2016

Evolving Spatially Aggregated Features for Regional Modeling and its Application to Satellite Imagery

Sam Kriegman
University of Vermont

Recommended Citation

Kriegman, Sam, "Evolving Spatially Aggregated Features for Regional Modeling and its Application to Satellite Imagery" (2016). *Graduate College Dissertations and Theses.* Paper 648.

Evolving Spatially Aggregated Features for Regional Modeling and its Application to Satellite Imagery

A Thesis Presented

by

Sam Kriegman

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 Statistics

October, 2016

Defense Date: August 5th, 2016
Thesis Examination Committee:

Jeff Buzas, Ph.D., Advisor
Josh C. Bongard, Ph.D., Chairperson
Chris Skalka, Ph.D.
Cynthia J. Forehand, Ph.D., Dean of the Graduate College

Abstract

Satellite imagery and remote sensing provide explanatory variables at relatively high resolutions for modeling geospatial phenomena, yet regional summaries are often desirable for analysis and actionable insight. In this paper, we propose a novel method of inducing spatial aggregations as a component of the statistical learning process, yielding regional model features whose construction is driven by model prediction performance rather than prior assumptions. Our results demonstrate that Genetic Programming is particularly well suited to this type of feature construction because it can automatically synthesize appropriate aggregations, as well as better incorporate them into predictive models compared to other regression methods we tested. In our experiments we consider a specific problem instance and real-world dataset relevant to predicting snow properties in high-mountain Asia.

Keywords: spatial aggregation, feature construction, genetic programming, symbolic regression

Citations

Material from this thesis was accepted for publication in the following form:

Kriegman S., Szubert M., Bongard J.C., and Skalka C.. (2016) Evolving Spatially Aggregated Features From Satellite Imagery for Regional Modeling. *Parallel Problem Solving from Nature - PPSN XIV*, Lecture Notes in Computer Science.

Acknowledgements

Thanks to Dr. Marcin Szubert for your patience and willingness to help me with virtually every aspect of my research. To Dr. Josh Bongard and Dr. Chris Skalka for your invaluable guidance and our weekly meetings. To members of the Morphology, Evolution and Cognition Laboratory as well as the Vermont Complex Systems Center for your constructive comments and our thought-provoking conversations. To the Vermont Space Grant Consortium and NASA EPSCoR for funding my graduate research assistantship. To Dr. Jeff Dozier (UCSB) for posing the high-mountain Asia SWE problem and providing associated datasets. And to the Vermont Advanced Computing Core which is supported by NASA (NNX 06AC88G), at the University of Vermont for providing High Performance Computing resources that have contributed to the research results reported within this paper.

Table of Contents

List of Figures

List of Tables

> *Those who are inspired by a model other than nature,*
> *a mistress above all masters, are laboring in vain...*
>
> Leonardo di ser Piero da Vinci

1

On The Origin of Regression

Charles Darwin's carefully constructed case for a theory of evolution by natural selection [1] implied a simple and eloquent explanation for all of life and its consequences.[1] Few books have ever had such an impact on either science or philosophy as Darwin's had on both. Although it would be another seventy years before Ronald Fisher, the father of frequentist statistics, finally united Mendelian genetics with natural

[1] The theory itself was in fact jointly proposed by Charles Darwin and Alfred Russel Wallace [2] one year prior to Darwin's 'On the Origin of Species [1].'

selection [3], grounding the theory in a mathematical formalism and unifying the larger scientific community in the belief that it was the basic mechanism of evolution. In the meantime, Darwin's half-cousin, Francis Galton,[2] sought to understand heredity: how characteristics present in individuals of one generation manifest in individuals of the subsequent generation.[3] It was the problem of heredity that provided Galton the initial inspiration to conceive the modern notions of correlation and regression.

In describing that extra large and extra small individuals tended to have offspring of a size closer to the sample mean, Galton employed the usual meaning of regression, 'to go back,' calling this phenomena 'regression toward mediocrity' and creating the basic foundations of what statisticians still call regression today [4]. In regression, or more generally in statistical modeling, there are two main goals in analyzing data [5]:

Prediction. To predict responses with future input variables;

Information. To extract some information about the association between response and input variables.

Of course the two are closely related– it's hard to justify associations using a model with poor prediction performance. In classical statistics, we typically assume that the data are generated by a given stochastic *data model*: a function of input variables, random noise, and parameters estimated from the data. Alternatively, *algorithmic models* treat the data mechanism as unknown and seek a function, an algorithm that operates on input variables to predict the response variables [5]. Figure 1.1 illustrates the difference between these 'two cultures' of statistical modeling as outlined by Leo

[2] Darwin and Galton shared the same grandfather, Erasmus Darwin, and the two shared a regular correspondence archived at `http://galton.org/letters/darwin/correspondence.htm`

[3] Galton, however, was interested in 'improving human stock' through the selective mating of humans, and is responsible for coining the term 'eugenics' and popularizing the discipline.

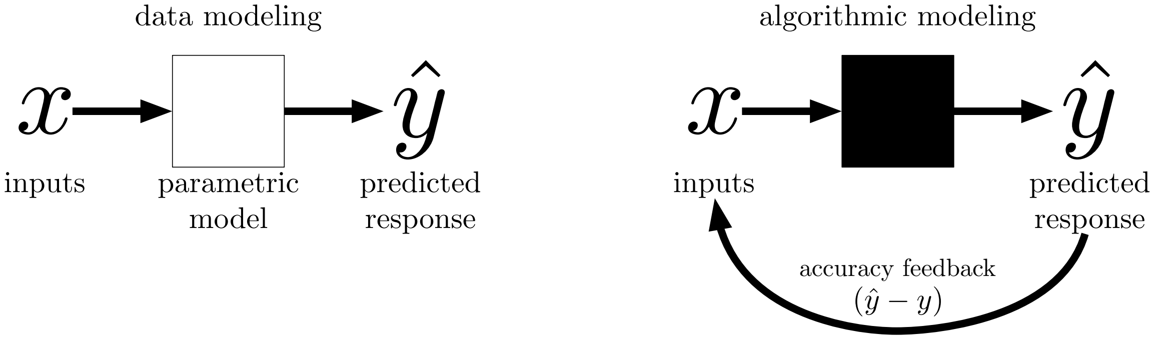


Figure 1.1: *The two cultures of statistical modeling presented by Breiman. Data modeling assumes the basic structure of the underlying relationship between inputs and response. Algorithmic modeling makes no assumptions about the true relationship and can be a black-boxed in pursuit of prediction accuracy.*

Breiman [5] in allusion to C.P. Snow's famous essay [6] about the cultural divide separating science and the humanities.

There are, inevitably, data problems more conducive to data modeling and others more so to algorithmic modeling. Though many problems lie somewhere in between. The benefit of data modeling is that it provides a simple explanation of the underlying data mechanism which in turn produces more testable hypotheses. However with strong model assumptions, any subsequent hypotheses are really about the validity of model rather than the structure of the underlying mechanism and this disconnect often leads to erroneous scientific conclusions [7–10]. In any case, data modeling assumes simplicity for interpretability, while algorithmic modeling accepts complexity for prediction performance.

Whether for prediction or comprehension, the central goal of statistics is to extract *useful* information from data. The emphasis is therefore not on interpretability, but accurate information. Higher predictive accuracy is associated with more reliable information about the underlying data mechanism. Therefore Breiman invites us to abandon “the belief that a statistician, by imagination and by looking at the data, can invent a reasonably good parametric class of models for a complex mechanism devised by nature,” and instead we should be satisfied with a model which generalizes well to previously unseen data.

1.1 A Modern Touch

The field of Statistics was born in a data-scarce environment. In the early days of statistics, typical data problems came from carefully designed experiments and were relatively small in scope: consisting of only a few dimensions, p, and a limited number of observations, n, but with $n >> p$. In turn, effective statistical methods leveraged probability theory at the cost of strong assumptions concerning how the data were generated. However, with the onset of the Information Age and the proliferation of computers, we now find ourselves in a data-rich environment.

Apart from an exponential increase in the amount and variety of data problems, their scope have exploded in both size and complexity. Algorithmic models generally provide the most accurate predictors possible for modern data problems, but inherently lack simple interpretability in their structure. “But the evolution of science is from simple to complex,” Breiman opines [5]:

> *There is no consideration given to trying to understand cosmology on the basis of Newton’s equations or nuclear reactions in terms of hard ball models for atoms. The scientific approach is to use these complex models as the best possible descriptions of the physical world and try to get usable information out of them.*

To be sure, data modeling continues to be the best solution to a meaningful set of problems but they are far outnumbered by larger, more complex data problems in the wild which have never heard of good experimental design.

The growing set of data-rich problems, along with a few of their more accomplished algorithmic solutions, gave rise to the field of Statistical Learning [11]. In statistical learning, models are often referred to as *learners* since, from our frame of reference, they appear to be 'learning' to recognize patterns in data (although this anthropomorphization is misleading [12, 13]). The preferred criterion to access the performance of models is generalization: a good learner is one that accurately predicts responses with unseen inputs. In other words, a good learner generalizes or adapts to new data environments.

With a rich set of robust examples of learning and adaptation from nature, it seems appropriate that many artificial learners explicitly mimic natural artifacts. After all, nature has always served as a great source of inspiration for scientist and engineers, a tradition which dates back at least as far as Leonardo da Vinci (1452 – 1519), who examined physiology of birds and fish in his famous designs of flying and swimming devices.

Today, in statistical learning, we have designed artificial neural networks [14] and immune systems [15], along with a myriad of algorithmic models imitating the behavior of grey wolves [16], monkeys [17], dolphins [18], cats [19], bats [20], eagles [21], cuckoos [22], tree frogs [23], fish [24], krill [25], bacteria [26], weeds [27], flowers [28], honey bees [29], fruit flies [30], fireflies [31], glow worms [32], ants [33], roaches [34], dand virtually every other type of insect [35] (but see [36]). Yet in doing so, we are ignoring the fact that these artifacts originated only after more than three billion years of evolution on this planet. Their structure is the accumulation of successive changes – ‘decent with modification,’ as Darwin called it – with each successive change increasing ability to survive and reproduce successfully [1]:

> *The framework of bones being the same in the hand of a man, wing of a bat, fin of the porpoise, and leg of the horse,–the same number of vertebrae forming the neck of the giraffe and of the elephant,–and innumerable other such facts, at once explain themselves on the theory of descent with slow and slight successive modifications.*

It follows that many aspects of biological structure can be understood only in the context of their ancestral heritage (if at all [37, 38]), obscuring the salient concepts of adaption we seek to reproduce. Therefore it may instead be more productive to mimic the process responsible for success rather than merely its result.

This distinction was famously made by Ernst Mayr’s (1961) ‘Cause and effect in biology [39],’ in which he distinguished between *proximate* and *ultimate causes*. A proximate cause is an immediate, mechanical explanation, whereas an ultimate cause is a historical explanation of why an organism has one trait rather than another [40]. Although Niko Tinbergen’s (1963) ‘On aims and methods of ethology [41],’ which followed shortly after Mayr’s work, provides a better framework by clearly distinguishing between past and present, as well as function and cause [42]. Tinbergen’s ‘four questions,’ distinguish between proximate ‘how questions,’ concerning how an individual organism’s structures function and ultimate ‘why questions,’ concerning why a species evolved the structures (adaptations) it has.

It may be impossible to know which proximate insights could and should be exploited for designing robust statistical models. Rather, the goal is to distill aspects of adaption through the recognition and imitation of ultimate processes. In fact this general philosophy underlies our success in arguably the most difficult engineering problem in human history: heavier-than-air flight. At first we attempted to emulate natural examples of flight and indeed many unsuccessful early attempts had flapping wings. From Leonardo da Vinci's 'flying machines' through subsequent designs over the next four hundred years, we fixated on the anatomy and mechanics of birds. It was only after we shifted focus from replicating proximate aspects of natural fliers, to the ultimate aerodynamic forces responsible for the capability of flight (drag and lift), that successful aircraft were constructed [43]. This insight enabled us to even surpass nature with designs capable of supersonic speed and space travel.

Natural evolution.

Biological organisms have evolved to solve incredibly complex problems in efficient and creative ways through essentially trail and error. According to Darwin, it is 'the struggle for existence' which eliminates some individuals before they are able to reproduce and pressures selection towards individuals with a facility to endure environmental and ecological conditions. Neo-Darwinian evolution links natural selection with an explicit method of heredity by which novel traits can arise to be selected for in the first place. Accordingly, adaptation, in the evolutionary sense, is often outlined by the following equation:

$$\textbf{adaptation} = \textbf{variation} + \textbf{heredity} + \textbf{selection}$$

where variation implies the existence of a population of at least two individuals that differ from one another to some extent [44,45].

Hereditary information is contained in an organism's genetic material, know as the *genotype*, whereas natural selection operates on the physical manifestation of the genetic information, known as the *phenotype*. Selected parents reproduce by transmitting their genotype to their offspring through an error-prone copying process. This small variation produces a range of traits in subsequent generation that in turn may affect their ability to survive and reproduce. However traits which are different but do not negatively effect individuals reproduction success may also be transmitted to future generations, a process known as neutral evolution [46]. Neutral genetic variants are thus hidden from selection and allowed to drift and accumulate in natural populations [38].

It is also important to recognize that current organisms may not be 'better' than previous generations when environmental and ecological conditions where different. This idea is expressed eloquently by Richard Dawkins in 'The Blind Watchmaker [47]':

> *Natural selection, the blind, unconscious, automatic process which Darwin discovered, and which we now know is the explanation for the existence and apparently purposeful form of all life, has no purpose in mind. It has no mind and no mind's eye. It does not plan for the future. It has no vision, no foresight, no sight at all.*

Yet, while genetic variation might be undirected, phenotypic variation is shaped by the processes of development which are in large part a product of evolution. Thus, random genetic changes might produce phenotypic changes that are 'informed' by past selection [48–50]. This idea, that environmentally induced phenotypes can become subject to heritable modification [51], implies the environment plays a formative as well as a selective role [38]. And that evolution cannot be completely reduced to a sequence of events whose unfolding is determined solely by natural selection and genetic code, but instead operates on a complex interplay between gene, organism and environment [52].

Artificial evolution.

Nils Aall Barricelli believed "a similar evolution should be possible with any kind of elements having the necessary fundamental properties," and in 1951 proposed "to perform numerical experiments by the use of large calculating machines, in order to clarify the first stages in the evolution of species [53]." In the spring of 1953 on one of the very first electronic computers, Barricelli took over the night shift between daily hydrogen bomb simulations lead by John Von Neumann [53]. Barricelli's experiments [54], which were ambitiously performed in 5 kilobytes, were as much about applying the powers of computing to evolution as they were about applying the power of evolution to computing.

Since Barricelli's pioneering work, algorithms inspired by neo-Darwinian evolution have proven capable of delivering high quality solutions to difficult problems in a variety of scientific and technical domains from robotics [55], to engineering [56–58], bioinformatics [59], and environmental modeling [60, 61]. Whereas natural evolution does not have a predetermined goal and is essentially an open-ended adaptation process, artificial evolution is an optimization procedure that attempts to find the best solution according to a predefined measurement – the *objective function* – which summarizes the performance or value of candidate solutions. Although in artificial

Algorithm 1 The basic evolutionary algorithm

```
$\mathcal{P} \leftarrow$ createRandomPopulation()
evaluatePopulation($\mathcal{P}$)
**while not** terminationCondition() **do**
    $\mathcal{S} \leftarrow$ selectParents($\mathcal{P}$)
    $\mathcal{P} \leftarrow$ recombineAndMutate($\mathcal{S}$)
    evaluatePopulation($\mathcal{P}$)
**end while**
**return** getFittestIndividual($\mathcal{P}$)
```

evolution this measurement is more commonly referred to as the *fitness function* since only the most ‘fit’ individuals are allowed to reproduce.

Evolution algorithms (outlined in Algorithm 1) generally involve a population of randomly generated individuals, or candidate solutions. Individuals are evaluated on a particular task based on a predefined fitness function and the worst performing individuals are deleted. Modified copies of the survivors are made by changing subtle aspects of an individual or combining aspects from a pair of individuals, in analogies to asexual mutation and sexual recombination (crossover), respectfully. Over many generations, the population tends towards more successful solutions [62–64].

From a computer-science perspective, evolutionary algorithms are stochastic (meta)heuristic search methods maintaining their working memory in the form of a population of candidate solutions [65]. Hence, it might be argued that evolutionary algorithms are not faithful models of natural evolution but merely a class of optimization procedures with an attached metaphor. Albeit the most successful metaphor used in the development of optimization algorithms, introducing many components and concepts which were truly new [36]. Indeed, evolutionary algorithms are an abstraction; it is simply not feasible to artificially synthesize biological evolution in every detail. However, they are unquestionably a form of evolution in their own right.

As Daniel Dennett [66] said “If you have variation, heredity, and selection, then you must get evolution.”

Most classical approaches to optimization involve hand designing a gradient or higher-order statistic of the objective function which, under regularity conditions, can be shown to generate sequences that asymptotically converge to locally optimal solutions [67]. Evolutionary algorithms are often used on difficult, poorly understood problems where these other methods fail or are trapped in suboptimal solutions. These problems typically include cases that have many free parameters with complex and nonlinear interactions, are characterized by noncontinuous functions, have missing or invalid data, an absence of subject-matter knowledge, or several local optima [45]. But even where adequate hand-designed solutions exist, they are tightly constrained by the way we think and function in our everyday physical world [12, 13, 68]. Evolutionary algorithms are less restricted by our innate human design bias and can generate unintuitive, and potentially superior, novel solutions.

There are many variations of the generic evolutionary computation template under various names differing in their representation of individuals (genotype), the genotype-phenotype mapping, the fitness function, the way in which ‘fit’ individuals are selected and how they subsequently reproduce. In perhaps the most flexible implementation, Genetic Programming (GP, [69]), individuals represent programs: algorithms which may be executed by a computer. Algorithms, in general, may represent any self-contained step-by-step set of operations which have the following five properties: (1) they must terminate after a finite number of steps, (2) they must be unambiguous, (3) they must accept input, (4) they must generate output, and (5) they must be reproducible, in principle by someone using paper and pencil [70]. Crucially, the size of a program and its genotypic representation (e.g. parse trees) are not fixed which facilitates the evolution of individuals with increasing complexity. This is important for data problems since we would like to explore a range of candidate solutions across various levels of complexity.

In GP, individual programs are constructed using a set of predefined components, known as *primitives*, which in turn define the space of possible resulting programs. Say, for example, that we would like to use GP to design an electrical circuit that performs some specific task (the details of which are unimportant here). One possible primitive set could include wire, resistors, capacitors, transistors, motors and integrated circuits. Although our primitive set should at least contain all of the necessary components required to build a minimally working circuit in this context. In other words, our set of primitives should be sufficient for the particular problem domain. However in many cases it is not be possible to know a priori exactly which primitives are required for sufficiency. Other primitives, not included in the minimally sufficient set, might be necessary to achieve adequate progress in optimization; but each new primitive exponentially increases the search space of possible programs.

1.2 Symbolic Regression

In classic statistical regression (or data modeling) we seek numerical coefficients of a mathematical relationship with a predefined functional form. In linear regression, for example, we search for coefficients of a linear combination of input variables that minimize the difference between the predicted and actual responses [71]. Symbolic regression [72–75] differs from classical regression in that we do not make assumptions about the functional form of the mathematical expression. Symbolic regression instead involves finding the mathematical expression itself, in symbolic form.

In GP-based symbolic regression, the individual programs are candidate regression models: free-form mathematical expressions which may be conveniently represented as parse trees (see Figure 1.2 for example). Accordingly, the primitive set will consist of arithmetic operators and operands which may be complied by evolution to form a mathematical function. Under *closure* – whereby all the variables, constants, arguments for functions, and values returned from functions are the same data type – we can ensure syntactically correct parse trees by simply restricting internal nodes to operators and leaf nodes (terminals) to operands. To handle multiple data types, the definition of what constitutes a legal parse tree has a few additional criteria (see [76]).

From now on we will refer to GP-based symbolic regression as simply 'GP' for convenience.

Walking through a run.

Suppose the actual mechanism generating our data is $3x^3 + 2x^2 + x + 1$, though we only observe $x \in \{-1.0, -0.9, \ldots, 0.9, 1.0\}$. How might an evolutionary process 'learn' the correct equation without knowing it beforehand? Let's walk through a small evolutionary run to get a sense of how GP works as an optimization procedure.

The first step is to specify our primitive set, defining the building blocks available to construct programs. Our primitive set will include operands: our observable variable x along with the constant 1.0; as well as the operators $\{+, -, \times, \div\}$. Next we must choose a fitness function to access the performance of candidate solutions. Let's use the sum of squared errors between the output of a candidate solution (its prediction) and the actual values observed at each datapoint.

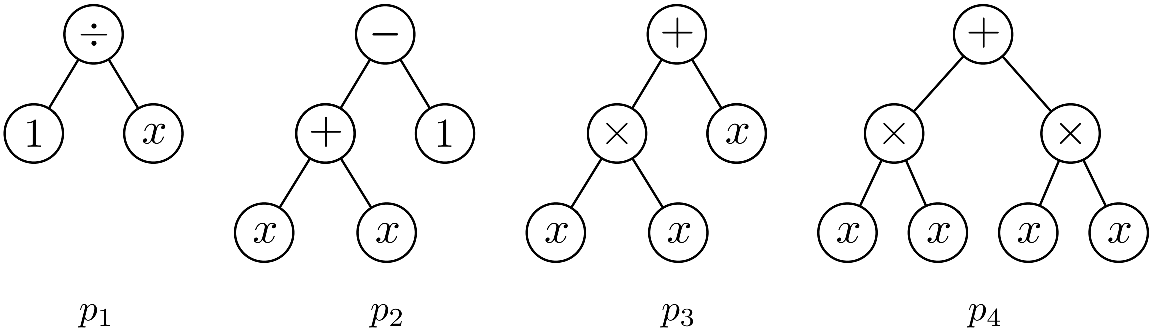


Figure 1.2: *A population of four programs.*

GP starts by randomly creating a population of individual programs, four in our case as illustrated in Figure 1.2. The first program (p_1) is equivalent to $1/x$. The second program (p_2) is $2x - 1$. The third program (p_3) is $x^2 + x$. The fourth program (p_4) is $2x^2$. Our programs are functions of x, so we will write them as

$$p_1(x) = 1/x, \quad p_2(x) = 2x - 1, \quad p_3(x) = x^2 + x, \quad p_4(x) = 2x^2.$$

The syntax of a program is it genotypic representation, and given a sample of datapoints, the output vector of predicted responses is its phenotype. Consequently, the behavior of each program corresponds to a point in n-dimensional space (in our case $n = 21$), known as the *sampling semantics* [77]. Selection operates on the sampling semantics (predicted responses) of programs through a fitness function which measures their distance from true semantics (actual responses) defined by $3x^3 + 2x^2 + x + 1$.

Randomly initialized combinations of components will naturally provide very poor solutions to most problems. However some programs will inevitably be better than others. The fitness (sum of squared errors) of our programs are: 368.5, 178.8, 77.1, and 94.7, respectfully. We will compare solutions pairwise, in what is known as *tournament* *selection*, deleting the inferior solution. First, p_1 is compared to p_2, and because $1/x$ has higher squared error than $2x - 1$, p_1 is deleted. Next, p_3 and p_4 are compared and p_4 is deleted. The remaining programs are $2x - 1$ and $x^2 + x$.

In tournament selection, only the best individual in the entire population is guaranteed to survive. For the rest of the population, it's the luck of the draw. We saw this when the two best solutions in our population, p_3 and p_4, competed against each other and the second best solution overall, p_4, was deleted. Moreover, p_2 survived while p_4 was deleted even though p_4 would have defeated p_2 if they met face-to-face. However, in the aggregate of larger populations the survivors will tend to be better overall than the individuals which were deleted, and the additional element of stochasticity may help our procedure escape local optima that would otherwise trap a more greedy selection process.

The survivors now have the opportunity to reproduce offspring which will fill in the two vacancies in our population. We use crossover to create a new program by swapping randomly chosen subtrees from (copies made of) the survivors, resulting in a new program $o_1(x) = 2x - x^2$ (Figure 1.3). And we copy p_3 with mutation to fill the final vacancy resulting in $o_2(x) = 2x^2 + x$ (Figure 1.4). It is important to note that in practice, mutation and crossover are applied probabilistically and mutually exclusive of one another: an exact/unmodified copy of an individual may occur, as well as a copy with both mutation and crossover.

Our population now includes p_2, p_3, o_1 and o_2, with fitness 178.8, 77.1, 125.7 and 56.6, respectfully. We repeat the steps of selection and reproduction again, restarting the loop. Note that selected programs in the first generation had fitness 178.8 and 77.1, whereas selected programs in the second generation (p_3 and o_2) have fitness

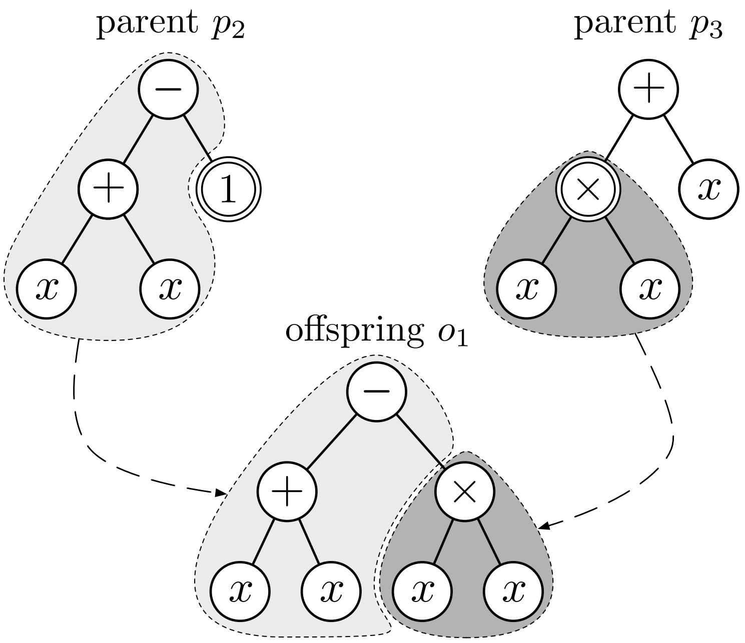


Figure 1.3: *Subtree-swapping crossover. The double edged nodes indicate a randomly selected crossover point. The subtrees may then be swapped between parents p_2 and p_3. Note that two potential offspring are made if we swap subtrees; the other possibility, not shown here, is $1 + x$.*

77.1 and 56.6. Thus selection and reproduction successfully improved the overall fitness of our population both in terms of the best individual and the average. Each successive generation, GP directs its search in the general directions of the survivors, incrementally replacing individuals with better alternatives (regression models with lower squared error). GP proceeds until some criterion of convergence is met or until a prespecified number of generations is surpassed. For more details on practical GP implementations see [78].

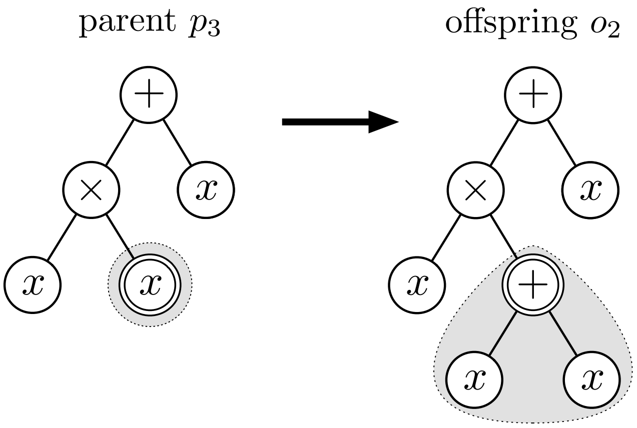


Figure 1.4: *Subtree-replacing mutation. A modified copy of* p_3 *which grew the subtree* $2x$ *in place of the terminal node* x.

Another objective.

GP lends itself well as a statistical tool for regression in complex, data-rich problems. There are no assumptions about model structure and the solutions are white-box: the resulting model's mathematical expression can be analyzed, unlike in artificial neural networks for example. Additionally, GP inherently performs dimensional reduction, only incorporating the variables chosen by the survivors. However there are some issues with the simple approach to GP summarized above.

In real data there is always noise accompanying the signal we attempt to uncover. Unless we incorporate a preference for simpler models our candidate solutions will become bloated in size and complexity, eventually overfitting the training data.[4] There are simple ways to remove this bias however. One approach is to explicitly incorporate selection pressure towards more concise solutions through an additional objective function. To do so we rely on multiobjective optimization and the notion of *Pareto*

[4] Bloat, the generational tendency towards larger trees in GP, also increases the CPU time required to evaluate and copy individuals– the two most computationally expensive procedures within GP.

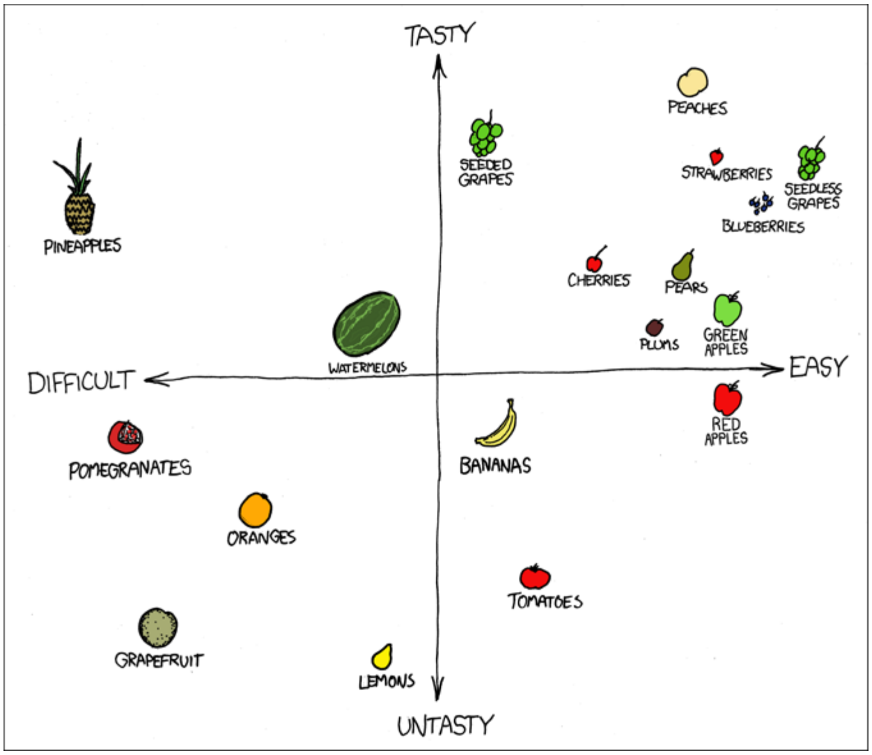


Figure 1.5: *An example of multiobjective optimization from xkcd.com (Randall Munroe, `https://xkcd.com/388/`). Maximizing taste and minimize difficulty, the Pareto front consists of peaches, strawberries, seeded and seedless grapes.*

dominance [79]. The goal is then to find solutions that are optimal according to all of the criteria (parsimony and accuracy) simultaneously.

Figure 1.5 illustrates the problem of multiobjective optimization in two dimensions, difficulty and taste. A fruit is said to Pareto dominate another if the first fruit is not inferior to the second in all objectives (may be equal), and is strictly better than the second in at least one objective. Oranges are better than pomegranates along the difficulty-axis, but are inferior to oranges along the tastiness-axis so neither dominate the other. Bananas are equivalent in taste to pomegranates but they are easier, thus bananas dominate pomegranates. The set of solutions that are non-dominated by any others is called the Pareto front. In our example, the Pareto front contains peaches, strawberries and seedless grapes. Since peaches are the tastiest, seedless grapes are the easiest, while strawberries are either easier or tastier than all other fruit. Note how blueberries would reside on the Pareto front if not for seedless grapes.

Incorporating this concept of dominance in tournament selection we can check, pairwise, if one solution dominates another. However it will now be more likely that a tournament match ends in a draw, so our population size might increase. Note that solutions on the Pareto front are implicitly guaranteed protection from deletion, whereas suboptimal solutions may still survive by chance in tournament selection.

In order to incorporate a preference for parsimony, it seems reasonable to define model complexity as the syntactic length of a candidate solution. If two models of different length have the same error then the shorter of the two must be a more concise representation and its simplicity will help with post hoc inference. Using error and complexity as objectives to drive evolution we can sort the final Pareto front of candidate regression models by length, simplest to most complex. More complex models will only be present if they have lower error than all smaller models in the population. The simplest model will contain a single constant term and will likely be the mean of the response value, though this really depends on the fitness function. The next model might incorporate our input variable in a simple linear combination. As we march down the Pareto front, we can extract useful information from the way in which input variables are incorporated and in what combinations, as well as how the form of the model changes when allowed more complexity.

A model is ultimately selected based on how well it balances the objectives. The selected model is then finally validated by calculating error on the test set of data held out from training.

Exploration, exploitation and diversity.

The start of evolutionary algorithms is know as the *exploration phase*, which involves globally investigating promising areas of the search space. In exploration, large changes in structure due to crossover and mutation are just as likely to be beneficial to fitness as they are to be detrimental. To support this phase, a population should consist of a diverse set of individuals. However after some relatively good solutions emerge in our population, we expect evolution will usually proceed by making fairly small adjustments to prior existing solutions. This is because drastic changes in structure often severely disrupt an individual's functionality– when one component is altered, it may no longer work in combination with other unchanged components. Evolution persists through locally refining solutions around the most promising regions, a phase of search know as *exploitation*.

In exploitation, better solutions will quickly begin to replicate more similar offspring which in turn reproduce even more similar offspring. This results in an exponential loss of diversity, limiting the scope of exploration, and can mean premature convergence to local optima. Yet without exploitation we are essentially performing random search by arbitrary jumping around the global search space. Similarly, without exploration we are just following the first randomly selected local optima we found. Exploration and exploitation are both necessary but antagonistic phases of search and finding a proper balance between the two is a challenging task.

One solution is to promote diversity through adding a third independent optimization objective that, either implicitly or explicitly, rewards individuals for behaving differently than other individuals in the population. As an example of implementing three objectives, we might add another goal of nutrition to our fruit optimization. The individual fruit would then be pushed back or pulled forward along a third axis in our illustration. Consequentially, fruit have an additional opportunity to be nondominated and more individuals will reside on our Pareto front.

> *When once you have tasted flight, you will forever walk the earth with your eyes turned skyward, for there you have been, and there you will always long to return.*
>
> Leonardo di ser Piero da Vinci

2

Satellite Imagery

Satellite imagery is a quintessentially modern source of data problems which are inherently complex. The data derived from this imagery can be enormous in spatial and temporal dimensions, and its scope raises important questions concerning scale and meaningful units of estimation. Moreover, remotely monitoring phenomena from outer space can introduce measurement error, irregularly distributed in geographic space, which may dampen signals in unintuitive ways at different scales. It is naive to think that a simple model can adequately explain such a complex convolution of system components and measurement error. Yet even if a particular phenomena was truly the simple function of potentially observable variables at some scale, most Earth systems are still not well understood and lack the domain knowledge necessary to realize the simple data model. The most efficient way to perform any statistical analyses is by first finding solutions with adequate predictive accuracy and only then figuring out why they work so well. Theory and insight to domain scientists may then follow this empiricism as we decompose particularly useful solutions.

2.1 Towards Meaningful Units

Regional modeling focuses on explaining phenomena occurring at a regional, as opposed to site-specific or global scales [80]. Regional models are of interest in many remote sensing applications, as they provide meaningful units for analysis and actionable insight to policymakers. Yet satellite imagery and remote sensing provide variables at relatively high resolutions. Consequently, studies often involve decisions concerning how to integrate this information in order to model regional processes. Considering measurements at each individual spatial unit as a separate model feature can result in a high dimensional problem in which high variance and overfitting are major concerns. For this reason, spatial aggregation is often applied in this setting to uniformly up-sample variables to be consistent with the response. Although in averaging variables across all spatial units in the region, we discard information which could in turn diminish prediction accuracy and our understanding of underlying phenomena.

Rather than strictly incorporating individual spatial units or uniformly up-sampling, it might instead be beneficial to construct features of a regional model using particularly important subsets of geographical space. In this paper, we move away from uniform up-sampling aggregations towards more flexible and interesting aggregation operations predicated on their subsequent use as features of a regional model. We propose a novel method of inducing spatial aggregations as a component of the statistical learning process, yielding features whose construction is driven by model performance rather than prior assumptions.

Related work. The general problem of modeling a response using features at a different scale is closely related to the *modifiable areal unit problem* [81] and the more general *change of support problem* [82]. These problems center around the different inferences obtained when the same set of data is grouped at increasingly higher scales or in alternative formations at the same scale. However, to the best of our knowledge, this relationship has not been explicitly exploited to improve prediction accuracy.

2.2 A Complex System

In experiments designed to explore these techniques, we consider a specific problem and real dataset: estimating the volume of water in snow – the Snow Water Equivalent (SWE) – in the Hindu Kush range of high-mountain Asia (Figures 2.1,2.2). A region which spans most of Afghanistan and extends into parts of Pakistan, India, China, Tajikistan, Uzbekistan, Turkmenistan, and Iran.

The accumulation of snow is a vital source of water for natural systems and humans [83]. For humans, snow is important because it forms its own reservoir [84], providing both flood control and water storage by capturing water in solid form in cold months and releasing it in warm months, concurrent with higher agricultural

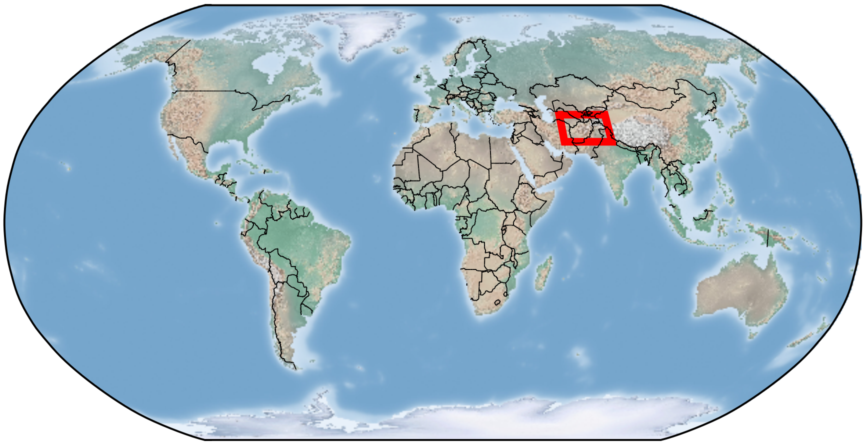


Figure 2.1: *The Hindu Kush region of High Mountain Asia, bounded by the red polygon, contains most of Afghanistan and extends into parts of Pakistan, India, China, Tajikistan, Uzbekistan, Turkmenistan, and Iran.*

and evapotranspirative demands [83, 85]. With more than one-sixth of the Earth's population relying on seasonal snow packs for their water supply [84], reliable SWE estimates are critical for resource management. Beyond potable drinking water, precise information about the water volume stored in the snowpack is necessary in the evaluation of hydroelectric power, sanitation, manufacturing, agriculture and environmental protection.

Accurate models of SWE can serve as a monitoring system by providing a benchmark to measure the advance of global warming, which influences the timing and magnitude of accumulation and melt [83, 85]. Moreover, irregular amounts of SWE can signal the onset of hydrologic extreme events like drought and flood [86], which are among the most influential environmental stressors affecting the development of human societies [87]. Monitoring SWE in this particularly unstable geopolitical region is especially important considering societal vulnerability to climatological events – most prominently drought – has led to societal disintegration, armed conflicts, and

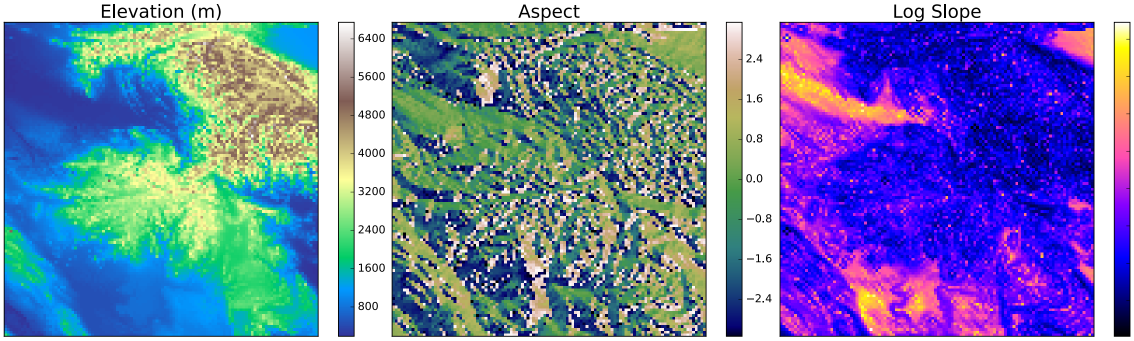


Figure 2.2: *Topography of the Hindu Kush region of High Mountain Asia. From left to right: elevation in meters, aspect in radians, and log slope in log radians.*

eventually societal collapse [88–93].

Unfortunately, however, accurate SWE estimation is notoriously difficult due to the complex characteristics of snow distribution [94] and the challenges of monitoring it in mountainous regions across many national boundaries [86]. High-mountain Asia is especially sensitive in this respect, and furthermore suffers from a dearth of relevant ground-based sensors, meaning satellite imagery is the sole source of data. And current estimation techniques based on this imagery fall short [86]. New models will need to include a more faithful representation of surface-water processes and provide a continuity of observations to account for nonstationarity [95].

Thus, our broader practical goal is improved near-real-time estimation of SWE in this region. We aim in particular to estimate regional SWE (the response variable) of the Hindu Kush range, given a set of explanatory variables that are measured across a regular grid nested within the response. Regional SWE is of scientific and practical interest, and furthermore the methods we explore here can be scaled down to smaller prediction areas using the same data sources, e.g. basin scale, though this is beyond the scope of this paper.

> *Although nature commences with reason and ends in experience it is necessary for us to do the opposite, that is to commence as I said before with experience and from this to proceed to investigate the reason.*
>
> Leonardo di ser Piero da Vinci

3

Experiments and Results

We take a comparative approach to the SWE problem, considering ridge regression, lasso, and GP-based symbolic regression.[1] For each regression model, we consider a filter-based method of feature construction in addition to a second, more dynamic method. For linear regression, we incorporate a wrapper approach in which constructed features and the regression model are induced in separate learning processes, with

[1] The source code necessary for reproducing our results is available at `https://github.com/skriegman/ppsn_2016`.

Table 3.1: *Regression models and their implemented feature construction methods.*

	Standard	Filter	Wrapper	Embedded
Ridge	✓	✓	✓	×
Lasso	✓	✓	✓	×
GP	✓	✓	×	✓

feedback between the two. For symbolic regression, we use an embedded approach where constructed features and the regression model are induced simultaneously over the course of an evolutionary run. Table 3.1 provides a summary of our methods, indicating which feature construction methods are implemented in combination with particular regression models.

The Dataset.

The SWE dataset is derived from data collected by NASA's Advanced Microwave Scanning Radiometer (AMSR-E; aboard the Aqua satellite) and Moderate Resolution Imaging Spectroradiometer (MODIS; aboard the Terra and Aqua satellites) for March 1 - September 30, in 2003 - 2011, over the Hindu Kush region of high-mountain Asia.[2] We have three explanatory variables measured daily across a 113×113 regular grid within the region for 1935 days. The first explanatory variable, a, is a physical estimate of SWE itself derived from AMSR passive microwaves [96–98]. This passive microwave-based SWE estimate has a number of issues which are outlined in [86] and highlighted in Table 3.2. Additionally, there are two explanatory variables derived from MODIS data which measure different statistics of the fraction of snow covered area at a *sub-pixel* level [99–101]. Concretely, in addition to a, we have sub-pixel snow

[2] Raw satellite data was pre-processed by Dr. Jeff Dozier (UCSB) using previously reported techniques and is available upon request.

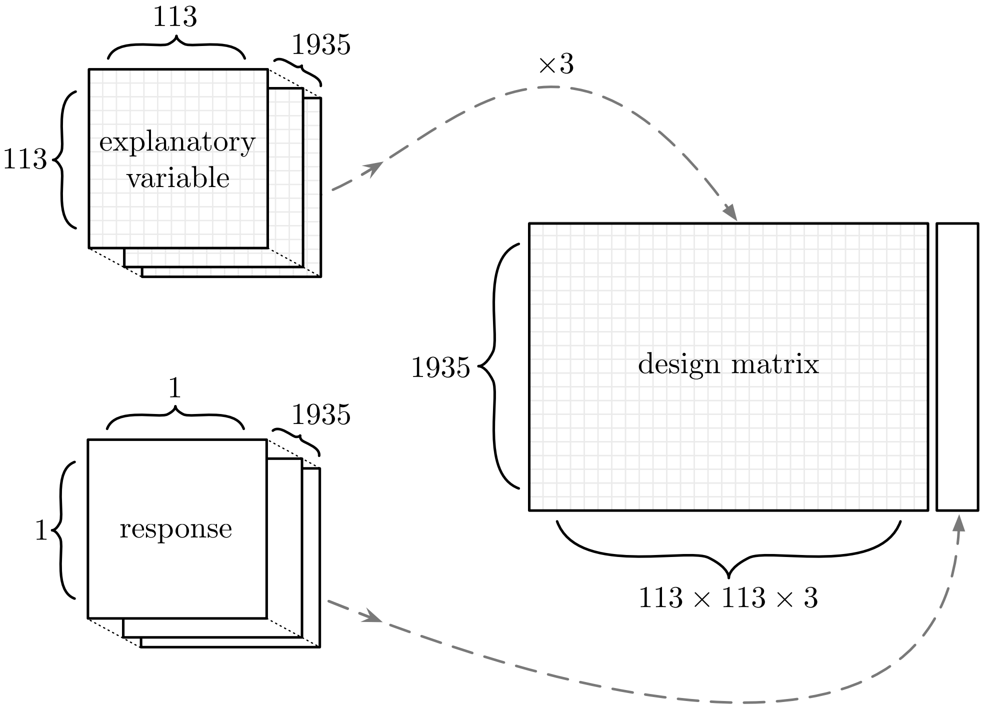


Figure 3.1: *From rasters to panel data (without spatial aggregation). Incorporating measurements at each of the* 113×113 *individual spatial units, of each of our 3 explanatory variables, results in* $113 \times 113 \times 3$ *features – or columns in the design matrix – of a regional model. The response summarizes the entire study region with a single value for each of the 1935 days.*

covered area mean, m_μ, and sub-pixel snow covered area standard deviation, m_σ.

Whereas each of the explanatory variables $(a,\ m_\mu,\ m_\sigma)$ are measured across a 113×113 raster image, the response variable is regional SWE, s, an attribute of the entire study region, represented as a single 1×1 value for each of the 1935 days (Figure 3.1). The response, s, was 'reconstructed' by combining snow cover depletion record with a calculation of the melt rate to retroactively estimate how much snow had existed in the region (see [102] for details). While the explanatory variable a and the response s both represent an estimate of SWE, a is inaccurate but available on a daily basis, whereas s is considered 'ground truth' but available only retroactively after the snow has melted.

Table 3.2: *Correlation between uniformly upsampled explanatory variables and the regional response, s.*

	a	m_μ	m_σ	s
a	1.0			
m_μ	0.7343	1.0		
m_σ	0.6567	0.8631	1.0	
s	0.4349	0.7423	0.6677	1.0

Without any spatial aggregation, each of the 113×113 spatial units (pixels) within the region, for each of the 3 explanatory variables, can be treated as a separate regional model feature (Figure 3.1). In other words, the columns of the design matrix correspond to a particular explanatory variable at a particular spatial location within the region, and the rows of the design matrix correspond to their 1935 daily measurements. There are $113 \times 113 \times 3 = 38307$ features without considering any interaction terms.

Table 3.2 compares the Pearson correlation[3] between the regional response and our three explanatory variables after mean uniform upsampling (their mean across space for each day). AMSR SWE (our explanatory variable a) is a standard approach to modeling SWE with satellite imagery that is used in practice. However, the correlation between the upsampled AMSR SWE estimate, a, and the retroactive regional ground truth, s, is particularly low at 0.4349. This disparity is the motivation behind pursuing inductive estimates of SWE and incorporating the related MODIS variables.

[3] The Pearson product-moment correlation coefficient – the covariance of two variables divided by the product of their standard deviations – was developed by Karl Pearson, however it is based on the ideas originally introduced by Francis Galton.

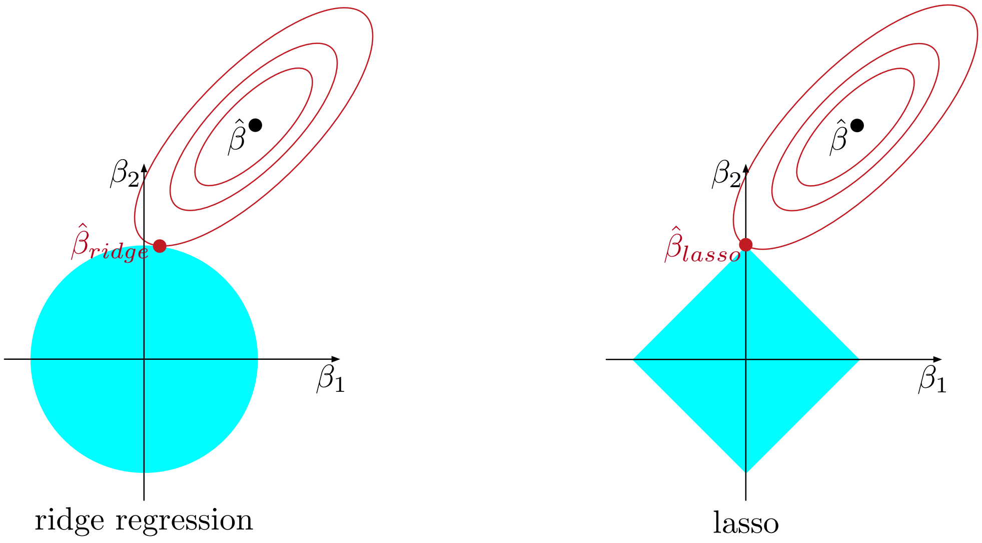


Figure 3.2: *The geometry of ridge regression (left) and lasso (right) constraints as solid blue regions in two dimensions of coefficients, β_1 and β_2. The red ellipses represent the contours of the error function (sum of squares) as it moves away from the unconstrained minimum at $\hat{\beta}$ (the OLS solution). The biased solutions of ridge and lasso, depicted as red points, are restricted to reside along the perimeter of their blue constraint. Figure adopted from [11].*

3.1 Regression Models

Ridge regression [103] is similar to ordinary least squares (OLS) but subject to a bound on the L_2-norm of the coefficients. Because of the nature of its quadratic constraint, ridge regression cannot produce coefficients exactly equal to zero and keeps all of the features in its model (Figure 3.2). Lasso (Least Absolute Shrinkage and Selection Operator, [104]) modifies the ridge penalty and is subject to a bound on the L_1-norm of the coefficients. The geometry of this L_1-penalty has a strong tendency to produce sparse solutions with coefficients exactly equal to zero (Figure 3.2). In many high dimensional settings, lasso is the state-of-the-art regression method given its ability to produce parsimonious models with excellent generalization performance. For both lasso and ridge regression, the parameter constraining the coefficients is set through the default cross-validation search procedure in Python's scikit-learn.

Genetic Programming (GP, [69]) is a very flexible heuristic technique which can conveniently represent free-form mathematical equations (candidate regression models) as parse trees. GP's inherent flexibility is well-suited for our particular problem because it can efficiently express spatial aggregations and seamlessly combine them into the learning process with minimal assumptions. Furthermore, the white box nature of GP may provide physical insights about this complex problem that is currently lacking, as in other domains [73, 105].

To search the space of possible GP trees we use a variant of Age-Fitness Pareto Optimization (AFPO, [106]). AFPO is a multiobjective method that relies on the concept of *genotypic age* of an individual, defined as the number of generations its genetic material has been in the population [107]. The age attribute is intended to protect young individuals before being dominated by older already optimized solutions. Each randomly initialized individual starts with age of one which is then incremented by one every generation. An offspring inherits age of the older parent.

The AFPO algorithm starts with a population of n randomly initialized individuals. In each generation, it proceeds by selecting random parents from the population and applying crossover and mutation operators (with certain probability) to produce $n - 1$ offspring. The offspring, together with a single randomly initialized individual, are added to the population extending its size to $2n$. Then, Pareto tournament selection is iteratively applied by randomly selecting a subset of individuals and removing the dominated ones until the size of the population is reduced back to n.

We extend AFPO to include an additional objective of model size, defined as the syntactic length of an individual tree. The size attribute protects parsimonious models which are less prone to overfitting the training data. To determine which individuals are dominated, the algorithm identifies the Pareto front using using three objectives (all minimized): age, error (fitness), and size. For the fitness objective, we use a correlation-based function rather than pure error, and define

$$f_{COR} = 1 - |\phi(\hat{s}, s)|$$

where $\phi(\hat{s} - s)$ denotes Pearson correlation between model predictions ($\hat{s}$) and actual values of our response (s), regional SWE. Correlation has recently been shown to outperform error-based search drivers given that if a model makes a systematic error it could be easily eliminated by linearly scaling the output and therefore should be protected [61]. Accordingly, for all GP implementations, we apply a linear transformation after f_{COR} -driven evolution has concluded, by using an individual program (model) output as the single input of OLS on the training data.

We used the settings in Table 3.3 for all implemented GP experiments. Each experiment consists of 30 trials, from which the best model (lowest training f_{COR}) is selected. The selected model is then transformed using OLS, and subsequently validated using unseen test data.

Standard Methods. Ridge regression, lasso, and GP may be performed on the raw data using each variable at each individual spatial unit as a separate feature (Figure 3.1). We denote these methods as Standard Ridge (SR), Standard Lasso (SL) and Standard GP (SGP). SR, SL and SGP each have access to $113 \times 113 \times 3 = 38307$ features, but only 1720 observations in each fold of data.

Table 3.3: *Genetic programming settings.*

Parameter	Value
population size	1000
generations	1000
initialization	ramped half-and-half height range $2 - 6$
instruction set	$\{+, -, \times, /, exp, log, sin, cos\}$
tournament size	2
crossover probability	0.75
mutation probability	0.01
maximum tree height	17
maximum tree size	300
number of runs	30

3.2 Feature Construction Methods

Feature construction is a well studied problem and the utility of genetic programming for feature construction has been recognized in many previous studies [108]. The key difference in our work from this past work is the nature of the data being modeled. We presume that there exist spatial autocorrelations of varying size and shape that, if aggregated to improve the signal to noise ratio, yield features supporting more accurate predictions.

In a regional model, we can construct features by aggregating higher dimensional variables across space. However, it is not entirely clear what kind of aggregations are useful as features of a predictive model. Grouping variables based on similarity or dissimilarity does not necessarily produce useful regional features. In this paper, we make an assumption about the importance of distance and continuity in effective spatial aggregations, based on Tobler's first law of geography [109] which states that "everything is related to everything else, but near things are more related than distant things.” Accordingly, we limit the space of possible spatial aggregations to be an average of values within a circular spatial area defined by its centerpoint and radius. However, where to aggregate, how many aggregations to perform, and how to combine the aggregates must still be determined manually or decided during model optimization. We view filters and wrappers as intermediary steps in relaxing assumptions towards our embedded approach, which automates all three of these aspects.

The Filter Method.

Filter-based feature construction methods transform or ‘filter’ the original variables as a preprocessing step, prior to modeling. Our filter for the SWE problem represents a static up-sampling transformation of the original variables. Each variable is decomposed in space by a grid of overlapping circles[4] of equal radii centered on a square lattice pattern of points (see Figure 3.3 for example). Each constructed feature corresponds to the average (arithmetic mean) of a particular variable sampled within a particular circle of space. Units that reside in an overlapping region of two separate circles are included in the calculation of both features. Since there are three explanatory variables in the SWE dataset, an $R \times R$ grid corresponds to $p = 3R^2$ constructed features. The constructed features are then used as inputs for ridge regression, lasso, and GP, which we will refer to as Filtered Ridge (FR), Filtered Lasso (FL), and Filtered GP (FGP). We will also specify the value of R used in a particular model instance as a subscript, e.g. FR_{15} denotes Filtered Ridge with R=15. We consider filters with $R \in \{1, 2, \ldots, 20\}$, however note that the standard methods are essentially filters with $R = 113$, albeit with the non-overlapping square pixels.

[4] The shape of circles are in reality so-called ‘small circles,’ as they lie on the surface of earth.

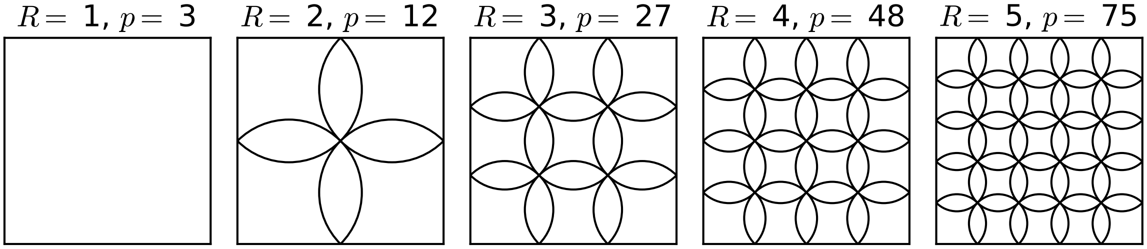


Figure 3.3: *Overlapping circle grids for a particular variable representing the regions sampled by the filter approach Each of the three SWE variables are decomposed by overlapping circle grids at resolution R resulting in* $p = 3R^2$ *constructed features.*

The Wrapper Method.

Wrapper-based feature construction methods incorporate feedback from the fit of the model. We implement wrappers around both ridge regression and lasso in order to enable the circular sampling regions to define their own center and radius. The circles are no longer fixed on a grid with a predetermined size. Instead, each constructed feature is uniquely parameterized by the coordinates of a center unit (x, y), as a latitude and longitude tuple, and a radius r, as a single value floating point number in km. The center can be any spatial unit in the region, including one at the edge of the raster. The radius is restricted to be within 0 and 1000 km, which is flexible enough to contain only a single unit or span the entire region (see Figure 3.5b,d for example).

Wrapped Ridge (WR) and Wrapped Lasso (WL) separately use a ridge/lasso-driven hill climbing algorithm to construct features that minimize Mean Absolute Error (MAE), i.e.

$$\frac{1}{n}\sum_{i=1}^{n}|\hat{s_i} - s_i|$$

where s_i is the actual value of our response (regional SWE) and $\hat{s_i}$ is output predicted by the model over n observations. The algorithm uses the same number of circles for each of the three variables, initializing their parameters (x, y, r) randomly. For 1000 iterations, a single constructed feature (circle) is randomly selected and subject to a Gaussian mutation on one of its parameters with standard deviation equal to 25% of the radius and centered at zero. A new ridge/lasso model is then refit on the mutated set of features using a random subset of data sampled without replacement. If the mutation lowered model error on the complementing set of training data left out, then the change is accepted. Otherwise, the mutation is undone. If a proposed mutation to the radius would take it outside the restricted range of $0-1000$ km, then it is 'bounced-back' the distance it would have exceeded the boundary. For example, a random mutation that would result in a radius of 1200 km, becomes $1000-(1200-1000) = 800$ km. Thirty restarts are used from which the best model based on training data is selected. We consider $R \in \{1, 2, 3, 4\}$ for wrappers corresponding to $3 \times R^2$ features which really means $3 \times 3 \times R^2$ modifiable parameters.

The Embedded Method.

By using GP, we can allow for flexibility with respect to the placement and number of aggregations as well as the way in which they are combined to form a model. However, stochastic optimization methods like GP cannot be easily 'refit' in the same manner as deterministic algorithms like ridge regression or lasso. Therefore using wrapper approach for GP is computationally infeasible. Instead, modifications to aggregated features are implemented through mutation-based operators.

In Genetic Programming with Embedded Spatial Aggregation (GPESA) introduced here (Figure 3.4), our constructed features are represented as parameterized tree terminals, with parameters (x, y, r). Constructed features are randomly initialized

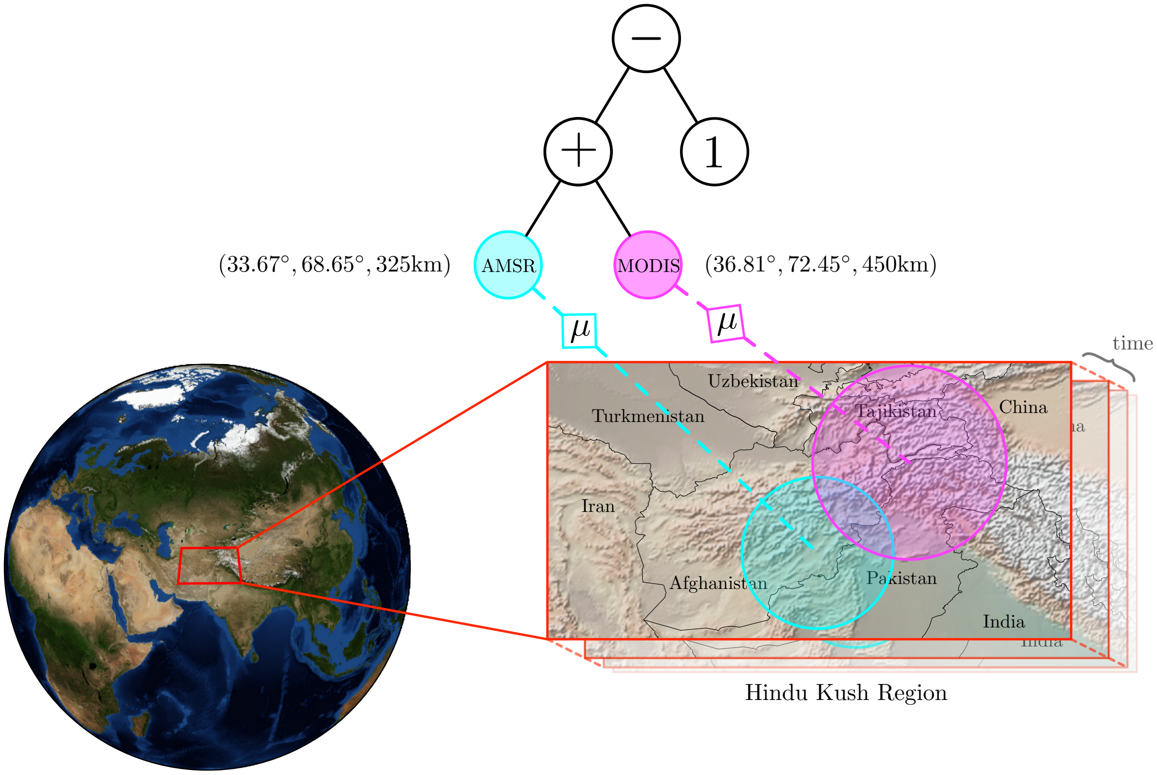


Figure 3.4: *GPESA trees employ specialized terminals which take the average of a spatially distributed variable within an adjustable circle of geographical space, specified by their centerpoint (longitude, latitude) and radius.*

in the same manner as the wrapper method, but separately for each terminal of each individual in the population. Greedy Gaussian mutations to the parameters (x, y, r) of a randomly selected constructed feature occur in the population with 20% probability, each generation. Mutations to r have mean zero and a standard deviation of 25%, subject to the bounce-back rule. Similarly, mutations to (x, y) have mean distance zero and a standard deviation of $0.25r$. For 25 iterations, greedy mutations modify the parameterized terminals within a particular GP tree. A modification is accepted if it successfully reduces average error (f_{COR}) on random subsets of training data sampled with replacement. Aside from the stochastic application, another key difference between the wrapper method's hill climbing algorithm and the GPESA's greedy mutations is that the overall regression model stays the same between mutations rather than being refit after each mutation.

Validation.

In order to validate the generalization of models we partition the dataset into nine overlapping folds. Each fold corresponds to leaving out one year for testing and training on the remaining eight (using years 2003 - 2011). We use MAE on the unseen test data as a metric to assess model performance. To account for a difference in scale across any set of features, all input model features are standardized over time by removing the mean and scaling to unit variance. This means that as wrapper and embedded methods construct new aggregations, the sampled data is scaled over time prior to being averaged over space. Since our goal is near-real-time estimation for a future day, the training values of a feature's mean and variance are reapplied when scaling the same feature in validation.

3.3 Results

Table 3.4 displays the test error of each valid regression and feature construction method combination. For filters and wrappers, only the best performing model is displayed in Table 3.4 and we indicate the particular value of parameter R as a subscript (see Figure 3.6a for all filter results). Since the ultimate goal of our paper is to synthesize a method better than existing approaches, we must statistically compare GPESA to SL, the state-of-the-art linear regression / variable selection algorithm. The null hypothesis of interest here is that of no difference between GPESA and a

Table 3.4: *Median mean-absolute error with corresponding standard errors in parentheses. Only the best testing filter- and wrapper-based results (choice of R) are displayed. We explicitly compare GPESA with the state-of-art, SL. Bold values indicate significance (at 0.05 level with Bonferroni correction) under a Wilcoxon singed rank test in which the null hypothesis asserts that distribution of the differences between GPESA and SL is symmetrically distributed about 0.*

Year	SR	SL	SGP	FR$_4$	FL$_{19}$	FGP$_{19}$	WR$_2$	WL$_3$	GPESA
2003	0.86	0.51	0.35 (0.14)	0.50	0.46	0.44 (0.08)	0.43 (0.10)	0.49 (0.09)	**0.29 (0.09)**
2004	0.47	0.30	0.32 (0.10)	0.34	0.29	0.26 (0.05)	0.37 (0.16)	0.35 (0.16)	**0.17 (0.05)**
2005	0.95	0.44	0.50 (0.13)	0.61	0.40	0.52 (0.06)	0.58 (0.11)	0.63 (0.09)	**0.32 (0.07)**
2006	0.66	0.27	0.41 (0.29)	0.57	0.52	0.36 (0.06)	0.53 (0.11)	0.54 (0.11)	0.27 (0.05)
2007	0.72	0.33	0.44 (0.10)	0.42	0.38	0.34 (0.05)	0.52 (0.13)	0.50 (0.11)	**0.24 (0.06)**
2008	1.46	0.46	0.60 (0.13)	0.71	0.64	0.58 (0.11)	0.70 (0.31)	0.54 (0.26)	0.52 (0.18)
2009	0.81	0.41	0.65 (0.08)	0.90	0.61	0.56 (0.08)	0.98 (0.10)	1.03 (0.09)	0.41 (0.10)
2010	0.62	0.48	0.44 (0.12)	0.43	0.47	0.41 (0.06)	0.43 (0.11)	0.52 (0.11)	**0.32 (0.07)**
2011	0.87	0.48	0.61 (0.17)	0.77	0.60	0.53 (0.10)	0.82 (0.20)	0.93 (0.16)	0.45 (0.12)
Mean	0.82	0.41	0.48	0.58	0.49	0.44	0.58	0.61	0.33

SL. Therefore we perform yearly Wilcoxon signed rank tests [110] comparing GPESA to SL with Bonferroni correction across the nine years. For five out of the nine test years, GPESA is significantly better than SL, while for the other four years there is no significant difference with SL.

Through displaying only the best testing filters and wrappers, we aim to focus speculation about GPESA performance through a conservative lens. Yet we ultimately view filters and wrappers as intermediary steps ‘working up’ to GPESA. Accordingly, the best test error better represents a bound on the potential performance of a particular intermediary method even though it may not be possible to achieve such performance through a parameter sweep based on the training data. And indeed, across all methods tested, GPESA reported the lowest recorded median mean-absolute error within all but two years (7 of 9) where it has the second lowest.

3.4 Discussion

Our results show that incorporating dynamic aggregations of higher resolution variables into a regional model is beneficial in our particular problem setting, as compared to both uniform up-sampling of variables and a state-of-the-art linear regression technique (SL) that incorporates individual spatial units. SL achieves competitive prediction performance through a sparse linear combination of the individual spatial units, on par with SGP which is not linearly constrained. Ultimately, GPESA performed significantly better (lower median test error) than SL on a majority (5 of 9) of cross validation folds. Moreover, whenever GPESA was not significantly better than SL it was not significantly worse.

A main reason why GPESA has an advantage in this application is the difficulty of knowing a priori what the most important spatial datapoints are, and how to best aggregate them. Additionally, the structure of the model itself is unknown and it depends on the resulting aggregations. Therefore this is not a fixed length optimization problem, which makes it well-suited for GPESA, which can search over different numbers and non-linear combinations of spatial aggregations. While SL can theoretically perform the same aggregation as a GPESA terminal (mean within a radius of a geographical point), SL is restricted to a single linear solution while GPESA is not.

However, it's important to emphasize that the computational cost of GPESA is higher than that of traditional GP and much higher than that of linear regression. In particular, the most expensive operation is the 'on the fly' aggregation component of GPESA which makes the fitness evaluation require 500% more time than in SGP. Part of the incurred cost is due to inefficiencies of our implementation that necessitated a copy with all spatial aggregation operations. In future work we will look at reducing this overhead through more efficient data structures (e.g. k-d trees).

Importance of Spatial Units.

To better understand the relevance of particular spatial locations, we define the *importance* of a spatial unit for both linear and symbolic methods, separately. For ridge regression and lasso, we can define importance by exploiting the disposition of coefficients to be larger for variables with a stronger correlation to the response, relative to a particular feature set. We define linear regression importance of a particular spatial unit as the average absolute coefficient of features that incorporate the unit into a regression model. While we cannot as easily determine relative importance within nonlinear models, we can instead define importance by exploiting the multiple candidate solutions provided from stochastic multiobjective optimization. We define GP importance of a particular spatial unit as the average absolute correlation $(1-f_{COR})$ of nondominated solutions that incorporate the unit.

To visualize the importance of spatial information, we generated a series of heatmaps (Figure 3.5). In Figures 3.5a, 3.5c, and 3.5e we show regional importance values of filter methods for each $R \in \{1, ..., 20\}$, with the relevant value of R annotated in the upper left corner of each box. Note that in lasso- and GP-based approaches, some variables are unused (white), while ridge cannot perform variable selection and uses all. Figures 3.5b and 3.5d plot WR and WL for $R \in \{1, 2, 3, 4\}$. Finally, Figures 3.5e and 3.5f plot the importance of spatial information in the GP sense, for FGP and GPESA, respectively. Overall, this visualization indicates an

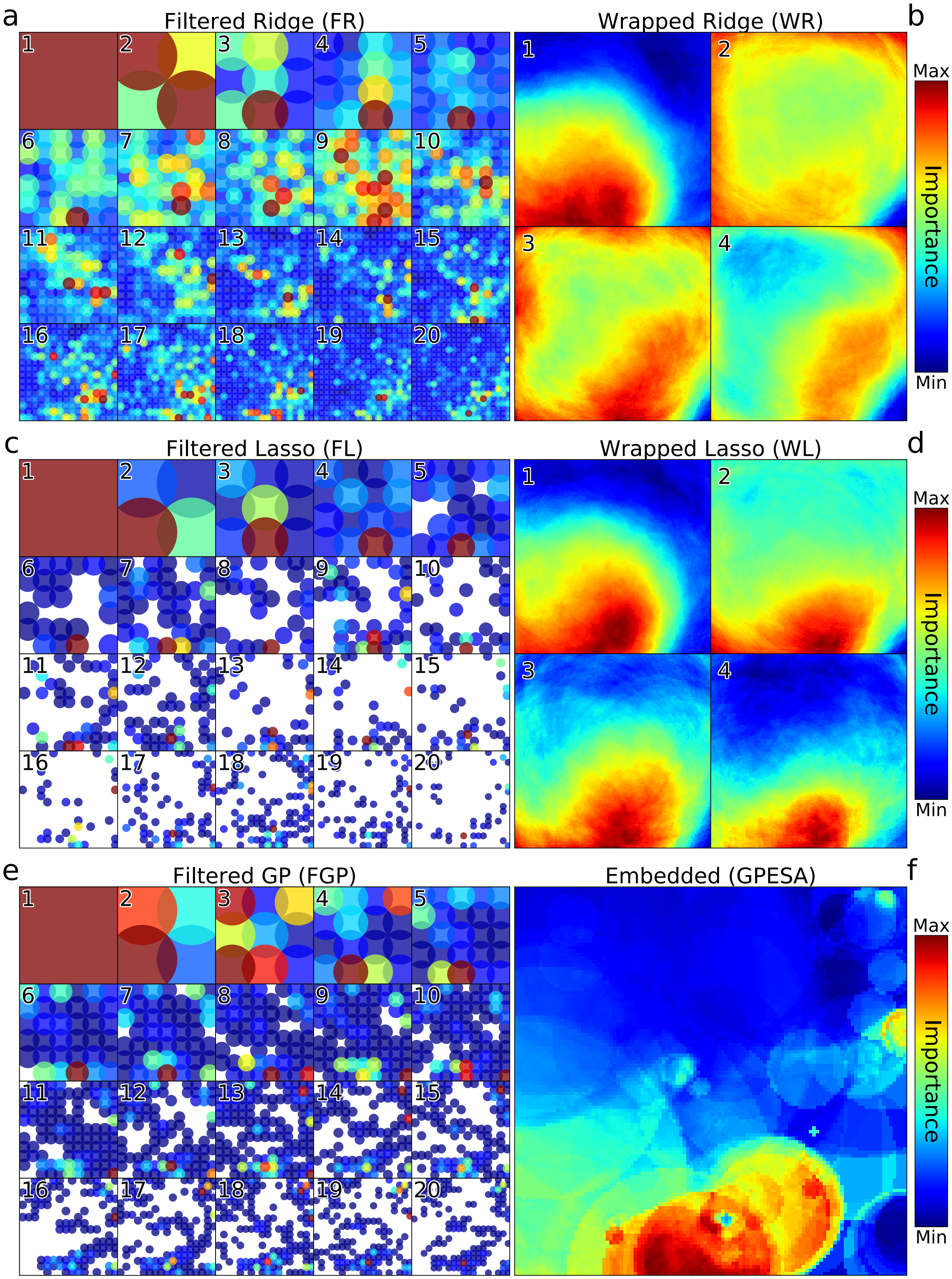


Figure 3.5: *(Continued on the following page.)*

Figure 3.5: *Importance (defined in Section 3.4) of spatial units. For filters a.) FR, c.) FL, and e.) FGP, importance is displayed at each resolution $R \in \{1, 2, \ldots 20\}$ and each individual filter subplot is annotated with the corresponding R. For wrappers b.) WR and d.) WL, $R \in \{1, 2, 3, 4\}$. Finally, f.) GPESA, which has no R parameter. White areas indicate spatial units unused in feature construction across all three exploratory variables.*

agreement among all methods on the relatively higher importance of information in the lower center/right region of the image.

Nevertheless, the placement of the important regions is particularly counter-intuitive. A human investigator might speculate that it would be optimal to monitor the regions of space with the most SWE, i.e. at the highest elevations. However as we can see from comparison to Figure 2.2, models are able to reasonably induce total areal SWE based solely on changes on the fringe of mountainous regions. Why these regions are so predictive would be interesting for domain scientists to consider, to potentially drive new physical insights or monitoring strategies. It appears as though the models are performing 'edge detection' by relating regional SWE to the melt rate and/or the eventually complete disappearance of snow in these low elevation regions.

It is more difficult to ascertain the importance in SL (Figure 3.6b) which combines seemingly random pixels, inconsistently across model years. This is due in part to the geometry of SL's constraint which tends to only include one of many correlated variables into it's sparse regression model. While it is preciously this constraint and the resulting sparsity that makes SL so good at generalization, the multiplicity of SL models and the pixels they incorporate further supports the idea of performing at least some spatial aggregation.

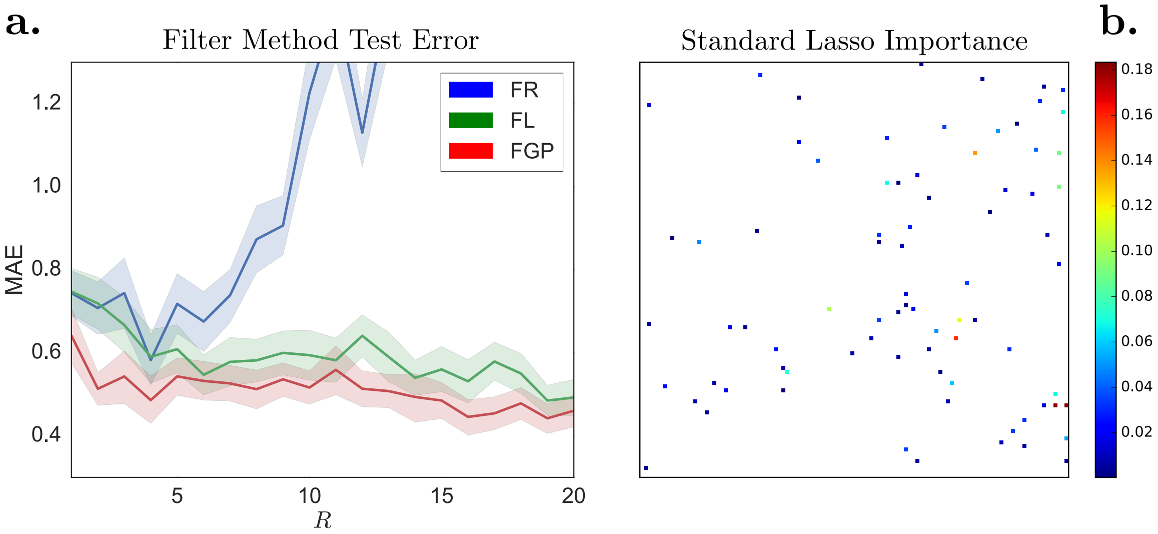


Figure 3.6: *Supplemental results. a.) Mean absolute test error for the filters by resolution R. b.) importance (defined in Section 3.4) of spatial units used by Standard Lasso (SL).*

> *Such is the supreme folly of man that he labors so as to labor no more.*
>
> Leonardo di ser Piero da Vinci

4 Summary and Conclusions

In this work we developed a novel method to address the problem of modeling a regional response with high resolution satellite imagery. We moved away from uniform up-sampling aggregations towards more flexible and interesting aggregation operations predicated on their subsequent use as features of a regional model. Our proposed technique, GPESA, is general and intended to apply to a variety of modeling problems on spatially organized data. But as an application example, and as a setting in which to evaluate our techniques, we considered the problem of estimating snow water equivalent in high mountain Asia using satellite imagery. Our results showed that using GP to evolve spatial aggregations outperforms lasso, the state-of-the-art method for directly incorporating individual spatial units into a sparse linear model.

4.1 Future Work

In future work we plan to explore more flexible spatial and temporal aggregations for more predictive modeling in real earth science applications. To do so we will concentrate on four connected avenues of research: (1) relax constraints on the **geometries in space** which highlight spatial units to be sampled, (2) incorporate varying **geometries in time**, (3) design more sophisticated **statistics** to aggregate elements of the resulting samples, and (4) improve the identification of useful aggregations by considering their **semantics**.

Geometries in space.

Throughout the paper, we use circles to select spatial units for aggregation because circles only have two adjustable parameters: a centerpoint and radius. We could proceed by including an additional focal point and use the resulting ellipse to select spatial units. If circles in fact support more effective aggregations in certain regions, they may still be formed by placing both foci are at the same point (the center). This additional flexibility and may accommodate superior model performance, but in general the search space increases exponentially in the number of additional modifiable parameters.

Even if an ellipse proves more effective than a circle, increasing complexity in this manner inevitably leads to many parameters governing smaller aspects of a complex shape (e.g. a convex hull, perimeter, or the union of many circles). In any case, we will eventually hit a complexity ceiling as the number of governing parameters increases. We could define complex geometries without any additional parameters by using threshold logic at each spatial unit: "if a variable is greater than c, include the unit in the aggregation sample." Variance or entropy may prove useful in defining threshold-based geometries, especially given that the most important regions, as indicated by Figure 3.5, were in areas with higher relative variance across time (the snow completely melted here and SWE went to 0). Metadata like elevation, aspect and slope could be used in a similar fashion although preliminary experiments suggested groupings based on distance in geographical space were significantly more effective than in elevation, aspect or slope. The issue with threshold logic is that it might produce erratic, sparse patterns that may not generalize well to unseen data. For this reason, the simplicity of circular spatial aggregations was potentially beneficial in our experiments, and it remains to be seen if perusing more complex geometries begets overfitting.

Alternatively, Compositional Pattern Producing Networks (CPPNs, [111, 112]) have demonstrated a fantastic ability to create complex geometries characterized by symmetry, repetition, and interesting variation using convolutions of a small set of simple functions (Figure 4.1). Replacing GPESA terminals with CPPNs could allow for more interesting geometries with a modest number of modifiable parameters. Moreover, the continuous and symmetric properties of CPPNs could conceivably protect against overfitting the training data.

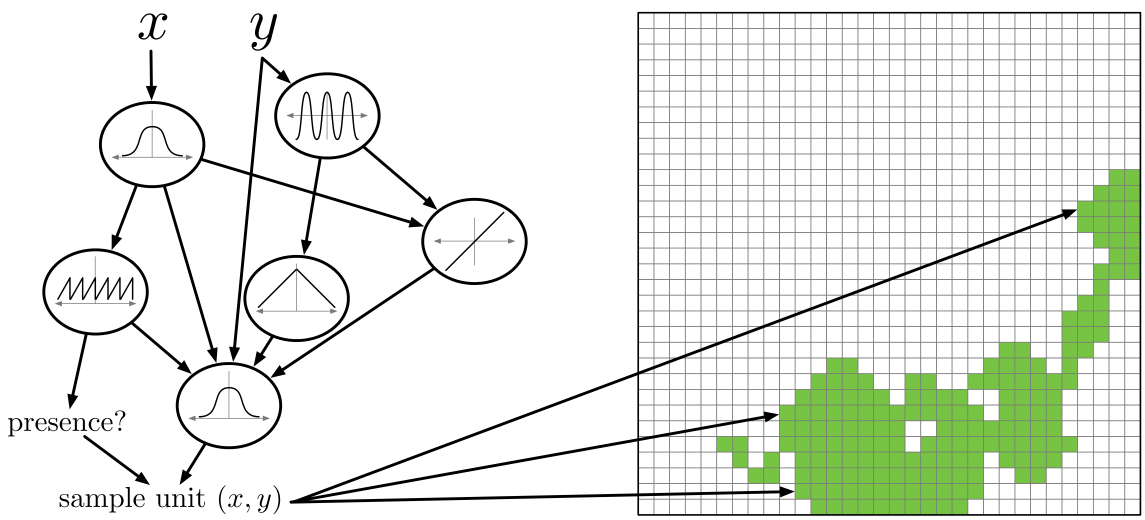


Figure 4.1: *A CPPN is iteratively queried for each location on the map and produces output values as a function of its coordinates. These outputs determine the presence of units in the sample of a spatially aggregated feature.*

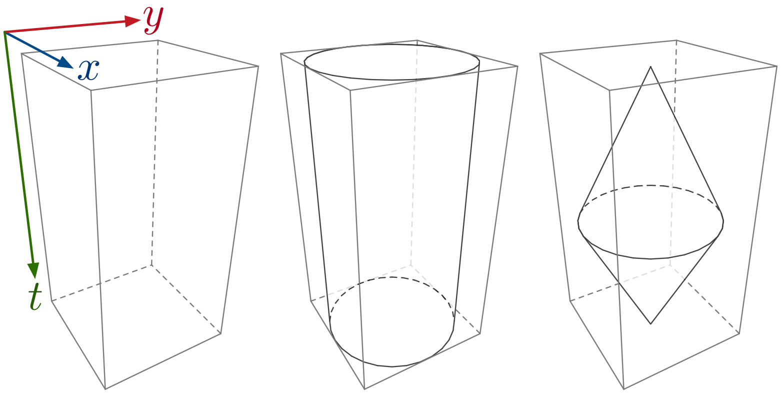
The CPPNs could simply take the coordinates of a particular spatial unit and output whether or not to include it in a sample to be aggregated (as in Figure 4.1). However, the most important regions indicated by Figure 3.5 reside in a relatively small subset of possible elevations as well as geographical coordinates. Latitude and longitude might have ultimately worked better because regions with high importance were more restricted in geographical space than in elevation. Perhaps in some combination these three inputs (elevation, latitude and longitude) could collectively form superior aggregation samples along Earth's surface.

Co-evolution. In order to increase the efficiency of sampling geometries we may have to adjust the GPESA algorithm. A natural approach is the co-evolution [113] of sampling geometries alongside a population GPESA trees with empty terminals. In the geometry population, the algorithms could adjust the parameters of ellipses or alter the network structure of CPPNs. In this manner, individual models would gain exposure to a range of samples across geographical space and vice versa, which could provide a refinement in both selection geometries as well as the way in which they are combined into a predictive model.

Geometries in time.

So far, we only explored spatial aggregations but we could adjust aggregation in time as well. The first step is to incorporate 'day of year' as a fourth explanatory variable. We expect there are at least coarse grain temporal autocorrelations in snow dynamics since overall it melts away over the course of the season. Thus, we could predict SWE for a particular day based on data from certain ranges of time. If the more recent months of a variable exhibit a stronger association with the response, then limiting spatial aggregations to a window in time could increase predictive accuracy. However, we must be careful not to surrender useful observations in a high dimensional problem.

A more economical approach is to simply vary the selected spatial units across time. Currently, aggregation occurs in a circle throughout a stack of images in time. The semantics of a GPESA terminal are the aggregations of a particular circle across each of the 1720 training day images. So the sampling geometry is actually a cylinder in spacetime (Figure 4.2). Instead of a cylinder, we could have a more curvilinear shape dictating which spatial units to aggregate for a given time of year. With relatively few parameters we can use a bicone (two cones placed base-to-base). The widest portion of the bicone will correspond to the current day we are estimating and as we move away from the current day the selected region will shrink to a single unit (Figure 4.2). This structure will repeat in time for each snow year in our training set, eight



time invariant
(GPESA terminal)

time variant

Figure 4.2: *Sampling geometries in space (x, y) and across time (t) for a single training year. Current GPESA sampling geometries are time-invariant; in future work we will investigate relaxing this constraint.*

bicones placed apex-to-apex. For this shape to be more effective than a cylinder, days of the year in the immediate future or past must have larger spatial autocorrelations associated in some way with the current day response. But perhaps the opposite is true, in which case we can just shift our cones up to touch apex-to-apex on the current day (rather than base-to-base).

If we cut off the upper part of a larger cone with a plane (the resulting shape is called a frustum) then we can expand the apex from a single unit to a circle of any size. The parameters dictating this conical frustum are the shared centerpoint and two radiuses of each base. So there is just one extra parameter from a cylinder, three in total. Next, we can allow the centerpoint of each base to shift in space as well (another two additional parameters to specify the second centerpoint, five in total). If there are asymmetrical autocorrelations in past days versus the future days, base centerpoints and radiuses could also change according to a function of the current day (one additional parameter, six in total)...

Clearly this hand-designed path through increasing parameters will again lead us exponentially fast into a complexity ceiling. Yet by simply including time as an additional CPPN input we could produce a much wider assortment of interesting 3D geometries (in spacetime), as evolved and printed in [114].

Statistics.

We fixed the statistic responsible for aggregating spatial units selected within a circular region. We chose the arithmetic mean but this data reduction can be done in a variety of more sophisticated ways which may potentiate more accurate modeling. For example, we could allow for the choice between mean, median, minimum and maximum; but again we run into the problem of additional parameters. Indeed preliminary experiments suggested that restricting the statistic to the mean was superior to a selection between these four summary statistics. Although it would be interesting to try entropy or variance in addition to mean.

Another way to extend the flexibility of aggregation statistics is to use a weighted average. This could be accomplished through CPPNs with an additional output specifying the weight. Or, in accordance with Tobler's first law of geography, we could replace circles with Gaussian distributions, their standard deviations controlling a distance-decayed weighting of neighboring units. The resulting predictive model may be interpreted as a mixture of Gaussians.

In a way this idea is reminiscent of geographically weighted regression (GWR, [115]), which calculates regression coefficients at every point in geographical space based on a distance-decayed weighted sample of its neighbors. However, GWR is primarily an exploratory tool for investigating non-stationary on a map, whereas our goal is regional model accuracy. In a similar fashion, we could incorporate explanatory variables across different images into locally weighted regressions. Such an approach could prove beneficial if there are generally useful areas of space with higher signal-to-noise ratios across multiple variables. But if generally useful regions do exist across or within variables, then we might be able to isolate them by modeling their importance directly and incorporating the resulting distribution through an a priori change of basis, similar to [116].

Semantics.

Standard tree-based GP searches the space of programs using crossover and mutation operators that replace or modify subtrees. These operators are guaranteed to produce syntactically correct offspring, however their actual effects on the behavior of the program are unpredictable because the genotype-phenotype mapping is characterized by low locality: even a minimal change at the syntax level may diametrically alter program semantics. With nontrivial fitness landscapes, such large phenotypic changes are problematic because the probability of a mutation being beneficial is inversely proportional to its magnitude [3]. Recently, many semantically-aware search operators have been proposed [117–122], and have proved to be effective on a number of symbolic regression problems.

We could simply replace the mutation and crossover operators in GPESA with one of the semantic procedures referenced above. Additionally, information about the semantics of spatial aggregations could be exploited to maintain their semantic diversity across the population [123] or for intelligent initializations [124]. However we could also design new semantic operators that guide how aggregations are formed and modified. Using semantic forward or backwards propagation [125,126], we could deduce the necessary semantics of particular aggregation terminals in GPESA. In other words, the exact aggregation values over time that would result in zero training error when passed up the tree. The optimization problem then becomes a matter of finding aggregation terminals that yield similar semantics or simple operations that exploit given aggregations. We can then archive previously explored aggregations in order to reuse those that best approximate the exact target semantics. This archive could remove the need for GPESA's embedded hill climber along with its subsequent distance calculations and tree evaluations.

GPESA 2.0

In summary, we see semantic operators, CPPNs, and co-evolutionary approaches as the most promising avenues for future research towards producing more cogent aggregations. However, while future research is focused primarily on the aggregation terminals of GPESA, there are many other general algorithmic improvements from the GP literature that could be applied to the other end of the trees or their population. Given the separate success of both lasso and GPESA, a hybrid approach is particularly appealing. Perhaps the easiest adjustment we could make is to use lasso on all the unique subtrees within solutions on the final Pareto front [127]. Although we could also apply a similar idea every generation, guiding selection towards individuals that contribute to a collective lasso solution [128,129].

Regardless of which enhancements are made, it is crucial that we apply GPESA to other datasets in order to determine if its success was merely an artifact of a single environment. To this end, we plan to pursue data for SWE in the larger High-mountain

Asia region (extending eastwards) as well as Normalized Difference Vegetation Index (NDVI) in the Amazon rainforest. Additionally, synthetic datasets should be created to test the sensitivity of GPESA and to get a better understanding of the kinds of spatiotemporal autocorrelations it can and cannot exploit. Ultimately, by refining the generalization performance of GPESA and dissecting its complex solutions, we should gain a deeper understanding of the natural phenomena we model.

Bibliography

[1] Charles Darwin. *The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life.* 1859.

[2] Charles Darwin and Alfred Wallace. On the tendency of species to form varieties; and on the perpetuation of varieties and species by natural means of selection. *Journal of the proceedings of the Linnean Society of London. Zoology*, 3(9):45–62, 1858.

[3] Sir Fisher, Ronald A. *The genetical theory of natural selection.* Oxford University Press, Oxford, 1930.

[4] Francis Galton. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263, 1886.

[5] Leo Breiman et al. Statistical modeling: the two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199–231, 2001.

[6] Charles Percy Snow. *The two cultures.* Cambridge University Press, 1959.

[7] Jacob Cohen. The earth is round (p < .05). *American psychologist*, 49(12):997–1003, 1994.

[8] Paul A Murtaugh. In defense of p values. *Ecology*, 95(3):611–617, 2014.

[9] Aaron M Ellison, Nicholas J Gotelli, Brian D Inouye, and Donald R Strong. P values, hypothesis testing, and model selection: it's déjà vu all over again. *Ecology*, 95(3):609–610, 2014.

[10] Regina Nuzzo et al. Statistical errors. *Nature*, 506(7487):150–152, 2014.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer, Berlin: Springer series in statistics, 2011.

[12] Andy Clark. *Being there: Putting brain, body, and world together again.* 1998.

[13] Rolf Pfeifer and Josh Bongard. *How the body shapes the way we think: a new view of intelligence.* MIT press, 2006.

[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[15] J Doyne Farmer, Norman H Packard, and Alan S Perelson. The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, 22(1):187–204, 1986.

[16] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.

[17] Antonio Mucherino and Onur Seref. Monkey search: a novel metaheuristic search for global optimization. In *Data Mining, Systems Analysis and Optimization in Biomedicine*, volume 953, pages 162–173. AIP Publishing, 2007.

[18] A Kaveh and N Farhoudi. A new optimization method: Dolphin echolocation. *Advances in Engineering Software*, 59:53–70, 2013.

[19] Shu-Chuan Chu, Pei-Wei Tsai, and Jeng-Shyang Pan. Cat swarm optimization. In *Pacific Rim International Conference on Artificial Intelligence*, pages 854–858. Springer, 2006.

[20] Xin-She Yang. A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pages 65–74. Springer, 2010.

[21] Xin-She Yang and Suash Deb. Eagle strategy using lévy walk and firefly algorithms for stochastic optimization. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pages 101–111. Springer, 2010.

[22] Xin-She Yang and Suash Deb. Cuckoo search via lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 210–214. IEEE, 2009.

[23] Hugo Hernández and Christian Blum. Distributed graph coloring: an approach based on the calling behavior of japanese tree frogs. *Swarm Intelligence*, 6(2):117–150, 2012.

[24] Mehdi Neshat, Ghodrat Sepidnam, Mehdi Sargolzaei, and Adel Najaran Toosi. Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. *Artificial Intelligence Review*, 42(4):965–997, 2014.

[25] Amir Hossein Gandomi and Amir Hossein Alavi. Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12):4831–4845, 2012.

[26] Kevin M Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE control systems*, 22(3):52–67, 2002.

[27] Ali Reza Mehrabian and Caro Lucas. A novel numerical optimization algorithm inspired from weed colonization. *Ecological informatics*, 1(4):355–366, 2006.

[28] Xin-She Yang, Mehmet Karamanoglu, and Xingshi He. Flower pollination algorithm: a novel approach for multiobjective optimization. *Engineering Optimization*, 46(9):1222–1237, 2014.

[29] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39(3):459–471, 2007.

[30] Wen-Tsao Pan. A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, 26:69–74, 2012.

[31] Xin-She Yang. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2):78–84, 2010.

[32] KN Krishnanand and Debasish Ghose. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm intelligence*, 3(2):87–124, 2009.

[33] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

[34] Timothy C Havens, Christopher J Spain, Nathan G Salmon, and James M Keller. Roach infestation optimization. In *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, pages 1–7. IEEE, 2008.

[35] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems.* Number 1. Oxford university press, 1999.

[36] Kenneth Sörensen. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, 2015.

[37] Gerd B Muller and Gunter P Wagner. Novelty in evolution: restructuring the concept. *Annual Review of Ecology and Systematics*, 22:229–256, 1991.

[38] Armin P Moczek, Sonia Sultan, Susan Foster, Cris Ledón-Rettig, Ian Dworkin, H Fred Nijhout, Ehab Abouheif, and David W Pfennig. The role of developmental plasticity in evolutionary innovation. *Proceedings of the Royal Society of London B: Biological Sciences*, 278(1719):2705–2713, 2011.

[39] Ernst Mayr. Cause and effect in biology. *Science*, 134(3489):1501–1506, 1961.

[40] Kevin N Laland, Kim Sterelny, John Odling-Smee, William Hoppitt, and Tobias Uller. Cause and effect in biology revisited: is mayr's proximate-ultimate dichotomy still useful? *Science*, 334(6062):1512–1516, 2011.

[41] Niko Tinbergen. On aims and methods of ethology. *Zeitschrift für Tierpsychologie*, 20(4):410–433, 1963.

[42] Patrick Bateson and Kevin N Laland. Tinbergen's four questions: an appreciation and an update. *Trends in ecology & evolution*, 28(12):712–718, 2013.

[43] Peter L Jakab. *Visions of a flying machine: The Wright brothers and the process of invention.* Smithsonian Institution, 2014.

[44] Gary William Flake. *The computational beauty of nature: Computer explorations of fractals, chaos, complex systems, and adaptation.* MIT press, 1998.

[45] Dario Floreano and Claudio Mattiussi. *Bio-inspired artificial intelligence: theories, methods, and technologies.* 2008.

[46] Martijn A Huynen, Peter F Stadler, and Walter Fontana. Smoothness within ruggedness: the role of neutrality in adaptation. *Proceedings of the National Academy of Sciences*, 93(1):397–401, 1996.

[47] Richard Dawkins. *The blind watchmaker: why the evidence of evolution reveals a universe without design.* WW Norton & Company, 1986.

[48] Geoffrey E Hinton and Steven J Nowlan. How learning can guide evolution. *Complex systems*, 1(3):495–502, 1987.

[49] Kevin N Laland, Tobias Uller, Marcus W Feldman, Kim Sterelny, Gerd B Müller, Armin Moczek, Eva Jablonka, and John Odling-Smee. The extended evolutionary synthesis: its structure, assumptions and predictions. In *Proc. R. Soc. B*, volume 282, page 20151019. The Royal Society, 2015.

[50] Richard A Watson and Eörs Szathmáry. How can evolution learn? *Trends in ecology & evolution*, 31(2):147–157, 2016.

[51] Conrad H Waddington. Canalization of development and the inheritance of acquired characters. *Nature*, 150(3811):563–565, 1942.

[52] Richard C Lewontin. *The triple helix: Gene, organism, and environment.* Harvard University Press, 2001.

[53] George Dyson. *Turing's cathedral: the origins of the digital universe.* Vintage, 2012.

[54] Nils Aall Barricelli. Numerical testing of evolution theories. *Acta Biotheoretica*, 16(1-2):69–98, 1962.

[55] Josh C Bongard. Evolutionary robotics. *Communications of the ACM*, 56(8):74–83, 2013.

[56] Ingo Rechenberg. Evolutionsstrategien. In *Simulationsmethoden in der Medizin und Biologie*, pages 83–114. Springer, 1978.

[57] Jason D Lohn, Gregory S Hornby, and Derek S Linden. An evolved antenna for deployment on nasa's space technology 5 mission. In *Genetic Programming Theory and Practice II*, pages 301–315. Springer, 2005.

[58] Chen Wang, Shuangcheng Yu, Wei Chen, and Cheng Sun. Highly efficient light-trapping structure design inspired by natural evolution. *Scientific reports*, 3, 2013.

[59] Jérémy Besnard, Gian Filippo Ruda, Vincent Setola, Keren Abecassis, Ramona M Rodriguiz, Xi-Ping Huang, Suzanne Norval, Maria F Sassano, Antony I Shin, Lauren A Webster, et al. Automated design of ligands to polypharmacological profiles. *Nature*, 492(7428):215–220, 2012.

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

[61] 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 on Genetic and Evolutionary Computation Conference*, pages 1279–1286. ACM, 2015.

[62] Agoston E Eiben and James E Smith. *Introduction to evolutionary computing*, volume 53. Springer, 2003.

[63] Kenneth A De Jong. *Evolutionary computation: a unified approach*. MIT press, 2006.

[64] David E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Number 2. Addison-Wesley, Reading, MA, 1989.

[65] Agoston E Eiben and Jim Smith. From evolutionary computation to the evolution of things. *Nature*, 521(7553):476–482, 2015.

[66] Daniel C Dennett. Darwin's dangerous idea. *The Sciences*, 35(3):34–40, 1995.

[67] David B Fogel. An introduction to simulated evolutionary optimization. *IEEE transactions on neural networks*, 5(1):3–14, 1994.

[68] George Lakoff and Rafael Núñez. Where mathematics comes from. Santa Fe Institute, 2003.

[69] John R. Koza. *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, MA, USA, 1992.

[70] Donald E Knuth. *Fundamental Algorithms: The art of computer programming*. 1973.

[71] Charles E McCulloch and John M Neuhaus. *Generalized linear mixed models*. Wiley Online Library, 2001.

[72] John R Koza. Hierarchical genetic algorithms operating on populations of computer programs. In *IJCAI*, pages 768–774. Citeseer, 1989.

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

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

[75] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[76] David J Montana. Strongly typed genetic programming. *Evolutionary computation*, 3(2):199–230, 1995.

[77] Nguyen Quang Uy, Nguyen Xuan Hoai, Michael O'Neill, R. I. Mckay, and Edgar Galván-López. Semantically-based crossover in genetic programming: Application to Real-valued Symbolic Regression. *Genetic Programming and Evolvable Machines*, 12(2):91–119, 2011.

[78] Riccardo Poli, William B Langdon, Nicholas F McPhee, and John R Koza. *A field guide to genetic programming.* Lulu. com, 2008.

[79] V. Pareto. *Manual of political economy.* Scholars Book Shelf, 1906.

[80] JG Rees, AD Gibson, Matthew Harrison, Andrew Hughes, and JC Walsby. Regional modelling of geohazard change. *Geological Society, London, Engineering Geology Special Publications*, 22(1):49–63, 2009.

[81] Stan Openshaw and Peter J Taylor. A million or so correlation coefficients: three experiments on the modifiable areal unit problem. *Statistical applications in the spatial sciences*, 21:127–144, 1979.

[82] Carol A Gotway and Linda J Young. Combining incompatible spatial data. *Journal of the American Statistical Association*, 97(458):632–648, 2002.

[83] Justin S Mankin, Daniel Viviroli, Deepti Singh, Arjen Y Hoekstra, and Noah S Diffenbaugh. The potential for snow to supply human water demand in the present and future. *Environmental Research Letters*, 10(11):114016, 2015.

[84] Tim P Barnett, Jennifer C Adam, and Dennis P Lettenmaier. Potential impacts of a warming climate on water availability in snow-dominated regions. *Nature*, 438(7066):303–309, 2005.

[85] Tim P Barnett, David W Pierce, Hugo G Hidalgo, Celine Bonfils, Benjamin D Santer, Tapash Das, Govindasamy Bala, Andrew W Wood, Toru Nozawa, Arthur A Mirin, et al. Human-induced changes in the hydrology of the western united states. *science*, 319(5866):1080–1083, 2008.

[86] J. Dong, J. P. Walker, and P. R. Houser. Factors affecting remotely sensed snow water equivalent uncertainty. *Remote Sensing of Environment*, 97(1):68–82, 2005.

[87] Carl-Friedrich Schleussner, Jonathan F. Donges, Reik V. Donner, and Hans Joachim Schellnhuber. Armed-conflict risks enhanced by climate-related disasters in ethnically fractionalized countries. *Proceedings of the National Academy of Sciences*, 2016.

[88] Ulf Büntgen, Willy Tegel, Kurt Nicolussi, Michael McCormick, David Frank, Valerie Trouet, Jed O Kaplan, Franz Herzig, Karl-Uwe Heussner, Heinz Wanner, et al. 2500 years of european climate variability and human susceptibility. *Science*, 331(6017):578–582, 2011.

[89] Heidi M Cullen, S Hemming, G Hemming, FH Brown, T Guilderson, F Sirocko, et al. Climate change and the collapse of the akkadian empire: Evidence from the deep sea. *Geology*, 28(4):379–382, 2000.

[90] Peter B DeMenocal. Cultural responses to climate change during the late holocene. *Science (New York, NY)*, 292(5517):667–673, 2001.

[91] Gerald H Haug, Detlef Günther, Larry C Peterson, Daniel M Sigman, Konrad A Hughen, and Beat Aeschlimann. Climate and the collapse of maya civilization. *Science*, 299(5613):1731–1735, 2003.

[92] Douglas J Kennett, Sebastian FM Breitenbach, Valorie V Aquino, Yemane Asmerom, Jaime Awe, James UL Baldini, Patrick Bartlein, Brendan J Culleton, Claire Ebert, Christopher Jazwa, et al. Development and disintegration of maya political systems in response to climate change. *Science*, 338(6108):788–791, 2012.

[93] Joseph Tainter. *The collapse of complex societies*. Cambridge University Press, 1990.

[94] David Buckingham, Christian Skalka, and Joshua Bongard. Inductive learning of snowpack distribution models for improved estimation of areal snow water equivalent. *Journal of Hydrology*, 524:311–325, 2015.

[95] PCD Milly, B Julio, F Malin, M Robert, W Zbigniew, P Dennis, and J Ronald. Stationarity is dead. *Ground Water News & Views*, 4(1):6–8, 2007.

[96] Claire L Parkinson. Aqua: An earth-observing satellite mission to examine water and other climate variables. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(2):173–183, 2003.

[97] M Tedesco, REJ Kelly, JL Foster, and ATC Chang. Amsr-e/aqua daily l3 global snow water equivalent ease-grids v002. *National Snow and Ice Data Center: Boulder, CO*, 2004.

[98] Marco Tedesco and Parag S Narvekar. Assessment of the nasa amsr-e swe product. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 3(1):141–159, 2010.

[99] T. H. Painter, K. Rittger, C. McKenzie, P. Slaughter, R. E. Davis, and J. Dozier. Retrieval of subpixel snow-covered area, grain size, and albedo from MODIS. *Remote Sensing of Environment*, 113:868–879, 2009.

[100] J. Dozier, T. H. Painter, K. Rittger, and J. Frew. Time-space continuity of daily maps of fractional snow cover and albedo from MODIS. *Advances in Water Resources*, 31(11):1515–1526, 2008.

[101] Dorothy K Hall and George A Riggs. Accuracy assessment of the modis snow products. *Hydrological Processes*, 21(12):1534–1547, 2007.

[102] J Martinec and A Rango. Areal distribution of snow water equivalent evaluated by snow cover monitoring. *Water Resour. Res*, 17(5):1480–1488, 1981.

[103] Arthur E Hoerl and Robert W Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

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

[105] Michael D Schmidt, Ravishankar R Vallabhajosyula, Jerry W Jenkins, Jonathan E Hood, Abhishek S Soni, John P Wikswo, and Hod Lipson. Automated refinement and inference of analytical models for metabolic networks. *Physical biology*, 8(5):055011, 2011.

[106] Michael Schmidt and Hod Lipson. Age-fitness pareto optimization. In *Genetic Programming Theory and Practice VIII*, volume 8 of *Genetic and Evolutionary Computation*, pages 129–146. Springer, 2011.

[107] Gregory S. Hornby. ALPS: the age-layered population structure for reducing the problem of premature convergence. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 815–822. ACM, 2006.

[108] 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.

[109] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, pages 234–240, 1970.

[110] Myles Hollander, Douglas A Wolfe, and Eric Chicken. *Nonparametric statistical methods*. John Wiley & Sons, 2013.

[111] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.

[112] Jimmy Secretan, Nicholas Beato, David B D Ambrosio, Adelein Rodriguez, Adam Campbell, and Kenneth O Stanley. Picbreeder: evolving pictures collaboratively online. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1759–1768. ACM, 2008.

[113] Mitchell A Potter and Kenneth A De Jong. Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evolutionary computation*, 8(1):1–29, 2000.

[114] Jeff Clune and Hod Lipson. Evolving three-dimensional objects with a generative encoding inspired by developmental biology.

[115] Chris Brunsdon, A Stewart Fotheringham, and Martin Charlton. Geographically weighted regression: a method for exploring spatial nonstationarity. *Encyclopedia of Geographic Information Science*, page 558, 2008.

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

[117] Lawrence Beadle and Colin G. Johnson. Semantically driven crossover in genetic programming. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1-6, 2008, Hong Kong, China*, pages 111–116. IEEE, 2008.

[118] Josh C. Bongard. A probabilistic functional crossover operator for genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 925–932. ACM, 2010.

[119] Alberto Moraglio, Krzysztof Krawiec, and Colin G. Johnson. Geometric semantic genetic programming. In *Parallel Problem Solving from Nature - PPSN XII*, volume 7491 of *Lecture Notes in Computer Science*, pages 21–31. Springer Berlin Heidelberg, 2012.

[120] Krzysztof Krawiec and Tomasz Pawlak. Locally geometric semantic crossover: a study on the roles of semantics and homology in recombination operators. *Genetic Programming and Evolvable Machines*, 14(1):31–63, 2013.

[121] Marcin Szubert, Anuradha Kodali, Sangram Ganguly, Kamalika Das, and Joshua Bongard. Reducing antagonism between behavioral diversity and fitness in semantic genetic programming. In *GECCO '16: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*, pages 797–804, Denver, USA, 2016. ACM.

[122] Krzysztof Krawiec. *Behavioral program synthesis with genetic programming*, volume 618 of *Studies in Computational Intelligence*. Springer, 2016.

[123] David Jackson. Promoting phenotypic diversity in genetic programming. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature, PPSN XI*, volume 6239 of *Lecture Notes in Computer Science*, pages 472–481. Springer Berlin Heidelberg, 2010.

[124] Lawrence Beadle and Colin G. Johnson. Semantic analysis of program initialisation in genetic programming. *Genetic Programming and Evolvable Machines*, 10(3):307–337, 2009.

[125] Tomasz P Pawlak, Bartosz Wieloch, and Krzysztof Krawiec. Semantic backpropagation for designing search operators in genetic programming. *IEEE Transactions on Evolutionary Computation*, 19(3):326–340, 2015.

[126] Marcin Szubert, Anuradha Kodali, Sangram Ganguly, Kamalika Das, and Joshua C Bongard. Semantic forward propagation for symbolic regression. In *Parallel Problem Solving from Nature - PPSN XIV*, Lecture Notes in Computer Science, 2016.

[127] 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.

[128] Vinícius Veloso De Melo. Kaizen programming. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 895–902. ACM, 2014.

[129] 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.