multiobjective Pareto optimization scheme in which at least one of the objectives tries to minimize model complexity [42]. A simple approach is to add an objective that minimizes tree size although this is not always indicative of the semantic complexity of a mathematical expression. A more robust complexity measure is to estimate the best fit polynomial of an expression and take the degree as a measure of complexity [50]. We will employ complexity objectives in this research to ensure simple models are available on the Pareto front.

1.4 Paper Structure

This paper is organized as follows: First we develop the Range Terminal. Then we show that SR with the Range Terminal outperforms traditional SR and Lasso regression on a nonlinear synthetic data set and explore the learned features for meaning. Finally we show the technique outperforms traditional SR (although not Lasso regression) on a high resolution satellite data set and offers insights not found in a competing Lasso regression model. Code for this work can be found at: https://github.com/cfusting/arctic-browning.

2 THE RANGE TERMINAL

As discussed previously field research has shown that anomalous events may cause browning events. An example of this is a sudden rise in temperature during the winter inducing a snow melt. The shrubs and small trees previously insulated by the snow are exposed and when the cold returns will die. This leads to less overall vegetative growth during the coming summer months. Capturing an event of this type is particularly challenging as it can happen at any time during a window over which multiple temporal variables span. This is a case where we are interested not in the explanatory power of the predictors but a function of their aggregate over a

range in time. In this particular example what we are interested in is the maximum temperature over some range of temporal variables given expectations formed by previous research.

To facilitate explicit construction of this type of feature we endowed SR with a Range Terminal. This terminal is a leaf on a tree representing a mathematical expression (see Koza[23] for details) and uses a variable type, a range of time and an aggregate function. The resulting value is produced by applying the aggregate function to the variable type during the specified range of time. We will refer to this idea as follows

$$\psi(f, a, b) \tag{2}$$

where f is the aggregate function, and a, b the start and end of the variable range in question (initialization and evolution will be discussed in section 3.2). In the previous example the variable type would be LST, the range of time Winter, and the aggregate function the maximum expressed as $\psi(\max, \mathbf{lst}_{129}, \mathbf{lst}_1)$ where the subscript denotes the day of the year.

It is important to note that SR without the Range Terminal might (given the time and quite a bit of luck) be able to build the equivalent (or nearly) mathematical expression. The previous example can be written as a softmax function

$$\ln\left\{e^{\mathbf{lst}_i} + e^{\mathbf{lst}_{i+1}} + \ldots + e^{\mathbf{lst}_n}\right\} \tag{3}$$

$$|\mathbf{lst}_i| > 1 \qquad |\mathbf{lst}_i - \mathbf{lst}_j| > 1 \qquad i, j \in T \qquad i \neq j$$

where $\mathbf{lst}_i \cdots \mathbf{lst}_{t+n}$ are LST variables over the winter and $|x|$ is the absolute value. In fact there an infinite number of ways we could describe the maximum of a set of variables with elementary mathematical operators. The purpose of the Range terminal is to make this explicit and the parameters that govern the idea evolvable. This both constrains the search space to meet the assumptions we have about our data set and makes the result easy to interpret.

3 SYNTHETIC DATA

We tested the Range Terminal on a synthetic data set to examine its efficacy under controlled conditions. To do so 60 variables of 1200 observations were generated. Each variable x_i was built as follows:

$$x_i \sim \mathcal{N}(a_i, b_i)$$

$$a_i \sim \mathcal{U}(10, 50)$$

$$b_i \sim \mathcal{U}(0, 10)$$

where $\mathcal{N}(\mu, \sigma)$ is a normal distribution with mean μ and standard deviation σ and $\mathcal{U}(r, w)$ is a uniform distribution. We then generated a target variable y using the following equation:

$$\boldsymbol{y} = \sum_{j=0}^{19} \boldsymbol{x}_j + \max(\boldsymbol{x}_{12}, \boldsymbol{x}_{13}, \cdots, \boldsymbol{x}_{24}) + \boldsymbol{x}_{52} + \frac{1}{10} \sum_{j=49}^{58} \boldsymbol{x}_j + \boldsymbol{x}_{32} \tag{4}$$

where j is the index of the variable. Note that the only part of this equation that might be tricky for SR without the Range Terminal to generate is the maximum (which could be generated using the softmax function in equation 3). We created the equation with this in mind to see if the Range Terminal was able to constrain the search space as we hypothesized.

Parameter	Value
Individuals	500
Generations	2000
Tourn. Size	2
Crossover Prob	0.9
Mutation Prob	0.1
Init. Min Height	1
Init. Max Height	6
Max Height	17
Max Size	200
Internal Node Sel. Bias	0.9

Table 1: Control and RT experiment common configurations.

3.1 Standardization

Many machine learning algorithms perform better when data is centered around zero and has a standard deviation of one. Regularized regression algorithms such as Lasso and Ridge in fact depend on this quality of the data to constrain the coefficients of the model fairly [12]. Standardization can however have unintended consequences.

When we built the Range Terminal we assumed that each variable has roughly equal importance and thus each variable x_i is assigned coefficient $a_i \approx a_j, i \neq j$. However standardizing the data causes each variable to be divided by its respective standard deviation (note a_j is relevant only to a variable's importance in the context of a model and is not involved in standardization):

$$a_i x_i \xrightarrow{\text{standardize}} \frac{a_i}{\sigma_{x_i}}(x_i - \bar{x_i}) \tag{5}$$

where we see the coefficient now depends on σ_{x_i}. Thus unless each variable is drawn from the same distribution, the size of the coefficients relative to each other are not stable, breaking our assumption that they are roughly the same. Because of this we require that variables are not standardized prior to running SR with Range Terminals.

3.2 Experiment Setup

We conducted two experiments using SR, each with 40 runs (enough samples to give reasonable power to statistical tests) where 1000 data points were used as training data and 200 left for testing. One experiment, the control, was run without Range Terminals and the other experiment, the RT experiment, was run with Range Terminals. Aside from the addition of the Range Terminal to the choice of terminals and the associated mutation techniques, all other settings were the same in both experiments. The operators available to the trees were: $+, -, *, \ln, \exp, x^2, x^3$ and constant values sampled from $\mathcal{N}(0, 10)$. In the RT experiment the functions min max, sum, mean, variance, and skew were available to the Range Terminals. Range Terminal initialization was done by selecting an aggregate function, a variable type (in this case there is only one), and a begin and end range, all with uniform probability. See table 1 for additional running parameter configurations for the control and RT experiment.

We evolved a population of randomly initialized individuals with age, and fitness (mean squared error) Pareto optimization [38]. Expression trees were randomly initialized by selecting operators and terminals with uniform probability and adhering to initialization minimum and maximum height limits as described in table 1. In addition we add to the Pareto optimization two more objectives: size, where the goal is to minimize the size of the tree and complexity, where the goal is to minimize the degree of the best fitting polynomial approximation of the tree semantics (for details on the complexity measure see [50]). The purpose of these objectives is to balance our desire to minimize error with our need to build an interpretable equation. At each generation we generate a set of additional individuals the size of the population, most with crossover, some with mutation, and one randomly as during initialization. These new individuals were are added to the population and tournament selection was performed to bring the population down to its original size. Crossover was single point biased as described in Koza[23]. For the control experiment mutation was single point biased with tree generation as in initialization. For the RT experiment mutation was single point biased with 0.5 probability and also with 0.5 probability was Range Terminal mutation in which mutation was performed as described in algorithm 1. Note that the

Algorithm 1: Mutate Range Terminal

Data: Tree representing an individual
Result: Tree with a mutated Range Terminal
R = Range Terminals in tree.
θ = Select a node uniformly from R.
P = Select the high or low range parameter of node with uniform probability.
ϕ = high - low | 1
x = Take one sample from $\mathcal{N}(0, \sqrt{\phi})$.
x' = ceiling($|x|$)
if $x < 0$ **then**
  $x' = x' * -1$
end
$P = P + x$
Ensure P does not extend past the other range parameter.

aggregate function does not change as this may introduce a large and potentially detrimental mutation.

3.3 Experiment Results

To assess the predictive power of the control and RT experiments we took the individual with the lowest validation error from the Pareto front of each of the 40 runs. This left us with a sample of 40 individuals for both the control and RT experiment. The resultant 80 errors were scaled between 0 and 1 according to $\frac{x}{\max x}$ to make the test errors easier to interpret. See table 2 for statistics . We compared the two samples using the Wilcoxon rank sum test [53] because the samples were not normally distributed. The RT experiment was significantly different than the control with $W = 1600$ and p-value < 0.001 where the difference in location is 0.294 and falls within with a 99% confidence interval of $(0.236, 0.409)$. Thus we find the RT experiment performs significantly better than the control on the synthetic data.

	Min	Max	Median
Control	0.094	1.000	0.303
RT experiment	0.000	0.088	0.007

Table 2: Synthetic Data test error statistics for the 40 best validation score individuals from the Control and RT Experiments.

We also built a linear baseline model using Lasso regression. In Lasso regression[47] the error of the simple linear model from equation 1 is defined as

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|$$

where t_n is the nth training value of the target variable and λ is a real valued regularization parameter. Lasso is an appealing model to use because it drives the coefficients to zero for λ sufficiently large. This creates a more parsimonious model that is easier to interpret and will be especially useful later when we construct the basis functions $\boldsymbol{\phi}$ using the features found in the RT experiment.

To build the Lasso model we used the same 1000 observations and 60 variables as in the SR runs. Prior to regression data was standardized by subtracting the mean and dividing the result by the standard deviation for each variable. Lasso was performed with 10-fold cross validation. We calculated the test error using the remaining 200 observations. The resulting normalized value, 0.034, was clearly lower than the control and we thus find the linear baseline is superior in predictive ability of standard SR.

To test the performance of the RT experiment compared to the linear baseline we ran a one-sided Wilcoxon signed-rank test. The RT experiment error was significantly less than the linear baseline error with $V = 80$, p-value < 0.001 where the median fell within the 99% confidence interval $(-Inf, 0.013)$[1]. Therefore the RT experiment's predictive ability is superior to the linear baseline.

Also of significant interest are the predictive features (notably Range Terminals) found by the RT experiment. To explore these features we took the Pareto fronts from each of the 40 runs of the RT experiment and extracted the features from the individuals on the front. Features are extracted by first simplifying the trees representing the individuals via standard mathematical rules and then splitting the equations wherever we found a plus (+) symbol. We found 294 unique features. The top 10 ordered by frequency can be found in table 3.

3.4 Discussion

Consider again equation 4 and note the appearance of x_{32} and various sums similar to $\sum_{i=0}^{19} x_i$. The prolific number of sums in the top ten is probably due to the magnitude of explanatory variance it predicts while x_{32} may appear frequently simply because it is easy to randomly initialize. To better understand the importance of features we used them as basis functions $\boldsymbol{\phi}$ as described in equation

[1]Because there are more than ten samples in this test a normal approximation is used and thus -Inf is included in the interval.

Feature
Constant Value
$\mathbf{x}_{32}$
$\psi(\text{sum}, \mathbf{x}_0, \mathbf{x}_{20})$
$\psi(\text{sum}, \mathbf{x}_0, \mathbf{x}_{21})$
$\mathbf{x}_{21}$
$\psi(\text{skew}, \mathbf{x}_{13}, \mathbf{x}_{24})$
$\psi(\text{sum}, \mathbf{x}_0, \mathbf{x}_{22})$
$\psi(\text{skew}, \mathbf{x}_{19}, \mathbf{x}_{22})$
$\psi(\text{min}, \mathbf{x}_{32}, \mathbf{x}_{33})$
$\mathbf{x}_{56}$

Table 3: Top ten most frequent features of all runs of RT experiment on the synthetic data.

Coefficient	Feature
0.840	$\psi(\text{sum}, \mathbf{x}_0, \mathbf{x}_{19})$
0.294	$\mathbf{x}_{32}$
0.190	$\psi(\text{max}, \mathbf{x}_{12}, \mathbf{x}_{24})$
0.081	$\mathbf{x}_{52}$
0.038	$\psi(\text{mean}, \mathbf{x}_{49}, \mathbf{x}_{58})$
0.029	$\ln(\psi(\text{mean}, \mathbf{x}_{32}, \mathbf{x}_{33}) + \psi(\text{sum}, \mathbf{x}_0, \mathbf{x}_{19}) + \mathbf{x}_{32})$
0.013	$\ln(\psi(\text{mean}, \mathbf{x}_{50}, \mathbf{x}_{58}))^2$

Table 4: Synthetic data features order by magnitude. Features were derived from the RT experiment and used as basis functions in a Lasso model.

1 and put them into a Lasso regression model. All 294 unique features were used to predict the target variable y with the same standardization, cross validation and testing scheme used to develop the linear baseline.

The normalized test error for the resulting Lasso model is 0.000 and is thus competitive with the linear baseline. This does not imply the model is significantly better than the linear baseline as it used the Pareto fronts from every run to derive features. To indicate statistical significance we would need to run an additional 30 to 40 experiments in this fashion and show a model of this type outperforms the linear baseline on average.

Low test error is however indicative of a good model and features that have strong predictive power. The features with coefficients greater than 0.01 appear in table 4. Remarkably, the Lasso model assigns the five greatest coefficients to exactly the five features used to construct the target variable in equation 4. Although we can make no claim that our method will always recover the features from this data set without further experimentation, it is certainly an encouraging result and will be the topic of future research. Note however that even with further experimentation we can only show that the method works for data sets in which we know the answer a priori. Practically speaking a machine learning practitioner has nothing to lose by extracting features from the individuals on all available Pareto fronts and inputting them into Lasso or other regularized linear model.

4 ARCTIC BROWNING DATA

The development of the Range Terminal was motivated by the desire to capture events over a range of temporal variables to better understand how Land Surface Temperature (LST) and Snow impact the MTI-NDVI in the Arctic as described in equation 7. We chose Snow and LST for two reasons: First, field research suggested that temperature and snow cover were both important factors in determining plant health. Second, LST and Snow are both available as high quality data products in the Arctic which is not the case for all remotely sensed products (precipitation for example). To investigate this question we studied an area in Northwestern Russia, Moderate Resolution Imaging Spectroradiometer (MODIS) tile h20v02 (longitude minimum: 40°, longitude maximum: 87.7385°, latitude minimum: 60°, latitude maximum: 70°) from 2002 to 2016. We chose the area because about half of it is in the Arctic circle and since the overall climate has been warming we felt trends observed now in the Arctic may have been observed a bit further south a decade ago. To predict gross primary productivity we used The Normalized Vegetation Index is a measure of greenness derived from the red and near-infrared (NIR) bands of a multispectral camera according to

$$v = \frac{NIR - RED}{NIR + RED}$$

Multispectral cameras have been carried on satellites since the launch of Landsat 1 by NASA in 1972 and the derived NDVI is commonly used to assess vegetation health over large areas of the earth. To measure the gross primary productivity of a season the Time Integrated Normalized Vegetation Index (TI-NDVI) is commonly used [49]. It is defined as

$$\int_T v(t)dt \tag{6}$$

where T is the growing season and $t \in T$. In this research we will use a discrete representation of the TI-NDVI, Mean Time Integrated NDVI (MTI-NDVI) as defined by

$$\frac{1}{|T|} \sum_{t \in T} v(t)dt \tag{7}$$

where T is a discrete set of time steps and $|T|$ is the number of elements in T. Note the previous is proportional to equation 6 and thus for our purposes is an equivalent form of measurement while being robust to missing values.

The Land Surface Temperature[51], Snow[36], and NDVI[11] data were derived from the MODIS sensor aboard the Terra satellite. The Land Surface Temperature (MOD11A2) and NDVI (MOD13A3) data products were retrieved from the online Data Pool, courtesy of the NASA Land Processes Distributed Active Archive Center (LP DAAC), USGS/Earth Resources Observation and Science (EROS) Center, Sioux Falls, South Dakota[2]. The Snow (MOD10A2) data was retrieved from courtesy of the National Snow and Ice Data Center[3].

4.1 Data Preprocessing

We did the minimal amount of preprocessing necessary to ensure data was of high quality and not altered by interpolation. We downloaded LST and NDVI data at a one kilometer resolution because we

[2]https://lpdaac.usgs.gov/data_access/data_pool
[3]https://n5eil01u.ecs.nsidc.org/MOST/MOD10A2.006/

felt it granular enough to capture small variations in temperature and coarse enough to allow for relatively computationally efficient modeling. For the LST data, daytime observations were used. The snow data was only available at a 500 meter resolution and was upsampled to one kilometer. Each data set came with detailed quality control information which we used to remove pixels of questionable quality. The quality control information was available per pixel and we applied the rules as described in algorithms 2, 3, 4 to decide if a pixel would remain in the data set. The algorithms describe the bitwise operations applied per pixel that build up a binary mask specifying valid and invalid data.

Algorithm 2: Create NDVI mask. Note "&" and "|" are bitwise operators.

Data: 16 bit quality control pixel.
Result: Mask were 1 = valid data and 0 = invalid data.
mask = bits 0,1 $\leq$ 1 & bit 8 == 0
mask = mask & bit 10 == 0
mask = mask & bit 15 == 0
mask = mask & bits 2,5 $\leq$ 11
dummy = bits 6,7 == 1 — bits 6,7 == 2
mask = mask & dummy

Algorithm 3: Create LST mask. Note "&" and "|" are bitwise operators.

Data: 8 bit quality control pixel.
Result: Mask were 1 = valid data and 0 = invalid data.
mask = bits 0,1 $\leq$ 1 & bits 2,3 $\leq$ 1
mask = mask & bits 4,5 $\leq$ 2
mask = mask & bits 6,7 $\leq$ 2

Algorithm 4: Create snow mask.

Data: Snow data.
Result: Mask were 1 = valid data and 0 = invalid data.
mask = Matrix of 0's
mask = 1 where pixel is 25 or 200

The temporal resolution of the NDVI data was upsampled from monthly to yearly according to equation 7. The included months span from day 152 of the year to day 245 of the year, effectively capturing June through September. After this first stage of preprocessing we were left with one kilometer pixels with an observation every eight days for LST and Snow and a yearly observation for NDVI. Note data was not reprojected from its native Sinusoidal projection with the understanding that pixels further north are slightly smaller than pixels further South. We did not feel the change in size to be significant enough to merit a weighting scheme.

To model the effect of LST and Snow on MTI-NDVI we used the LST and Snow 8 day observations for a year leading up to the calculation of MTI-NDVI as defined in equation 7. Variables are indexed by the day of year on which they begin an 8 day sample.

	Min	Max	Median
Control	0.575	1.000	0.742
RT experiment	0.604	0.819	0.686

Table 6: Arctic Data test error statistics for the 40 best validation score individuals from the Control and RT Experiments.

We naively assumed each area in space behaves the same as all the others and therefore used every pixel in a given year as an observation. However because many pixels were lost during quality control it was necessary to remove some variables in which many of the observations were missing to prevent further data loss. We removed any variable that had a proportion of missing values $\geq$ 0.15. In addition we removed any snow variable with mean snow cover $\geq$ 0.98. There were two reasons for this: First, the snow data requires daytime observations and in the high latitudes the Arctic night creeps over much of our sample area creating a blanket of missing pixels. If the area south of the Arctic night is almost completely snow covered it is reasonable to assume the area North is as well. Second, a binary snow cover variable is not so useful if everything is covered in snow; rather it is during the Spring, Summer, and Fall that this variable shows meaningful variance.

After removing variables violating the constraints we were left with 43 (out of a possible 90) predictors and 4103815 (out of a possible 21600000 about 19%) observations. Of the LST variables (whose index is by the day of the year when the eight day sample starts) those indexed by 241, 161, 57, 1, 361, 321, 313, 305, 297, 289, 281, 273, and 265 were thrown out. Of the Snow variables those indexed by 209, 201, 193, 185, 177, 169, 161, 129, 121, 105 were kept. Note that variables with an index less than 255 correspond to the Winter, Spring, and Summer leading up to the calculation of MTI-NDVI while variables greater than 255 correspond to the previous year's Fall. See table 5 for a summary of the data.

4.2 Experiment Setup

We setup a control and RT experiment nearly as we did for the synthetic data set with the following differences: The Arctic data set was divided into training and test data by using the years 2002−2013 (3412751 observations, about 83%) for training and 2014 − 2016 (691064 observations, about 17%) for testing. Because of the size of the data set we trained the SR models using a random sample without replacement of 10% (341275 observations) of the training data which refreshed every ten generations.

4.3 Experiment Results

As was the case in the evaluation of the Synthetic data set results, we took the individual with the lowest validation error from the Pareto front of each of the 40 runs. This left us with a sample of 40 individuals for both the control and RT experiment. The resultant 80 errors were scaled between 0 and 1 to make the test errors easier to interpret. See table 6 for data statistics. We compared the two samples using the Wilcoxon rank sum test and found the RT experiment to be significantly better than the control with $W = 1142$, p-value < 0.001. The difference in location was 0.064 falling within the 99% confidence interval $(0.022, 0.106)$.

Data	Spatial Resolution	Temporal Resolution	Value Range	Unit with Multiplier	Multiplier
LST	1 km	8 day	7500 - 65535	Kelvin	0.02
Snow	1 km	8 day	0 or 1	Binary	NA
NDVI	1 km	Yearly	-2000 to 10000	Greenness	0.0001

Table 5: Arctic data after upsampling.

We built a linear baseline model in the same manner as in the synthetic data using the data from 2002 to 2013 as training and data from 2014 to 2016 as testing. The normalized test error was 0.656, clearly outperforming both the control and the RT experiment. Unlike the synthetic data set we do not a priori know the underlying mechanisms governing the system from which the target variable was derived. Thus it is difficult to say why the linear model is superior. It may be that the relationship between LST, Snow and MTI-NDVI is mostly linear and thus a method able to achieve a global optimum will be superior.

We explored the features found by the RT experiment in the same way as with the synthetic data set. The Pareto fronts of each of the 40 RT experiment runs were extracted for features from the individuals on the fronts. We found 493 unique features and as before used them as predictors of MTI-NDVI in a Lasso regression model with the same setup as in the synthetic data.

The Lasso mode model built with the features found from the RT experiment is competitive with the linear model with a slightly lower normalized test error of 0.646. As discussed previously we cannot claim this method will produce a model with similar test error results reliably. We can however use the knowledge that this model performs very well to substantiate our claim that the features within it are useful and worth analyzing further. Table 7 compares features from the RT experiment with features from the linear baseline. The features are ordered by coefficient magnitude.

4.4 Discussion

Interestingly, the first three features of the baseline Lasso model stagger backwards from Spring through Winter indicating that above average temperature during each of these 8 day periods is indicative of more vegetative growth[4]. However by examining the first two features of the RT experiment Lasso model we find not that three 8 day periods are important, but that the average value over a period of winter and a single 8 day period are important. This difference is subtle but paints a much fuller picture of how MTI-NDVI is actually reacting. We observe the same pattern when considering features four, five and six of the baseline and three and four of the RT experiment[5].

Feature five of the RT experiment Lasso model is the first example of the Range Terminal capturing a nonlinear effect not captured (or easily interpreted) by the linear model. It is clear that the minimum temperature plays an important role from late Winter through mid Summer, although with standardization and the other operations in the feature it is difficult to infer whether this will cause an increase or decrease in MTI-NDVI. Features seven and eight are also nonlinear, more clearly indicating that a high temperature event in Winter through Spring causes MTI-NDVI to increase while the opposite is true for a high temperature event in the Summer.

One of the most interesting features is number ten in the RT experiment Lasso model. We can see that when most of the mass of the distribution is centered over the Spring and early Summer (positive skew) MTI-NDVI suffers. This also implies that the Winter had to be quite cold and that perhaps going from a very cold winter to a hot Summer is shocking to the vegetation.

Although there are many interesting signals in the RT experiment Lasso model not found in the baseline Lasso model, the performance of the models clearly indicate that this system is mostly linear. One approach to uncovering the nonlinear effects would be to model anomalies separately. This in effect allows the model to focus on the tails of the distribution which are probably where the events we are looking for lie. This model can be used in an ensemble with a model that captures linear interactions between the predictors and the target. We leave this for future research.

5 CONCLUSIONS

In this paper we introduced the Range Terminal as a method to find features represented by by a function of the aggregate of variables over time. We showed significant improvement in the performance of SR with Range Terminals compared to SR without Range Terminals on both a synthetic and real world data set. Additionally we explored a feature recovery and analysis method using Lasso regression competitive with linear methods in predictive ability. This method allowed us to identify features with predictive ability and gain a richer understanding of the underlying mechanisms driving the target variable.

Although difficult to prove mathematically, it is likely that Range Terminals constrain the search space available to Genetic Programming and therefore enhances the speed at which it is able to converge upon a useful solution. Feature extraction is also aided by the Range Terminal in that it neatly packs large amounts of information into a simple function. Future research should explore variables varying over multiple dimensions, notably space-time, to unlock the true potential of the Range terminal in a spatiotemporal data set.

[4]Recall that in Lasso models data has been standardized and therefore a feature with a positive coefficient indicates above average values raise the value of the target variable, in this case MTI-NDVI.

[5]Note that in feature two of the RT experiment the model is trying to undo a log with the exponent 4 and in feature 3 it is trying to undo subtracting 12878.893 with the exponent 12 and then undo an exponent 12 with a log. The important signal in these features is expressed by the Range Terminal.

REFERENCES

[1] Nicholas Allgaier, Tobias Banaschewski, Gareth Barker, Arun LW Bokde, Josh C Bongard, Uli Bromberg, Christian Büchel, Anna Cattrell, Patricia J Conrod, Christopher M Danforth, and others. 2015. Nonlinear functional mapping of the human brain. *arXiv preprint arXiv:1510.03765* (2015).

[2] Ignacio Arnaldo, Krzysztof Krawiec, and Una-May O'Reilly. 2014. Multiple regression genetic programming. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, 879–886.

	Experiment Model		Control Model	
Number	**Coefficient**	**Feature**	**Coefficient**	**Feature**
1	0.131	$\psi(\text{mean}, \mathbf{lst}_{73}, \mathbf{lst}_{33})$	0.193	$\mathbf{lst}_{121}$
2	0.114	$\ln(\mathbf{lst}_{121})^4$	0.150	$\mathbf{lst}_{89}$
3	0.108	$\ln((\psi(\text{mean}, \mathbf{lst}_{97}, \mathbf{lst}_{65}) - 12878.893)^{12})$	0.118	$\mathbf{lst}_{33}$
4	0.105	$\psi(\text{mean}, \mathbf{lst}_{105}, \mathbf{lst}_{33})$	0.114	$\mathbf{lst}_{105}$
5	0.067	$\ln(\psi(\text{min}, \mathbf{lst}_{201}, \mathbf{lst}_{97}) + (\psi(\text{min}, \mathbf{lst}_{201}, \mathbf{lst}_{81}) - 12462.507)^2)^2$	0.081	$\mathbf{lst}_{81}$
6	0.063	$\psi(\text{sum}, \mathbf{snow}_{169}, \mathbf{snow}_{161}) * (\mathbf{lst}_{169} - \mathbf{lst}_{193})$	0.080	$\mathbf{lst}_{73}$
7	0.062	$\psi(\text{max}, \mathbf{lst}_{121}, \mathbf{lst}_{113})$	0.075	$\mathbf{lst}_{137}$
8	-0.053	$\psi(\text{max}, \mathbf{lst}_{193}, \mathbf{lst}_{177})$	-0.074	$\mathbf{snow}_{161}$
9	0.044	$\ln(\mathbf{lst}_{137} + (\psi(\text{mean}, \mathbf{lst}_{145}, \mathbf{lst}_{89}) - 12802.222)^3)$	0.068	$\mathbf{lst}_{57}$
10	-0.042	$\psi(\text{skew}, \mathbf{lst}_{169}, \mathbf{lst}_{33})^3$	0.066	$\mathbf{snow}_{121}$

Table 7: Arctic data features ordered by magnitude. Features were derived from the RT experiment and control and used as basis functions in a Lasso model.

[3] Ignacio Arnaldo, Una-May O'Reilly, and Kalyan Veeramachaneni. 2015. Building predictive models via feature synthesis. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM, 983–990.

[4] Christopher M Bishop. 2006. *Pattern recognition and machine learning*. springer.

[5] Jarle W Bjerke, Stein Rune Karlsen, Kjell Arild Høgda, Eirik Malnes, Jane U Jepsen, Sarah Lovibond, Dagrun Vikhamar-Schuler, and Hans Tømmervik. 2014. Record-low primary productivity and high plant damage in the Nordic Arctic Region in 2012 caused by multiple weather events and pest outbreaks. *Environmental Research Letters* 9, 8 (2014), 084006.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[7] Stef F Bokhorst, Jarle W Bjerke, Hans Tømmervik, Terry V Callaghan, and Gareth K Phoenix. 2009. Winter warming events damage sub-Arctic vegetation: consistent evidence from an experimental manipulation and a natural event. *Journal of Ecology* 97, 6 (2009), 1408–1415.

[8] Josh Bongard and Hod Lipson. 2007. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* 104, 24 (2007), 9943–9948.

[9] David Buckingham. 2014. *Inductive learning of snowpack distribution models for improved estimation of areal snow water equivalent*. Ph.D. Dissertation. University of Vermont.

[10] Ishanu Chattopadhyay and Hod Lipson. 2014. Data smashing: Uncovering lurking order in data. *Journal of the Royal Society Interface* 11, 101 (2014), 20140826.

[11] Kamel Didan. 2017. MOD13A3:MODIS/Terra Vegetation Indices Monthly L3 Global 1km Grid SIN V006. (2017). DOI: http://dx.doi.org/10.5067/MODIS/MOD13A3.006 Years: 2002 to 2016 Tile: h20v02 Accessed: 06/01/2017.

[12] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin.

[13] Mauricio Garcia-Limon, Hugo Jair Escalante, Eduardo Morales, and Alicia Morales-Reyes. 2014. Simultaneous generation of prototypes and features through genetic programming. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, 517–524.

[14] Akhil Garg, Ankit Garg, and K Tai. 2014. A multi-gene genetic programming model for estimating stress-dependent soil water retention curves. *Computational Geosciences* 18, 1 (2014), 45.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[16] Ulrike Grömping. 2007. Estimators of relative importance in linear regression based on variance decomposition. *The American Statistician* 61, 2 (2007), 139–147.

[17] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of machine learning research* 3, Mar (2003), 1157–1182.

[18] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1 (1979), 100–108.

[19] Ilknur Icke, Nicholas A Allgaier, Christopher M Danforth, Robert A Whelan, Hugh P Garavan, and Joshua C Bongard. 2014. A deterministic and symbolic regression hybrid applied to resting-state fmri data. In *Genetic Programming Theory and Practice XI*. Springer, 155–173.

[20] Ilknur Icke and Joshua C Bongard. 2013. Improving genetic programming based symbolic regression using deterministic machine learning. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 1763–1770.

[21] MO Jeffries, J Richter-Menge, JE Overland, and others. 2015. Arctic Report Card 2015. (2015).

[22] Uday Kamath, Jessica Lin, and Kenneth De Jong. 2014. SAX-EFG: an evolutionary feature generation framework for time series classification. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, 533–540.

[23] John R Koza. 1992. *Genetic programming: on the programming of computers by means of natural selection*. Vol. 1. MIT press.

[24] Krzysztof Krawiec. 2002. Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines* 3, 4 (2002), 329–343.

[25] Sam Kriegman, Marcin Szubert, Josh C Bongard, and Christian Skalka. 2016. Evolving Spatially Aggregated Features From Satellite Imagery for Regional Modeling. In *Parallel Problem Solving from Nature - PPSN XIV*. Springer.

[26] Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding Neural Networks through Representation Erasure. *arXiv preprint arXiv:1612.08220* (2016).

[27] Yingqiang Lin and Bir Bhanu. 2005. Evolutionary feature synthesis for object recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 35, 2 (2005), 156–171.

[28] Ammarin Makkeasorn, Ni-Bin Chang, and Jiahong Li. 2009. Seasonal change detection of riparian zones with remote sensing images and genetic programming in a semi-arid watershed. *Journal of Environmental Management* 90, 2 (2009), 1069–1080.

[29] Trent McConaghy. 2011. FFX: Fast, scalable, deterministic symbolic regression technology. In *Genetic Programming Theory and Practice IX*. Springer, 235–260.

[30] Randall K McRee. 2010. Symbolic regression using nearest neighbor indexing. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*. ACM, 1983–1990.

[31] Nitin Muttil and Kwok-Wing Chau. 2006. Neural network and genetic programming for modelling coastal algal blooms. *International Journal of Environment and Pollution* 28, 3-4 (2006), 223–238.

[32] Gareth K Phoenix and Jarle W Bjerke. 2016. Arctic browning: extreme events and trends reversing arctic greening. *Global change biology* (2016).

[33] Cesar Puente, Gustavo Olague, Stephen V Smith, Stephen H Bullock, Alejandro Hinojosa-Corona, and Miguel A González-Botello. 2011. A genetic programming approach to estimate vegetation cover in the context of soil erosion assessment. *Photogrammetric Engineering & Remote Sensing* 77, 4 (2011), 363–376.

[34] Carl Edward Rasmussen and Christopher KI Williams. 2006. *Gaussian processes for machine learning*. Vol. 1. MIT press Cambridge.

[35] Friedrich Recknagel. 2001. Applications of machine learning to ecological modelling. *Ecological Modelling* 146, 1 (2001), 303–310.

[36] Miguel Román, Dorothy Hall, and George Riggs. 2017. MODIS/Terra Snow Cover 8-Day L3 Global 500m Grid, Version 6. (2017). DOI: http://dx.doi.org/10.5067/MODIS/MOD11A2.006 Years: 2002 to 2016 Tile: h20v02 Accessed: 06/01/2017.

[37] Michael Schmidt and Hod Lipson. 2009. Distilling free-form natural laws from experimental data. *science* 324, 5923 (2009), 81–85.

[38] Michael Schmidt and Hod Lipson. 2011. Age-fitness pareto optimization. In *Genetic Programming Theory and Practice VIII*. Springer, 129–146.

[39] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. 1997. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*. Springer, 583–588.

[40] C Sivapragasam, R Maheswaran, and Veena Venkatesh. 2008. Genetic programming approach for flood routing in natural channels. *Hydrological processes* 22, 5 (2008), 623–628.

[41] Guido Smits, Arthur Kordon, Katherine Vladislavleva, Elsa Jordaan, and Mark Kotanchek. 2006. Variable selection in industrial datasets using pareto genetic

programming. *GENETIC PROGRAMMING SERIES* 9 (2006), 79.
[42] Guido F Smits and Mark Kotanchek. 2005. Pareto-front exploitation in symbolic regression. In *Genetic programming theory and practice II*. Springer, 283–299.
[43] Karolina Stanislawska, Krzysztof Krawiec, and Zbigniew W Kundzewicz. 2012. Modeling global temperature changes with genetic programming. *Computers & Mathematics with Applications* 64, 12 (2012), 3717–3728.
[44] Karolina Stanislawska, Krzysztof Krawiec, and Timo Vihma. 2015. Genetic Programming for Estimation of Heat Flux between the Atmosphere and Sea Ice in Polar Regions. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM, 1279–1286.
[45] Sean Stijven, Wouter Minnebo, and Katya Vladislavleva. 2011. Separating the wheat from the chaff: on feature selection and feature importance in regression random forests and symbolic regression. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. ACM, 623–630.
[46] Bruce Thompson. 1995. Stepwise regression and stepwise discriminant analysis need not apply here: A guidelines editorial. (1995).
[47] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), 267–288.
[48] Isaac Triguero, Joaquín Derrac, Salvador Garcia, and Francisco Herrera. 2012. A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 1 (2012), 86–100.
[49] C. J. TUCKER and P. J. SELLERS. 1986. Satellite remote sensing of primary production. *International Journal of Remote Sensing* 7, 11 (Nov 1986), 1395–1416. DOI: http://dx.doi.org/10.1080/01431168608948944
[50] Ekaterina J Vladislavleva, Guido F Smits, and Dick Den Hertog. 2009. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *IEEE Transactions on Evolutionary Computation* 13, 2 (2009), 333–349.
[51] Zhengming Wan. 2017. MOD11A2:MODIS/Terra Land Surface Temperature and Emissivity 8-Day L3 Global 1 km Grid SIN V006. (2017). DOI: http://dx.doi.org/10.5067/MODIS/MOD11A2.006 Years: 2002 to 2016 Tile: h20v02 Accessed: 06/01/2017.
[52] Wen-Chuan Wang, Kwok-Wing Chau, Chun-Tian Cheng, and Lin Qiu. 2009. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *Journal of hydrology* 374, 3 (2009), 294–306.
[53] Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics bulletin* 1, 6 (1945), 80–83.
[54] Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon. 2017. Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data.. In *AAAI*. 4559–4566.
[55] Jan Žegklitz and Petr Pošík. 2017. Symbolic Regression Algorithms with Built-in Linear Regression. *arXiv preprint arXiv:1701.03641* (2017).
[56] Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67, 2 (2005), 301–320.

Bibliography

[1] Nicholas Allgaier, Tobias Banaschewski, Gareth Barker, Arun LW Bokde, Josh C Bongard, Uli Bromberg, Christian Büchel, Anna Cattrell, Patricia J Conrod, Christopher M Danforth, et al. Nonlinear functional mapping of the human brain. *arXiv preprint arXiv:1510.03765*, 2015.

[2] Ignacio Arnaldo, Krzysztof Krawiec, and Una-May O'Reilly. Multiple regression genetic programming. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 879–886. ACM, 2014.

[3] Ignacio Arnaldo, Una-May O'Reilly, and Kalyan Veeramachaneni. Building predictive models via feature synthesis. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 983–990. ACM, 2015.

[4] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[5] Jarle W Bjerke, Stein Rune Karlsen, Kjell Arild Høgda, Eirik Malnes, Jane U Jepsen, Sarah Lovibond, Dagrun Vikhamar-Schuler, and Hans Tømmervik. Record-low primary productivity and high plant damage in the nordic arctic region in 2012 caused by multiple weather events and pest outbreaks. *Environmental Research Letters*, 9(8):084006, 2014.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[7] Stef F Bokhorst, Jarle W Bjerke, Hans Tømmervik, Terry V Callaghan, and Gareth K Phoenix. Winter warming events damage sub-arctic vegetation: consistent evidence from an experimental manipulation and a natural event. *Journal of Ecology*, 97(6):1408–1415, 2009.

[8] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

[9] David Buckingham. *Inductive learning of snowpack distribution models for improved estimation of areal snow water equivalent*. PhD thesis, University of Vermont, 2014.

[10] Ishanu Chattopadhyay and Hod Lipson. Data smashing: Uncovering lurking order in data. *Journal of the Royal Society Interface*, 11(101):20140826, 2014.

[11] Kamel Didan. Mod13a3:modis/terra vegetation indices monthly l3 global 1km grid sin v006, 2017. Years: 2002 to 2016 Tile: h20v02 Accessed: 06/01/2017.

[12] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[13] Mauricio Garcia-Limon, Hugo Jair Escalante, Eduardo Morales, and Alicia Morales-Reyes. Simultaneous generation of prototypes and features through genetic programming. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 517–524. ACM, 2014.

[14] Akhil Garg, Ankit Garg, and K Tai. A multi-gene genetic programming model for estimating stress-dependent soil water retention curves. *Computational Geosciences*, 18(1):45, 2014.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[16] Ulrike Grömping. Estimators of relative importance in linear regression based on variance decomposition. *The American Statistician*, 61(2):139–147, 2007.

[17] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[18] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[19] Ilknur Icke, Nicholas A Allgaier, Christopher M Danforth, Robert A Whelan, Hugh P Garavan, and Joshua C Bongard. A deterministic and symbolic regression hybrid applied to resting-state fmri data. In *Genetic Programming Theory and Practice XI*, pages 155–173. Springer, 2014.

[20] Ilknur Icke and Joshua C Bongard. Improving genetic programming based symbolic regression using deterministic machine learning. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 1763–1770. IEEE, 2013.

[21] MO Jeffries, J Richter-Menge, JE Overland, et al. Arctic report card 2015. 2015.

[22] Uday Kamath, Jessica Lin, and Kenneth De Jong. Sax-efg: an evolutionary feature generation framework for time series classification. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 533–540. ACM, 2014.

[23] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[24] Krzysztof Krawiec. Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines*, 3(4):329–343, 2002.

[25] Sam Kriegman, Marcin Szubert, Josh C Bongard, and Christian Skalka. Evolving spatially aggregated features from satellite imagery for regional modeling. In *Parallel Problem Solving from Nature - PPSN XIV*. Springer, 2016.

[26] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.

[27] Yingqiang Lin and Bir Bhanu. Evolutionary feature synthesis for object recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(2):156–171, 2005.

[28] Ammarin Makkeasorn, Ni-Bin Chang, and Jiahong Li. Seasonal change detection of riparian zones with remote sensing images and genetic programming in a semiarid watershed. *Journal of Environmental Management*, 90(2):1069–1080, 2009.

[29] Trent McConaghy. Ffx: Fast, scalable, deterministic symbolic regression technology. In *Genetic Programming Theory and Practice IX*, pages 235–260. Springer, 2011.

[30] Randall K McRee. Symbolic regression using nearest neighbor indexing. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pages 1983–1990. ACM, 2010.

[31] Nitin Muttil and Kwok-Wing Chau. Neural network and genetic programming for modelling coastal algal blooms. *International Journal of Environment and Pollution*, 28(3-4):223–238, 2006.

[32] Gareth K Phoenix and Jarle W Bjerke. Arctic browning: extreme events and trends reversing arctic greening. *Global change biology*, 2016.

[33] Cesar Puente, Gustavo Olague, Stephen V Smith, Stephen H Bullock, Alejandro Hinojosa-Corona, and Miguel A González-Botello. A genetic programming approach to estimate vegetation cover in the context of soil erosion assessment. *Photogrammetric Engineering & Remote Sensing*, 77(4):363–376, 2011.

[34] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

[35] Friedrich Recknagel. Applications of machine learning to ecological modelling. *Ecological Modelling*, 146(1):303–310, 2001.

[36] Miguel RomĂĄn, Dorothy Hall, and George Riggs. Modis/terra snow cover 8-day l3 global 500m grid, version 6, 2017. Years: 2002 to 2016 Tile: h20v02 Accessed: 06/01/2017.

[37] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[38] Michael Schmidt and Hod Lipson. Age-fitness pareto optimization. In *Genetic Programming Theory and Practice VIII*, pages 129–146. Springer, 2011.

[39] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.

[40] C Sivapragasam, R Maheswaran, and Veena Venkatesh. Genetic programming approach for flood routing in natural channels. *Hydrological processes*, 22(5):623–628, 2008.

[41] Guido Smits, Arthur Kordon, Katherine Vladislavleva, Elsa Jordaan, and Mark Kotanchek. Variable selection in industrial datasets using pareto genetic programming. *GENETIC PROGRAMMING SERIES*, 9:79, 2006.

[42] Guido F Smits and Mark Kotanchek. Pareto-front exploitation in symbolic regression. In *Genetic programming theory and practice II*, pages 283–299. Springer, 2005.

[43] Karolina Stanislawska, Krzysztof Krawiec, and Zbigniew W Kundzewicz. Modeling global temperature changes with genetic programming. *Computers & Mathematics with Applications*, 64(12):3717–3728, 2012.

[44] Karolina Stanislawska, Krzysztof Krawiec, and Timo Vihma. Genetic programming for estimation of heat flux between the atmosphere and sea ice in polar

regions. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1279–1286. ACM, 2015.

[45] Sean Stijven, Wouter Minnebo, and Katya Vladislavleva. Separating the wheat from the chaff: on feature selection and feature importance in regression random forests and symbolic regression. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, pages 623–630. ACM, 2011.

[46] Bruce Thompson. Stepwise regression and stepwise discriminant analysis need not apply here: A guidelines editorial, 1995.

[47] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[48] Isaac Triguero, Joaquín Derrac, Salvador Garcia, and Francisco Herrera. A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(1):86–100, 2012.

[49] C. J. TUCKER and P. J. SELLERS. Satellite remote sensing of primary production. *International Journal of Remote Sensing*, 7(11):1395â1416, Nov 1986.

[50] Ekaterina J Vladislavleva, Guido F Smits, and Dick Den Hertog. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *IEEE Transactions on Evolutionary Computation*, 13(2):333–349, 2009.

[51] Zhengming Wan. Mod11a2:modis/terra land surface temperature and emissivity 8-day l3 global 1 km grid sin v006, 2017. Years: 2002 to 2016 Tile: h20v02 Accessed: 06/01/2017.

[52] Wen-Chuan Wang, Kwok-Wing Chau, Chun-Tian Cheng, and Lin Qiu. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *Journal of hydrology*, 374(3):294–306, 2009.

[53] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[54] Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon. Deep gaussian process for crop yield prediction based on remote sensing data. In *AAAI*, pages 4559–4566, 2017.

[55] Jan Žegklitz and Petr Pošík. Symbolic regression algorithms with built-in linear regression. *arXiv preprint arXiv:1701.03641*, 2017.

[56] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.