MRS Communications (2019), *9*, 793–805
© Materials Research Society, 2019
doi:10.1557/mrc.2019.85

MRS Communications

Artificial Intelligence Prospective

Symbolic regression in materials science

Yiqun Wang†, Nicholas Wagner†, and **James M. Rondinelli**, Department of Materials Science and Engineering, Northwestern University, Evanston, IL 60208, USA

Address all correspondence to James M. Rondinelli at **jrondinelli@northwestern.edu**

(Received 13 January 2019; accepted 11 June 2019)

Abstract

The authors showcase the potential of symbolic regression as an analytic method for use in materials research. First, the authors briefly describe the current state-of-the-art method, genetic programming-based symbolic regression (GPSR), and recent advances in symbolic regression techniques. Next, the authors discuss industrial applications of symbolic regression and its potential applications in materials science. The authors then present two GPSR use-cases: formulating a transformation kinetics law and showing the learning scheme discovers the well-known Johnson–Mehl–Avrami–Kolmogorov form, and learning the Landau free energy functional form for the displacive tilt transition in perovskite $LaNiO_3$. Finally, the authors propose that symbolic regression techniques should be considered by materials scientists as an alternative to other machine learning-based regression models for learning from data.

Motivation

Era of big data in materials science

Modern scientists perpetuate the scientific process embodied by the works of Tyco Brahe, Johannes Kepler, and Isaac Newton in the heliocentric revolution. Brahe was the observationalist. He took extensive, precise measurements of the position of planets over time. Kepler was the phenomenologist. From Brahe's measurements, he derived concise analytic expressions that describe the motion of the solar system in a succinct manner. Last, Newton was the theorist. He realized that the mechanism behind the apple falling from the tree is the same as that underlying planets traveling around the sun, which could be formulated into a universal law (Newtonian gravitational law). All three scientific modalities are vital in making scientific discoveries: data acquisition (Brahe), data analysis (Kepler), and derivation from first principles (Newton).

With recent advances in computer science, theoretical modeling, and experimental instrumentation, materials scientists have, in many ways, created a "mechanical Brahe" and marched into a new era of big data. Datasets of materials information, obtained from advanced characterization techniques,[1–3] combinatorial experiments,[4–6] high-throughput first-principle simulations,[7,8] literature mining,[9,10] and other techniques, are created at a faster rate every day with less and less human labor. All of this data enables new opportunities to construct novel laws of phenomenological behavior for systems that previously lacked them.

Inspired by the Materials Genome Initiative,[11] the materials community is working collaboratively towards making digital materials data accessible to others. Multiple materials databases, such as Materials Project,[12] OQMD,[13] AFLOWLIB,[14] OMDB,[15] AiiDA,[16] Citrination and NOMAD, provide public access to millions of materials data points. Accessibility to an immense amount of materials data paves way for the next step of "automating Kepler" in the discovery of governing laws in materials processing–structure–properties–performance relationships, which could advance materials discovery, development, and technology innovation.

Since one of the fundamental research objectives of materials science and engineering is to deliver new materials with optimal performance under specified constraints, it is essential to understand how and which features govern the functionality. In other words, which degrees of freedom (or parameters) and their corresponding intrinsic relationships (or dependencies) to the material properties should be optimized. However, the multiscale nature of materials science,[5] e.g., from atomic-scale crystal structure to complex mesoscale domain structures and bulk mechanical properties or from femtosecond laser probes to hour-long recrystallization reactions, makes it particularly challenging to study many hierarchical relationships of different materials families. Given such a high-dimensional parameter space (e.g., chemical composition, crystal structure, and external conditions), materials scientists often explore a finite subspace of all the factors that govern materials properties and performance. In addition, the available data are typically sparsely distributed. Although access to a large materials database relieves, in part, the limited-data problem, there is an urgent need for a robust data-processing protocol to help discern governing laws in materials science and to deliver designer materials and synthesis/processing procedures.

† These authors contributed equally to this work.

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

Alternative to machine learning methods

Much of the burgeoning field of materials informatics focuses on the aforementioned challenges. Machine learning (ML) models are currently the tools of choice for uncovering these physical laws. Although they have shown some promising performance in predicting materials properties,[17] typical parameterized ML models are not conducive to the next stage of generalizing across domains—the ultimate goal of "automating Newton."

It is important to note that Newton's challenge was somewhat made easier, because Kepler's laws were parsimonious yet predictive. In a modern context, ML models can be predictive but their descriptions are often too verbose (e.g., deep learning models with thousands of parameters) or mathematically restrictive (e.g., assuming that the target variable is a linear combination of input features). Such black-box models have become more prevalent in modern materials science research; however, the interpretability of such models has always been a problem. Although there is a large body of work on data visualization and model understanding to address these issues, those subjects will be out of the scope of this perspective (see Ref. 18 for a review).

In this Prospective paper, we focus on an alternative to ML models: symbolic regression. Symbolic regression simultaneously searches for the optimal form of a function and set of parameters to the given problem and is a powerful regression technique when little if any a priori knowledge of the data structure/distribution is available.

Figure 1 shows the relative popularity of ML and symbolic regression in different research domains. We use data from the "Web of Science Core Collection" database in this analysis.[19] Among all publications whose topics are related to ML or symbolic regression, over 50% of the contributions come from the computer science research community, while multidisciplinary engineering is second. Social science and physical science each makes less than 20% of the contribution to the total number of publications. These two techniques are not so popular in materials science research, as the relative contribution is almost negligible compared to other research domains.

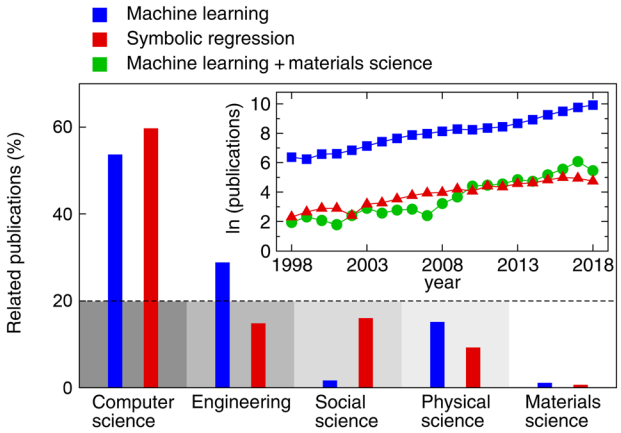


Figure 1. Relative contribution from different research domains to scientific journals related to ML and symbolic regression. Shaded panels indicate a 20% level of the research domain, emphasizing an opportunity in materials science. (inset) The trend in the number of related publications on a natural logarithmic scale (ordinate) related to ML, ML and materials science, and symbolic regression, with respect to time.

It is not surprising to see a dominant contribution from computer science in both the ML and symbolic regression communities, since it is where these techniques were born. It is interesting to notice that symbolic regression is relatively more popular than ML in social science research. One possible reason for this trend is that social science problems typically do not have a (known) physically motivated governing equation as in many physical sciences, where, for example, Newtonian equations of motion and the Schrodinger equation can be written formally. Symbolic regression arises naturally as a problem solver since it has the potential to find an appropriate functional form from social science datasets, e.g., questionnaire results and behavior patterns.

We also report the trend in the number of publications (in a natural logarithm scale) in the following research domains (Fig. 1 (inset)): ML, application of ML in materials science, and symbolic regression. All three domains exhibit a rapid (almost exponential) growth rate, whereas the number of ML-related publications is orders of magnitude larger than the other two. The trend of symbolic regression applications in materials science is not shown here since the base number is too small; nonetheless, it also reveals a potential previously underappreciated research domain. For materials science problems, one is often also presented with the problem of unknown relationships among many variables. Symbolic regression presents an opportunity then to help in the formulation of structure–property relationships derived from these variables.

In this Prospective paper, we encourage materials scientists and engineers to utilize symbolic regression techniques in solving their domain problems. To facilitate a better understanding of the utility and application of symbolic regression, we next introduce the genetic programming-based symbolic regression (GPSR) method and describe current research frontiers in symbolic regression. Next, we discuss several industrial applications of symbolic regression and propose potential uses in materials science. In addition, we present how GPSR can learn the Johnson–Mehl–Avrami–Kolmogorov (or Avrami; JMAK) equation to describe recrystallization kinetics, as well as the Landau free energy expansion describing the structural-phase transition in $LaNiO_3$. Last, we conclude with some open challenges in materials research that may benefit from symbolic regression methods.

Symbolic regression and current state-of-the-art methods

Genetic programming-based symbolic regression

Symbolic regression is a method of finding a suitable mathematical model to describe observed data.[20] In conventional

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

regression techniques, one optimizes parameters for a particular model provided as a starting point to the algorithm. For instance, a linear regression model is based on the assumption that the relationship of the dependent variables and regression is linear[21]; an artificial neural network (ANN) is a nonlinear model which relies on a predefined network infrastructure such as neuron connections and activation function (e.g., sigmoid and softmax function).

In symbolic regression, however, no such a priori assumptions on the specific form of the function are required. Instead, one provides a mathematical expression space containing candidate function building blocks, e.g., mathematical operators, state variables, constants, analytic functions, and then symbolic regression searches through the space spanned by these primitive building blocks to find the most appropriate solution. In other words, both model structures and model parameters are optimized in symbolic regression. Since there is no need for a predefined function form, optimization algorithms used in symbolic regression are different from conventional analytic/numerical optimization methods (e.g., conjugate gradient and Newton–Raphson methods). In this section, we briefly introduce one of the most prevalent methods used in symbolic regression by means of genetic programming (GP).

GP was developed by Koza[22] as a specific implementation of genetic algorithms (GAs),[23] which are often utilized in the materials community for atomic structure prediction.[24–26] The idea is to evolve the solution of a given problem following Darwin's theory of evolution and to find the fittest solution after a number of generations. Instead of using strings of binary digits to represent chromosomes as in GA, solutions in GP are represented as tree-structured chromosomes with nodes and terminals. Figure 2(a) shows a chromosome example of the mathematical function $1+\exp(-x_1)$. The tree consists of a set of interior nodes with mathematical operations (+, ×, exp) and terminal nodes with variables (x_1) and constants (±1). A depth-first search can be used to traverse the tree to get the final mathematical expression of each individual solution.

The structure of a chromosome tree is not necessarily binary; its structure depends on the number of arguments the mathematical operator takes. For demonstration purposes, we only introduce simple operators that are either unary or binary. Users of GP could include a variety of functions suitable for

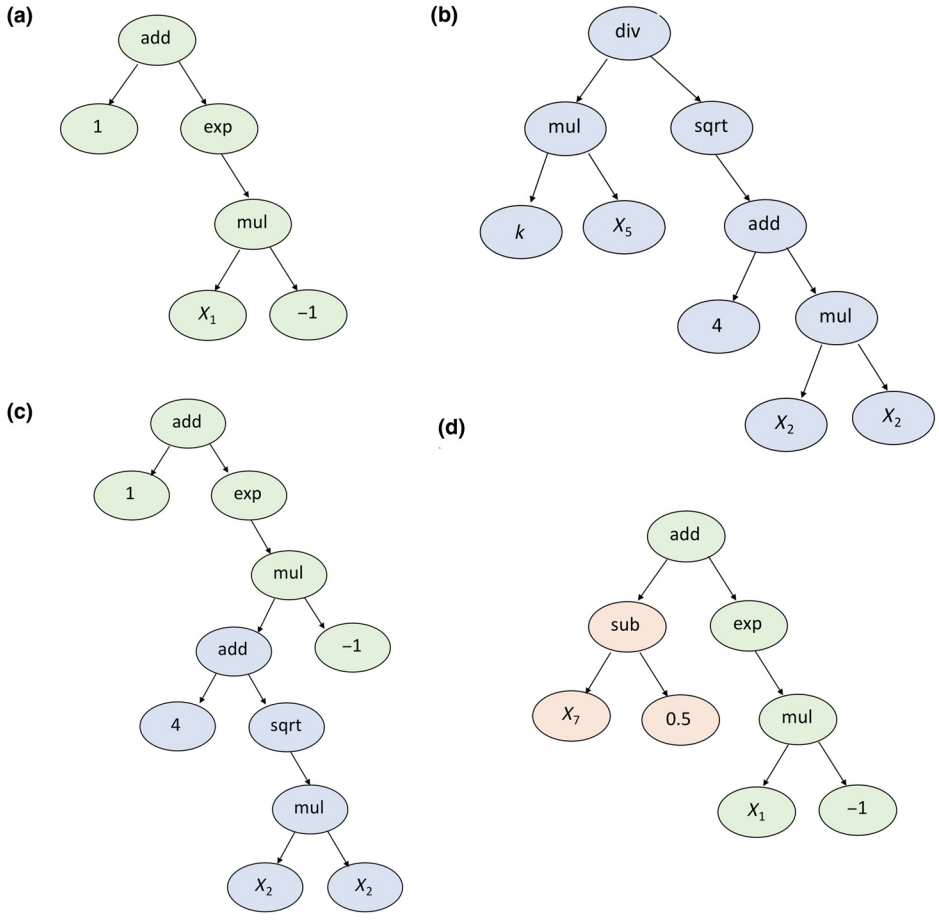


Figure 2. Tree structure chromosome representation of computer programs in GP. (a) parent 1 $(1+\exp(-x_1))$; (b) parent 2 $\left(kx_5/\sqrt{x_2^2+4}\right)$; (c) child of genetic crossover operation $\left(1+\exp\left(-\sqrt{x_2^2+4}\right)\right)$; and (d) child of subtree mutation operation $(x_7-0.5+\exp(-x_1))$.

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

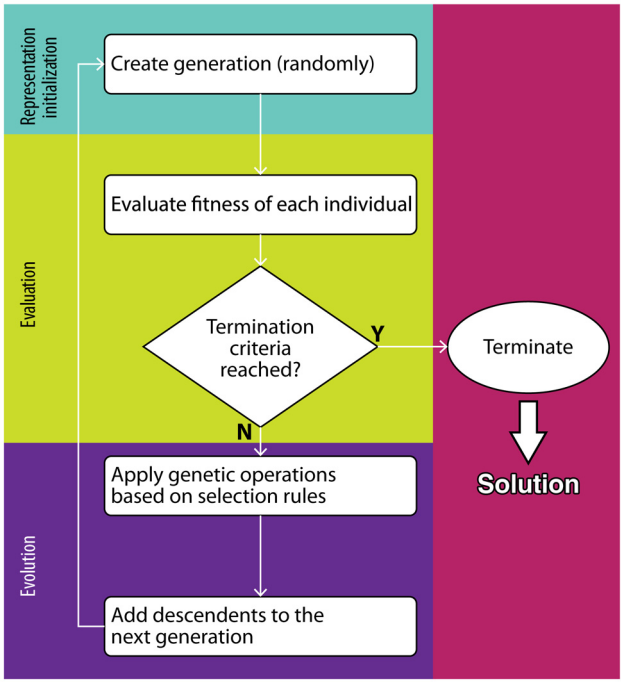


Figure 3. GP flowchart depicting the iterative solution-finding process.

their target problems. A large number of trees will be generated based on specified user settings and evaluated throughout the GP process. Each tree represents a potential solution of the problem. The way new trees are generated from the initial mathematical building blocks is a unique feature of GP since it mimics the natural evolution of Earth's ecosystem, i.e., through artificial sexual recombination and a natural selection process.

Figure 3 illustrates the process by which a solution of the symbolic regression problem is obtained using GP. The procedure starts with a set of randomly generated initial terminal nodes (variables, constants) and functions, forming individual trees with different sizes and structures (Figs. 2(a)–2(d)). These fundamental building blocks come from a user-defined input set. This starting population typically has a large variety of tree structures due to the random process, which facilitates further exploration of the variable space and reduces the potential risk of being trapped in local minima. The initialization process terminates once the number of individuals reaches a user-defined population size, where the natural selection process then comes into play. The "fitness" of each individual solution in the initial population is then evaluated by comparing their function output with the true value from the dataset. This fitness value describes how well the program performs in terms of solving the problem. The common error metrics used include mean squared error and root mean squared error. Then, GP evolves the current generation by randomly applying genetic operations to individuals, e.g., crossover and mutation. One or more individuals from the current generation will be selected as parent(s) based on the fitness score, typically the higher the score, the larger probability to be selected for reproduction. Such a selection rule agrees with the "survival-of-the-fittest" rule since good features are more likely to be inherited by the next generation, which is the essential step towards the optimal solution.

The genetic crossover operation takes two winners of the selection process as parents to breed their offspring. For instance, the two structures in Figs. 2(a) and 2(b) are taken as parents. The crossover operator then randomly takes a subtree from parent (b) and substitutes another random subtree in parent (a) with that from (b). One possible offspring from the crossover operator is illustrated in Fig. 2(c). Crossover is usually the dominant operation in the recombination process. Figure 2(d) shows an example of an offspring from the mutation operation. The mutation operator only takes one parent structure and randomly substitutes a subtree with another randomly generated structure; in case (d), the constant 1 is mutated to $(x_7 - 0.5)$. Although this operation is more aggressive compared to the crossover operation, since it adds randomness to the system, it is important to have a finite chance of mutation to introduce new variations, e.g., new constants and new features, and avoid being trapped in local minima.

The third category of genetic operations is reproduction, which duplicates the selected program and directly inserts its offspring to the next generation. It guarantees that some of the current generation will be preserved by the next generation and partially protects the similarity between two generations. Detailed definitions and implementations of each genetic operation can vary from case to case, but the main features should be the similar to what we described here.

The newborns are then added to the next generation after each genetic operation, until the new population size reaches the specified set number. Then, the new generation goes through the fitness evaluation and the natural selection process again until the fitness value reaches a certain criterion or the maximum number of generations is reached. After the termination of GP, the surviving individuals are expected to be highly evolved to adapt to the problem-dependent selection rule. More comprehensive descriptions of GP can be found in Koza's original paper.[22]

Advances in symbolic regression

Since Koza introduced the idea of GP in 1992, there have been significant efforts made to improve the performance of the original GPSR algorithm. The major problems to overcome in GPSR include:

(i) *Non-deterministic optimization*: It is not guaranteed that the performance of the descendent generations will be better than their parents.

(ii) *Difficulty in finding the proper constants*: Since the way GP generates constants is random, either in the initial input set or those brought into the population by mutations, there is

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

no effective way to obtain the ideal coefficients as in other numerical regression methods.

(iii) *Limited capability to preserve good components of the equation* due to the fitness evaluation method. The fitness is evaluated based on the complete structure of an individual. Having a good feature in a sub-branch does not necessarily lead to better individual performance, thus good equation components may get lost in the next generation.

We summarize some of the most popular alternative methods to conventional GPSR in Table I and discuss their similarities as well as the differences in four aspects, namely program representation, fitness evaluation, optimization method, and the solution form.

Multiple regression genetic programming (MRGP)[27] improves the program evaluation process by performing multiple regression on subexpressions of the solution functions. Instead of evaluating the fitness of each individual solution as a whole, MRGP decouples its mathematical expression tree into subtrees. The fitness of the solution is evaluated based on the best linear combination of these subtree structures. A least angle regression algorithm is used to solve the linear regression problem here. Such a fitness evaluation scheme places more emphasis on finding good components even though it might only be a partial solution. For instance, the individuals that contain a correct form of a subtree structure of the correct solution (if known) are more likely to survive the natural selection process and pass these good features to the descendents. MRGP essentially decouples the current basis functions to find the best solution in an enlarged space at the vicinity of the original GP space. Indeed, this feature may be well suited for multiscale materials problems where modeling of systems across different length/time-scale is desired.[28] While some subexpressions capture relationship among variables within each scale, the final symbolic regression solution assembles models across the scale and returns the multiscale model.

Geometric semantic genetic programming (GSGP)[29] evaluates the semantic performance of a computer program instead of the syntax performance as in conventional GPSR. While still using a rooted-tree structure to represent computer programs, GSGP focuses on its semantics, i.e., the behavior of a program. For instance, add(x_1, x_1) is equivalent to mul(2, x_1) in semantic space, but quite different in terms of syntax. It is reasonable to care more about the behavior of the program than how the function appears. By representing each program in a high-dimensional semantic space, the fitness evaluation is rather straightforward; one only needs to measure the distance of the program from the target point in that space. The closer a program is to the target point, the better performance it has in solving the given problem. Interestingly, the offspring of two parent vectors in semantic space lies between its parents in the semantic space; therefore, the offspring should be at least no worse performing than the poor-performing parent. Optimizing program semantics rather than syntax further frees symbolic regression from specific function forms, potentially making SR more efficient.[29]

Cartesian genetic programming (CGP)[30] has a more sophisticated design than conventional GP. Here, a computer program is represented as a directed acyclic graph, which may be visualized as a two-dimensional grid of nodes. Each node owns a set of genes that determines the input–output and mathematical function that the node performs; the whole set of genes of the computer programs forms its genotype. Decoding the genotype leads to the phenotype, i.e., the function forms of the computer programs. The genotype–phenotype mapping is a unique feature of CGP which makes it closer to the real natural process.

GP-based relevance vector machine (RVM)[31] is an alternative GP method that combines Kaizen programming (KP) and an RVM algorithm to solve symbolic regression problems. KP is a collaborative version of GP, where individuals work together with each other to solve the problem. The solution of a Kaizen process is a linear combination of GP individuals, and thus the fitness evaluation is based on a group of individual partial solutions instead of an individual program as a complete solution. RVM is a Bayesian kernel method that could extract important basis functions from the basis set without the prior knowledge to set a threshold and automatically deals with

Table I. Comparison of GPSR and its alternative methods: MRGP, GSGP, CGP, GP-RVM, EPR, and FFX.

	Program representation	Fitness evaluation	Optimization	Solution form
GPSR	Rooted-tree	Individual program	Genetic evolution	Rooted-tree
MRGP	Rooted-tree	Subexpressions	Genetic evolution + linear regression	Linear combination of subexpressions
GSGP	Rooted-tree	Distance in semantic space	Genetic evolution in semantic space	Rooted-tree/semantic vector
CGP	Acyclic graph	Individual program	Genetic evolution	2D grid of nodes
GP-RVM	Rooted-tree	Group of GP individual	Genetic evolution + RVM	Linear combination of GP individuals
EPR	Vector of integers	Individual program	Genetic evolution + linear regression	Polynomial function
FFX	Basis functions	Individual program	Pathwise regularized learning	Linear combination of basis functions

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

singularity. GP-RVM leverages advantages from both evolutionary algorithm and Bayesian kernel methods: the former mainly explores the parameter space, while the latter extracts basis functions to build and solve for the optimal solution function within that space.

Evolutionary polynomial regression (EPR)[32] hybridizes the parameter estimation used in conventional numerical regression methods with the evolutionary optimization scheme in GPSR. EPR first explores the function space using GAs and then performs linear regression (e.g., least squares) to optimize the coefficients of each mathematical building block. Although EPR specifically uses polynomial expansions for the form of the functions, the solution is not necessarily a simple polynomial function since the transformed variables used in the polynomial expansion could be nonlinear functions of independent input variables. Such a hybrid method improves the stochastic GPSR method moving it towards a more deterministic approach although the computational cost may be relatively higher. In fact, the polynomial form of the expressions could make EPR suitable for materials design or multiobjective optimization purposes. Since the analytic gradient and Hessian of the solution can be evaluated, materials scientists may have more insights regarding the system and know what parameters to tune in order to achieve optimal design.

Fast function extraction (FFX)[33] is an efficient way to find good basis functions and solve for the best solution within the space it spans. The first step in FFX is to generate a large number of candidate basis functions built from input variables and other predefined variables. The evolutionary optimization scheme is not involved in FFX; instead, a pathwise regularized learning technique is used to identify the best coefficients and basis functions for the solution. Then, models obtained from the previous step are assessed based on the validation dataset as well as their model complexity in order to identify the best solution. FFX is more efficient compared to other GP-based methods due to the deterministic optimization technique. Materials scientists could first use FFX to see whether the input function/variable basis is sufficient for their research problem, before further investigation using symbolic regression methods (either FFX or other variants).

The performance of some of the recently developed symbolic regression techniques has been assessed against popular ML methods,[34] and it is reported that symbolic regression performs considerably well compared to state-of-the-art ML algorithms with regard to predictive accuracy. However, the two methods do not simply exist in competition to one another. We also observe a trend of more hybridization between conventional ML algorithms and GP in symbolic regression solvers.[35–38] These advances have enabled symbolic regression to be used for solving real-world problems, which we will discuss in the following section.

Applications of symbolic regression

Although it seems that equations obtained from first principles (e.g., the Schrödiner equation) and empirical observations (e.g., the 18-electron rule[39]) are quite contradictory to each other, we see quite often that they symbiotically work together in solving real-world problems. For instance, both ab initio and experimental data have been used to develop effective interatomic force fields[40] or exchange–correlation functionals.[41] In fact, symbolic regression has the potential to serve as the bridge connecting experimental data to first principles. Schmidt and Lipson[42] demonstrated that symbolic regression is capable of predicting connections between dynamics of subcomponents of the system and distill natural laws from experimental data. Moreover, symbolic regression provides researchers with analytic equations, which expectably would have better interpretability over the raw data and potentially other black-box models. The equations could reveal how the dependent variable (system output) responds to multiple independent variables (system input), as well as the relationships between independent variables of the underlying function. We show this later in the section "Industrial applications."

Common motivations underlying the use of GPSR for complex problem solving include when the system in question is not effectively modeled by a linear model. Existing multiple linear regression models are much faster and are already easy to interpret. GPSR is best used for systems with complex interactions between observable variables for which the form is not known beforehand—a situation common in materials science and engineering.

In addition, a GPSR approach could be useful for design optimization purposes. Although the evolutionary search process is a black box, the final solution is analytic, which potentially contains important information (e.g., regarding the gradient or Hessian) about relationships between the design variables and objectives. There is also need for multiobjective optimization such as finding the Pareto optimal combination of model performance and complexity in various domains—it is here that the symbolic regression technique has shown to be effective and interpretable.[43] We next describe some applications of symbolic regression in various science and technology domains.

Industrial applications

GPSR has been applied to a wide variety of problems in fields outside of materials science and chemistry. Most prominently featured in the popular press was work published by Schmidt and Lipson in *Science*,[42] which showed that GPSR could discover Hamiltonians and Lagrangians for systems of simple harmonic oscillators and double pendulums. Reports of using GPSR for real-world systems, however, have been published since Koza's origination of the idea in the early 1990s and continue today.

Arkov et al.[44] used GPSR to identify equations governing gas turbine engines under multiple optimization conditions. Berardi et al.[45] used GPSR to find easy to interpret models for pipe failures in a UK water distribution system. Bongard and Lipson[46] applied GPSR to generate symbolic equations for nonlinear-coupled dynamical systems in mechanics,

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

ecology, and systems biology. The authors also emphasized that their symbolic models are easier to interpret than numerical models, which makes understanding more complex systems easier for future applications.

Cai et al.[47] identified correlation equations from experimental heat transfer measurements using GPSR with a sparsifying constraint. The authors' predicted correlations had lower percentage error than models developed graphically and numerically, albeit with more formula complexity than those traditional methods. Can and Heavey[48] applied GPSR to develop metamodels for predicting throughput rates in industrial serial production lines. McKay et al.[49] developed steady-state models for a vacuum distillation column and a chemical reactor.

La Cava et al.[50] applied GPSR to identify nonlinear governing equations of wind turbines. The Pareto front from their paper is reproduced in Fig. 4. The Pareto front illustrates the trade-off between their model complexity as defined by the number and type of operations in the equation and the normalized variance in the prediction error. La Cava et al.[51] also tested modifying standard GPSR with features from epigenetics, such as passive structure, phenotypic plasticity, and inheritable gene regulation. These researchers demonstrated their modifications improved the performance over standard GPSR by finding compact dynamic equations for synthetic data from nonlinear ordinary differential equations as well as real-world systems, e.g., cascaded tanks, a chemical distillation tower, and an industrial wind turbine. GPSR has also been

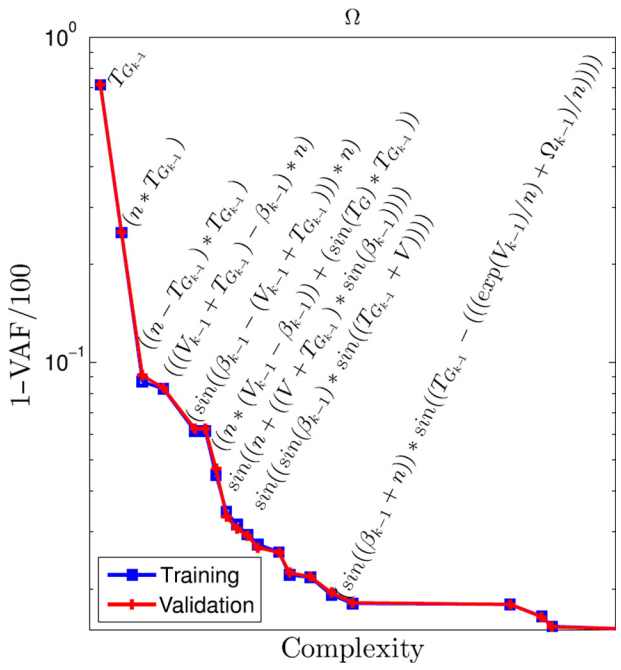


Figure 4. Example Pareto front showing trade-off between solution complexity and variance accounted for (VAF). Reproduced with permission from La Cava et al.[50]

applied to testing the efficient market hypothesis,[52] formulating the synchronization control in oscillator networks,[43] identifying the structure of helicopter engine dynamics,[53] real-time runoff forecasting in France[54] and Singapore,[55] designing circuits,[30] predicting solar power production,[56] finding dynamical equations for metabolic networks[57] in both cases where a starting model was known and from scratch, modeling global temperature changes,[58] and synthesizing second-order coefficient insensitive digital filter structures.[59]

The existing uses of GPSR within chemistry are more extensive than that for materials science. We refer the reader to the review by Vyas et al.[60] for further details. Some key studies with relevance to materials science are summarized here: Langdon and Barrett[61] developed a model for oral bioavailability of a small molecule given a few hundred data points from expensive experiments. Their model based on chemical descriptors showed promise for rapid drug screening, but had difficulty generalizing to novel molecules. Vyas et al.[62] also discovered structure–property relationships for drug absorption using GPSR. Here, they demonstrated R^2 values comparable to those achieved with ANNs and support vector regression. Barmpalexis et al.[63] performed a multiobjective optimization using GPSR. They found a function mapping levels of four polymers to three different properties of a pharmaceutical release tablet that was more predictive than a shallow neural network. Lastly, Muzny et al.[64] built a correlation model for the viscosity of hydrogen as a function of temperature and pressure.

Opportunities in materials science

Materials science has many potential areas where GPSR can be applied for the same reasons it find use in other disciplines. Nonlinear systems are abundant in materials science. Changes in materials properties occurring in response to structural, composition, and other external perturbations are frequently nonlinear in proximity to phase transitions or for large stimuli. For instance, changing the concentration of oxygen vacancies in a transition metal compound by an atomic percent can alter its ionic or electronic conductivity by orders of magnitude.[65,66] The dynamical behavior of materials performance as a function of time is also of broad interest and technological importance, e.g., corrosion of nickel cathodes under different conditions.[67] These are the areas where a dynamical multivariable model would be ideal to understand the correlation among the variables and assist optimization of materials properties, e.g., corrosion resistance.

Frequently, materials scientists look for relationships (f) among multiple variables with the aim to find some closed-form expression such as $y = f(X)$, where y is the objective value and X is a set of variables. These equations are typically expressed in differential form, e.g., the Schrödinger equation ($i\hbar\,(d/dt)|\Psi(t) = \hat{H}|\Psi(t)$) or Newton's second law ($\boldsymbol{F} = m\,(d\boldsymbol{v}/dt)$). It has been shown that symbolic regression can generate ordinary nonlinear partial differential equations for nonlinear-coupled dynamical systems[46,68] as well as approximate

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

ordinary differential equations.[69] Meanwhile, an often desire is to find conservation laws in physical systems. The ability to unearth conservation laws with symbolic regression goes beyond the aim of materials property predictions and helps researchers establish insight into the materials systems they study.[42,70] In fact, we do not necessarily need a rigorous expression of natural laws in every case; sometimes, an approximation with a simple yet effective expression serves well for the research purpose.[71] Symbolic regression could potentially balance the trade-off between model accuracy and simplicity, and might even help scientists discover new equations that redefine our understanding of functional materials in the same way those of Hall and Petch, and Harper and Dorn changed our understanding of the mechanical properties of metals or as, more recently, how Berry phases and topological band theory changed our understanding of electronic structures.

As we mentioned earlier, materials properties and performance are affected by phenomena that involve multiple length scales. Most theoretical models are formulated to be optimal at a particular length scale. However, recent emphasis has been placed on multiscale and hierarchical modeling in the materials science community,[28,72,73] and there is an increased need for effective, descriptive, and predicative, multiscale models. Symbolic regression techniques are potential solutions to this challenge by directly searching for the interactions among variables operating and passing between multiple spatial and temporal scales. Another possible approach is to utilize existing simulation methods within each length scale, while using symbolic regression to find the suitable coupling interactions between scales, i.e., connecting models of different scales.

Other applications of GPSR in materials and molecular systems are in areas where supervised ML has already demonstrated usefulness in providing new insight or solutions. While ML has produced many impressive results,[5,17,74] it is commonly understood that ML models exhibit a trade-off between performance on prediction metrics and the ability to explain the predictions of a model due to the complexity of state-of-the-art models like deep neural networks or gradient-boosted decision trees. GPSR offers a middle ground with comparable performance but with the added ability to read and directly interpret the output function.

GPSR is also well suited to the development of new descriptors[75] for materials properties. By combining features in a manner best suited to fitting data, new features are created that can be used as proxies for the property in question. This is also a common application of compressed sensing.[76,77] Compressed sensing differs from GPSR in that while GPSR uses GP to iteratively evolve a solution, compressed sensing tries to enumerate as many combinations of primary features as possible and then uses sparsifying operators to find a small dimensional subset that correlates with the target.

Materials scientists are not just interested in making predictions. They also want to identify the controlling features of a property and what role each feature plays; they want to understand which degrees of freedom to design or optimize to achieve targeted properties. Given some desired properties as objectives and the relevant variables, there are a number of numerical algorithms available for performing optimization,[78] but they typically require some a priori knowledge of the mathematical relationships within the system. The symbolic equations derived from GPSR offer insight into which microscopic or macroscopic knobs to turn for the design of desired functionality, such as corrosion resistance in steels.[79]

Some novel ideas include targeting physical variables for which we do not know the proper mathematical expressions. For instance, the exchange–correlation functional used in density functional theory, or the correlation function for the viscosity of normal hydrogen.[64] Contraindicated material property pairs such as ferromagnetism and ferroelectricity or optical transparency and electrical conductivity would be interesting areas to pursue in search for routes to decouple or circumvent perceived coexistence incompatibilities. In these cases, GPSR could provide candidate representations of these terms, where our physical knowledge can be used to filter meaningful results. One advantage of using symbolic regression is that the balance between model accuracy and complexity is tunable with GPSR settings, e.g., stopping criteria and penalty on individual size. This advantage is particularly evident when designing with constraints or optimizing for performance. The recommendations coming from the learned symbolic equations are more actionable than learned functions only optimized for test set accuracy since they are rigorously made to use fewer terms.

Use-cases in materials science

Discovering the JMAK equation

We now show how to use GPSR to learn the JMAK, hereafter, Avrami equation. The Avrami equation quantitatively describes the growth kinetics of phases in materials at constant temperature. Here, we specifically study the recrystallization process of copper; the original experimental data were obtained from Ref. 80. We expect the form of the function to be

$$y = 1 - \exp(-kt^n)\,,$$

where the phase fraction of transformation y is a function of time t. The coefficients k and n are unknown and change with respect to temperature and other environmental conditions.

We use GPSR as implemented in `gplearn`[81] to predict the relationship between fraction transformed y and time t. The hyperparameters used in GPSR are listed in Table II. When the population size is divided by the tournament factor, one obtains the number of individuals competing for reproduction each round. The parsimony coefficient regularizes the size of individuals by penalizing over-sized structures. We include operations of addition, subtraction, multiplication, negation, and the natural exponential function into the function set. The power function is not included since it easily causes numerical overflow or invalid operations (e.g., power $(-1,0.5)$). In general, this can be an issue when evaluating

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

Table II. List of hyperparameters used in GPSR to learn the Avrami equation. A grid-search method is used to find the optimal hyperparameters from the top three parameter sets.

Parameter	Values
Population size	{2000, 5000}
Tournament factor	{100, 500}
Parsimony coefficient	{0.001, 0.005}
Maximum generation	20
Constant range	(−1, 1)
Function set	{add, sub, mul, neg, exp}
Crossover probability	0.7

power law-dependent phenomena such as electrical transport equations. Working with log transforms of the original variables may be a more stable approach. Crossover operations dominate the genetic operations with a 70% probability to be applied; the other 30% chance corresponds to mutation operations (e.g., point mutation and subtree mutation). Additional details concerning the usage of gplearn are given in its documentation.

Data preprocessing is an essential step in ML before feeding data into the solver. Conventional preprocessing methods include shifting data to be zero-centered and scaling data to unit standard deviation. However, the conventional preprocessing methods are not ideal choices in our case. Zero-centered shifting is not applicable to either the phase fraction transformed (y) or time (t) since we want to obtain the exact function form of the Avrami equation. Furthermore, scaling the time frame would only change the constant k in the Avrami equation. Here, we scale time from 0 to 10 for all datasets so that the constant t lies in the range [−1, 1]. The y values remain unchanged as the experimental data range over [0, 1].

We take the experimental copper recrystallization data at temperatures 135, 113, and 102 °C as input and scale the time variable before performing regression. Since the experimental data only contain several points at each temperature (less than 10), we also take data points from interpolated lines and perform symbolic regression on the interpolated data for comparison. An ideal dataset is also generated, where y values are directly calculated as $1-\exp(-0.6t^2)$. The optimal k and n values in each dataset are obtained from numerical fitting using `Scipy`, given the form of the Avrami function. Ideally GPSR should be able to recover the correct function form as well as k and n constants for all datasets.

The best individual after 20 generations of evolution is collected for each hyperparameter setting. Finally, we manually pick the optimal individual with closest function form and constants to the Avrami equation within each dataset. Our parameter fitting and GPSR evolution results are shown in Table III and Fig. 5.

`Gplearn` successfully recovers the relationship between time (t) and fraction transformed (y) in most cases. With the ideal input data, `gplearn` evolves almost a perfect form of the Avarmi function as well as the constants (see the first row of Table III). The performance on the raw experimental and interpolated data is generally worse than the ideal case but still reasonable. In most cases, the exponential function form and polynomial function of t are both recovered. One source of error is the lack of the power function in function set, which was intentionally omitted as non-integer powers of variables cannot be correctly represented here. Rather, polynomial functions are used as an approximation. With the limited choice of mathematical functions, GP produces results that are very close semantically but exhibit different syntax.

We see in multiple cases that GPSR produces functions of the $\exp[-\exp(\Theta)]$ form, where the expected function form is $1-\exp(\Theta)$. Here, we introduce a mathematical trick to show their equivalence. The functions e^x and $1+x$ are called equivalent infinitesimals because

$$\lim_{x\to 0} e^x \sim 1+x.$$

In our case, the exponent $-\exp(\Theta)$ quickly reaches zero when Θ (ideally having the form $-kt^n$) becomes more negative;

Table III. Functions predicted by GPSR and numerical fitting results for the Avarmi equation describing copper recrystallization.

Dataset	Numerical fitting result	GPSR result	GPSR (transformed y) result
Ideal	$y=1-\exp(-0.6t^2)$	$y=0.994-\exp(-0.58t^2)$	$1-y=\exp(-0.58t^2)$
135 °C experimental	$y=1-\exp(-0.019t^{3.2})$	$y=\exp[-\exp(-t+2.64)]$	$1-y=\exp(-0.024t^3)$
135 °C interpolated	$y=1-\exp(-0.17t^{3.6})$	$y=\exp[-\exp(-t+2.18)]$	$1-y=\exp[-0.065t^2(t-0.851)]$
113 °C experimental	$y=1-\exp(-0.018t^{2.8})$	$y=0.986-\exp(-0.051t^2)$	$1-y=\exp(-0.014t^3)$
113 °C interpolated	$y=1-\exp(-0.042t^{2.5})$	$y=0.986-\exp[-0.108\times t\times(t-0.934)]$	$1-y=\exp[-0.097t(t-0.764)]$
102 °C experimental	$y=1-\exp(-0.0054t^{2.9})$	$y=0.1t-0.081$	$1-y=\exp(-0.0045t^3)$
102 °C interpolated	$y=1-\exp(-0.01t^{3.1})$	$y=\exp[-(1.11+2t)\times\exp(-t+1.42)]$	$1-y=\exp(-0.01t^3)$

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

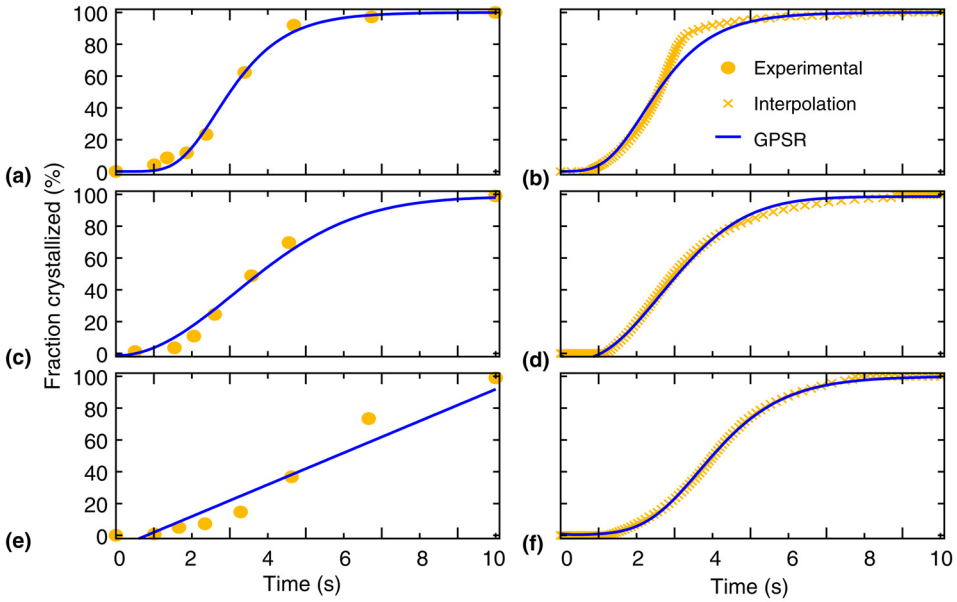


Figure 5. GPSR prediction and performance with different datasets. Left panels correspond to the direct experimental data while right panels use interpolated experimental data (a) 135 °C experimental data, (b) 135 °C extrapolated data, (c) 113 °C experimental data, (d) 113 °C extrapolated data, (e) 102 °C experimental data, and (f) 102 °C extrapolated data.

therefore, the equivalent infinitesimal relationship holds and the predicted form of the function is equivalent to the expected one appearing in the Avrami expression. Based on this example, we encourage professional materials researchers, who are novice data scientists, to perform careful analysis of GPSR results when making the final interpretation of the obtained model.

In real-world applications where the actual function forms in unknown, the exact syntax of GPSR results may not matter. There are potentially many cases where GPSR would produce semantically similar/equivalent functions that vary in their syntax, i.e., different function forms. Considering the trade-off between model accuracy and complexity, in some cases it might be a virtue to find a simple approximated solution instead of using rigorous but complex relationships among multiple variables.

The GPSR result from the 102 °C experimental dataset (Fig. 5(e)) turns out to be a linear relationship and deviates significantly from the original data. The poor result may be caused by the relatively small slope of the original data and the insufficient number of available data points. Taking the model performance and complexity into consideration, we find that the linear relationship survives due to its simple form. With interpolated data as input, functions evolved from GPSR agree better with the Avrami equation, as shown in Fig. 5(f). Therefore, having more training data could improve the performance of GPSR, which also applies to other data-driven methods.

Interestingly, the performance of GPSR can also be improved by transforming (or simplifying) the mathematical expression. We compare results of directly training $y \sim t$ relationships with $(1-y) \sim t$, where in the latter case the target value $1-y$ is the percentage of copper untransformed. The results are shown in the last column of Table III. The transformed functions show improved performance since we have already performed the subtraction function for the model. GPSR not only successfully recovers the exponential form of the equation, but also finds constants closer to the numerical fitting values.

Learning Landau free energy expansion

Next, we present a slightly more complicated case with two variables. The model we studied is the Landau free energy expansion for the cubic-to-rhombohedral structural-phase transition in perovskite $LaNiO_3$, where the free energy G of the system is expanded in powers of an order parameter θ as

$$G(\theta, T) = G_0(T) + \kappa(T - T_C)\theta^2 + \lambda\theta^4, \qquad (1)$$

where κ and λ are temperature-independent coefficients and θ is the angle of rotation about the [111] direction. This rotation angle of the corner-connected NiO_6 is the order parameter for the displacive transition. The parameters we used are obtained from ab initio DFT simulations,[82] where $\kappa = 1.696$ meV 10^{-3} K/(°)^2, $\lambda = 0.0171$ meV/(°)^4, and T_C is estimated to be 2.057 (10^3 K). We use 10^3 K as the unit for temperature so that the constants are brought into a smaller range.

In this case, we set the temperature t and order parameter θ as the input variables, and the free energy change $G(\theta, T) - G_0(T)$ as the output. We uniformly sampled 11 temperature points between [0, 1] (10^3 K), and 100 order parameter points within the range [−20°, 20°]. The corresponding value for the

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

change in free energy is calculated from Eq. (1) using the parameters reported in the literature. A population size of 10,000 is used, with tournament size 25, parsimony coefficient 0.02, and constant range [–2.0, 2.0]. In order to simplify the problem, we only consider addition, subtraction, and multiplication operations in our search. Other settings are the same to those in the Avrami case.

The best individual after 15 generations of evolution has the function form

$$G(\theta, T) = G_0(T) + 1.983(T - T'_C)\theta^2 + 0.0165\theta^4 + \xi, \quad (2)$$

where $T'_C = 1.894 + 8.32 \times 10^{-3}T^2$ and $\xi = (-1.72T^2 + 1.214T - 0.24)\theta - 1.334$. The GPSR-learned coefficients are quite close to the reported values, especially for the quartic term. This is probably because the penalty for a larger deviation in the leading (quartic) term is much higher than others. For the quadratic term of θ, GPSR successfully captured the coupling term $T\theta^2$ and its coefficient, but also an unexpected biquadratic $T^2\theta^2$. It should not have a strong impact on the function owing to its small coefficient (8.32×10^{-3}).

The Landau free energy with respect to the order parameter is plotted in Fig. 6. We find that GPSR results (filled symbols) agree well with the DFT-derived Landau free energy function not only within the training region, i.e., with $T = 0$, 500 and 1000 K, but also reasonably well beyond it. The dashed red line in Fig. 6 with $T = 3000$ K is not included in training the GPSR, yet the model reproduces both the shape and the correct global minimum position very well. However, the predicted function contains other coupling terms not present in Eq. (1). These extra terms in ξ destroy the symmetry of the function, i.e., the free energy expansion is an even function by symmetry. This would become an issue especially when T is large, as we can see a minor shift to positive θ values occurs for the red-filled symbols in Fig. 6. Again, the model may need more data to learn the correct even function form. In general, GPSR does an excellent job in learning the relationship between temperature and the order parameter without any knowledge of the physical system. Our results here also reveal the potential to perform effective feature selection using GPSR, where the insignificant variables (features) could be approximated or ignored in post-processing (e.g., terms with negligibly small coefficients).

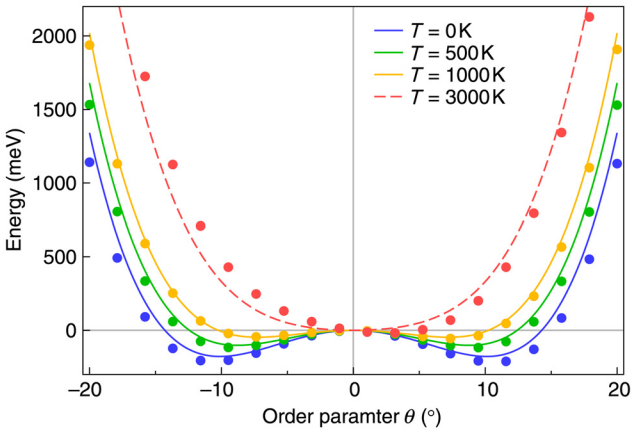


Figure 6. Landau free energy G for perovksite $LaNiO_3$ as a function of the order parameter θ at different temperatures. Both solid and dashed lines are calculated using Eq. (1) with coefficients reported in Ref. 82. Only solid lines are used during the training. The filled symbols correspond to GPSR-predicted results using Eq. (2).

The purpose of these two use-cases for learning the Avrami equation and Landau free energy expansion with GPSR is to show the potential of its application in materials science problems. Although these are relatively simple examples, the understanding and approaches applied could be generalized and utilized to solve real-world challenges. Apart from both function analysis and data preprocessing mentioned previously, hyperparameter tuning is also an essential step to obtain the optimal solution. A grid-search scheme to search the hyperparameter space (e.g., population size, number of generations, and regularization) is recommended since the optimal settings differ from case to case. The grid-searching process can be exhausting, but it might also be rewarding. Comparing the results from different hyperparameter settings could provide insight into the functional form of the optimal solution, especially if components of a particular function appears multiple times in the solution set.

The real challenge, however, is that very often materials science problems cannot be represented using regular functions (analytic and single-valued). A simple example would be to understand how different chemical compositions affect materials properties.[83] This problem originates from the inability to differentiate in the chemical space and is out of the scope of this prospective. It remains an open question in the materials research community.

Summary

Symbolic regression has shown competitive performance to other ML-based regression models in various research domains. While there are some shortcomings of the current state-of-the-art GPSR, e.g., high computational cost and non-deterministic optimization, there are numerous active research efforts focusing on improving the performance of symbolic regression to expand its use in real-world applications. The ability of symbolic regression to distill natural laws from datasets with high-dimensional parameter space makes it an ideal technique for materials science research, since these researchers typically face sparse datasets with multiple variables. Freed from having a fixed form of equations, symbolic regression can potentially reveal the significant interactions among physical variables. We recommend that more materials scientists utilize symbolic regression techniques in their own research domain.

Acknowledgments

Y.W. acknowledges partial support from the Predictive Science and Engineering Design (PS&ED) program at Northwestern

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

University. All authors acknowledge support from the National Science Foundation (NSF) through the Designing Materials to Revolutionize and Engineer our Future (DMREF) program under award no. DMR-1729303.

References

1. E. Deelman, C. Carothers, A. Mandal, B. Tierney, J.S. Vetter, I. Baldin, C. Castillo, G. Juve, D. Król, V. Lynch, B. Mayer, J. Meredith, T. Proffen, P. Ruth, and R.F. da Silva: PANORAMA: an approach to performance modeling and diagnosis of extreme-scale workflows. *Int. J. High Perform. Comput. Appl.* **31**, 4–18 (2017).
2. A.R. Lupini, M.P. Oxley, and S.V. Kalinin: Pushing the limits of electron ptychography. *Science* **362**, 399–400 (2018).
3. F. Ren, R. Pandolfi, D. Van Campen, A. Hexemer, and A. Mehta: On-the-fly data assessment for high-throughput X-ray diffraction measurements. *ACS Comb. Sci.* **19**, 377–385 (2017).
4. H.S. Stein, D. Guevarra, P.F. Newhouse, E. Soedarmadji, and J.M. Gregoire: Machine learning of optical properties of materials predicting spectra from images and images from spectra. *Chem. Sci.* **10**, 47–55 (2019).
5. K. Alberi, M.B. Nardelli, A. Zakutayev, L. Mitas, S. Curtarolo, A. Jain, M. Fornari, N. Marzari, I. Takeuchi, M.L. Green, M. Kanatzidis, M.F. Toney, S. Butenko, B. Meredig, S. Lany, U. Kattner, A. Davydov, E.S. Toberer, V. Stevanovic, A. Walsh, N.-G. Park, A. Aspuru-Guzik, D.P. Tabor, J. Nelson, J. Murphy, A. Setlur, J. Gregoire, H. Li, R. Xiao, A. Ludwig, L. W. Martin, A.M. Rappe, S.-H. Wei, and J. Perkins: The 2019 materials by design roadmap. *J. Phys. D: Appl. Phys.* **52**, 013001 (2019).
6. M.L. Green, C.L. Choi, J.R. Hattrick-Simpers, A.M. Joshi, I. Takeuchi, S.C. Barron, E. Campo, T. Chiang, S. Empedocles, J.M. Gregoire, A.G. Kusne, J. Martin, A. Mehta, K. Persson, Z. Trautt, J. Van Duren, and A. Zakutayev: Fulfilling the promise of the materials genome initiative with high-throughput experimental methodologies. *Appl. Phys. Rev.* **4**, 011105 (2017). doi:10.1063/1.4977487
7. W. Ye, C. Chen, S. Dwaraknath, A. Jain, S.P. Ong, and K.A. Persson: Harnessing the Materials Project for machine-learning and accelerated discovery. *MRS Bull.* **43**, 664–669 (2018).
8. I. Tanaka, K. Rajan, and C. Wolverton: Data-centric science for materials innovation. *MRS Bull.* **43**, 659–663 (2018).
9. E. Kim, K. Huang, A. Saunders, A. McCallum, G. Ceder, and E. Olivetti: Materials synthesis insights from scientific literature via text extraction and machine learning. *Chem. Mater.* **29**, 9436–9444 (2017).
10. M. Krallinger, O. Rabal, A. Lourenço, J. Oyarzabal, and A. Valencia: Information retrieval and text mining technologies for chemistry. *Chem. Rev.* **117**, 7673–7761 (2017).
11. U.S. Government: *Materials Genome Initiative National Science and Technology Council Committee on Technology Subcommittee on the Materials Genome Initiative*; Whitehouse.Gov, June 2014.
12. A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K.A. Persson: Commentary: the Materials Project: a materials genome approach to accelerating materials innovation. *APL Mater.* **1**, 011002 (2013).
13. J.E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton: Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD). *JOM* **65**, 1501–1509 (2013).
14. S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R.H. Taylor, L.J. Nelson, G.L.W. Hart, S. Sanvito, M. Buongiorno-Nardelli, N. Mingo, and O. Levy: AFLOWLIB.ORG: a distributed materials properties repository from high-throughput ab initio calculations. *Comput. Mater. Sci.* **58**, 227–235 (2012).
15. S.S. Borysov, R.M. Geilhufe, and A.V. Balatsky: Organic materials database: an open-access online database for data mining. *PLoS ONE* **12**, e0171501 (2017).
16. G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky: AiiDA: automated interactive infrastructure and database for computational science. *Comput. Mater. Sci.* **111**, 218–230 (2016).
17. Y. Zhuo, A.M. Tehrani, A.O. Oliynyk, A.C. Duke, and J. Brgoch: Identifying an efficient, thermally robust inorganic phosphor host via machine learning. *Nat. Commun.* **9**, 4377 (2018).
18. P. Hall and N. Gill: *An Introduction to Machine Learning Interpretability*, 1st ed. (O'Reilly Media, Inc., Sebastopol, California, 2018).
19. Https://apps.webofknowledge.com/ (Clarivate Analytics, Philadelphia, PA).
20. D.A. Augusto and H.J.C. Barbosa: Symbolic regression via genetic programming. In *Proceedings - Brazilian Symposium on Neural Networks, SBRN*, Vol. 2000, Janua; IEEE Computer Society, 2000; pp. 173–178.
21. G.A.F. Seber and A.J. Lee: *Linear Regression Analysis* (Wiley-Interscience, Hoboken, New Jersey, 2003), pp. 557.
22. J.R. Koza: Genetic programming as a means for programming computers by natural selection. *Stat. Comput.* **4**, 87–112 (1994).
23. S. Forrest: Genetic algorithms: principles of natural selection applied to computation. *Science* **261**, 872–878 (1993).
24. B. Meredig and C. Wolverton: A hybrid computational-experimental approach for automated crystal structure solution. *Nat. Mater.* **12**, 123–127 (2013).
25. A.L.-S. Chua, N.A. Benedek, L. Chen, M.W. Finnis, and A.P. Sutton: A genetic algorithm for predicting the structures of interfaces in multicomponent systems. *Nat. Mater.* **9**, 418–422 (2010).
26. C.E. Mohn, S. Stølen, and W. Kob: Predicting the structure of alloys using genetic algorithms. *Mater. Manuf. Processes* **26**, 348–353 (2011).
27. I. Arnaldo, K. Krawiec, and U.-M. O'Reilly: Multiple regression genetic programming. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation - GECCO'14*; ACM Press, New York, NY, 2014; pp. 879–886.
28. J.A. Moore, R. Ma, A.G. Domel, and W.K. Liu: An efficient multiscale model of damping properties for filled elastomers with complex microstructures. *Compos. Part B: Eng.* **62**, 262–270 (2014).
29. M. Castelli, S. Silva, and L. Vanneschi: A C++ framework for geometric semantic genetic programming. *Genet. Program. Evol. Mach.* **16**, 73–81 (2015).
30. J.F. Miller, D. Job, and V.K. Vassilev: Principles in the evolutionary design of digital circuits part I. *Genet. Program. Evol. Mach.* **1**, 7–35 (2000).
31. H.I. Rad, J. Feng, and H. Iba: GP-RVM: Genetic Programing-based Symbolic Regression Using Relevance Vector Machine. (2018). arXiv:1806.02502v2
32. O. Giustolisi and D.A. Savic: Advances in data-driven analyses and modelling using EPR-MOGA. *J. Hydroinform.* **11**, 225 (2009).
33. T. McConaghy: *FFX: Fast, Scalable, Deterministic Symbolic Regression Technology* (Springer, New York, NY, 2011) pp. 235–260.
34. P. Orzechowski, W. La Cava, and J.H. Moore: Where are we now? In *Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO'18*; ACM Press, New York, NY, 2018; pp. 1183–1190. arXiv:1804.09331
35. I. Icke and J.C. Bongard, Improving genetic programming based symbolic regression using deterministic machine learning. In 2013 *IEEE Congress on Evolutionary Computation*; IEEE, 2013; pp. 1763–1770.
36. K. Krawiec: Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genet. Program. Evol. Mach.* **3**, 329–343 (2002).
37. Q. Lu, J. Ren, and Z. Wang: Using genetic programming with prior formula knowledge to solve symbolic regression problem. *Comput. Intell. Neurosci.* **2016**, 1 (2016).
38. L. Li, M. Fan, R. Singh, and P. Riley: Neural-guided symbolic regression with semantic prior. (2019). *arXiv preprint*: arXiv:1901.07714.
39. C.A. Tolman: The 16 and 18 electron rule in organometallic chemistry and homogeneous catalysis. *Chem. Soc. Rev.* **1**, 337–353 (1972).
40. B.W.H. Van Beest, G.J. Kramer, and R.A. Van Santen: Force fields for silicas and aluminophosphates based on ab initio calculations. *Phys. Rev. Lett.* **64**, 1955 (1990).
41. T. Yanai, D.P. Tew, and N.C. Handy: A new hybrid exchange–correlation functional using the coulomb-attenuating method (cam-b3lyp). *Chem. Phys. Lett.* **393**, 51–57 (2004).
42. M. Schmidt and H. Lipson: Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press

43. J. Gout, M. Quade, K. Shafi, R.K. Niven, and M. Abel: Synchronization control of oscillator networks using symbolic regression. *Nonlin. Dyn.* **91**, 1001–1021 (2018).
44. V. Arkov, C. Evans, P.J. Fleming, D.C. Hill, J.P. Norton, I. Pratt, D. Rees, and K. Rodríguez-Vázquez: System identification strategies applied to aircraft gas turbine engines. *Annu. Rev. Control* **24**, 67–81 (2000).
45. L. Berardi, O. Giustolisi, Z. Kapelan, and D.A. Savic: Development of pipe deterioration models for water distribution systems using EPR. *J. Hydroinform.* **10**, 113 (2008).
46. J. Bongard and H. Lipson: Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.* **104**, 9943–9948 (2007).
47. W. Cai, A. Pacheco-Vega, M. Sen, and K.T. Yang: Heat transfer correlations by symbolic regression. *Int. J. Heat Mass Transf.* **49**, 4352–4359 (2006).
48. B. Can and C. Heavey: Comparison of experimental designs for simulation-based symbolic regression of manufacturing systems. *Comput. Ind. Eng.* **61**, 447–462 (2011).
49. B. McKay, M. Willis, and G. Barton: Steadystate modelling of chemical process systems using genetic programming. *Comput. Chem. Eng.* **21**, 981–996 (1997).
50. W. La Cava, K. Danai, and L. Spector: Inference of compact nonlinear dynamic models by epigenetic local search. *Eng. Appl. Artif. Intell.* **55**, 292–306 (2016).
51. W. La Cava, K. Danai, L. Spector, P. Fleming, A. Wright, and M. Lackner: Automatic identification of wind turbine models using evolutionary multiobjective optimization. *Renew. Energy* **87**, 892–902 (2016).
52. S.-H. Chen and C.-H. Yeh: Toward a computable approach to the efficient market hypothesis: an application of genetic programming. *J. Econ. Dyn. Control* **21**, 1043–1063 (1997).
53. G.J. Gray, D.J. Murray-Smith, Y. Li, K.C. Sharman, and T. Weinbrenner: Nonlinear model structure identification using genetic programming. *Control Eng. Pract.* **6**, 1341–1352 (1998).
54. S.T. Khu, S.Y. Liong, V. Babovic, H. Madsen, and N. Muttil: Genetic programming and its application in real-time runoff forecasting. *J. Am. Water Resour. Assoc.* **37**, 439–451 (2001).
55. S.-Y. Liong, T.R. Gautam, S.T. Khu, V. Babovic, M. Keijzer, and N. Muttil: Genetic programming: a new paradigm in rainfall runoff modeling. *J. Am. Water Resour. Assoc.* **38**, 705–718 (2002).
56. M. Quade, M. Abel, K. Shafi, R.K. Niven, and B.R. Noack: Prediction of dynamical systems by symbolic regression. *Phys. Rev. E* **94**, 012214 (2016).
57. M.D. Schmidt, R.R. Vallabhajosyula, J.W. Jenkins, J.E. Hood, A.S. Soni, J.P. Wikswo, and H. Lipson: Automated refinement and inference of analytical models for metabolic networks. *Phys. Biol.* **8**, 055011 (2011).
58. K. Stanislawska, K. Krawiec, and Z.W. Kundzewicz: Modeling global temperature changes with genetic programming. *Comput. Math. Appl.* **64**, 3717–3728 (2012).
59. K. Uesaka and M. Kawamata: Synthesis of low-sensitivity second-order digital filters using genetic programming with automatically defined functions. *IEEE Signal Process. Lett.* **7**, 83–85 (2000).
60. R. Vyas, P. Goel, and S.S. Tambe, Genetic programming applications in chemical sciences and engineering. In *Handbook of Genetic Programming Applications*; Springer International Publishing, Cham, 2015; pp. 99–140.
61. W.B. Langdon and S.J. Barrett: Genetic programming in data mining for drug discovery. In *Evolutionary Computation in Data Mining*, Vol. 163; Springer-Verlag, Berlin/Heidelberg, 2005; pp. 211–235.
62. R. Vyas, P. Goel, M. Karthikeyan, S.S. Tambe, and B.D. Kulkarni: Pharmacokinetic modeling of Caco-2 cell permeability using genetic programming (GP) method. *Lett. Drug Des. Discov.* **11**, 1112–1118 (2014).
63. P. Barmpalexis, K. Kachrimanis, A. Tsakonas, and E. Georgarakis: Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation. *Chemom. Intell. Lab. Syst.* **107**, 75–82 (2011).
64. C.D. Muzny, M.L. Huber, and A.F. Kazakov: Correlation for the viscosity of normal hydrogen obtained from symbolic regression. *J. Chem. Eng. Data* **58**, 969–979 (2013).
65. A.A. Markov, M.V. Patrakeev, V.V. Kharton, Y.V. Pivak, I.A. Leonidov, and V.L. Kozhevnikov: Oxygen nonstoichiometry and ionic conductivity of $Sr_3Fe_{2-x}Sc_xO_{7-\delta}$. *Chem. Mater.* **19**, 3980–3987 (2007).
66. A. Nakamura and J.B. Wagner: *Defect Structure, Ionic Conductivity, and Diffusion in Yttria Stabilized Zirconia and Related Oxide Electrolytes with Fluorite Structure*, Technical Report.
67. L. Daza, C.M. Rangel, J. Baranda, M.T. Casais, M.J. Mart´ınez, and J.A. Alonso: Modified nickel oxides as cathode materials for MCFC. *J. Power Sources* **86**, 329–333 (2000).
68. M. Maslyaev, A. Hvatov, and A. Kalyuzhnaya, Data-driven PDE discovery with evolutionary approach. (2019). arXiv:1903.08011
69. S. Gaucel, M. Keijzer, E. Lutton, and A. Tonda, Learning dynamical systems using standard symbolic regression. In *Genetic Programming*, edited by M. Nicolau, K. Krawiec, M. I. Heywood, M. Castelli, P. García-Sánchez, J.J. Merelo, V.M. Rivas Santos, and K. Sim (Springer, Berlin/Heidelberg, 2014) pp. 25–36.
70. M. Schmidt and H. Lipson: Symbolic regression of implicit equations. *Genet. Program. Theory Pract.* **7**, 73–85 (2009).
71. U. von Barth and L. Hedin: A local exchange correlation potential for the spin polarized case: I. *J. Phys. C: Solid State Phys.* **5**, 1629 (1972).
72. The Minerals, Metals & Materials Society: *Modeling Across Scales: A Roadmapping Study for Connecting Materials Models and Simulations Across Length and Time Scales*, Technical Report (2015), 2015.
73. A. Yadollahi, N. Shamsaei, S.M. Thompson, and D.W. Seely: Effects of process time interval and heat treatment on the mechanical and microstructural properties of direct laser deposited 316L stainless steel. *Mater. Sci. Eng. A* **644**, 171–183 (2015).
74. L. Ward and C. Wolverton: Atomistic calculations and materials informatics: a review. *Curr. Opin. Solid State Mater. Sci.* **21**, 167–176 (2017).
75. L.M. Ghiringhelli, J. Vybiral, S.V. Levchenko, C. Draxl, and M. Scheffler: Big data of materials science: Critical role of the descriptor. *Phys. Rev. Lett.* **114**, 105503 (2015). doi: 10.1103/PhysRevLett.114.105503
76. L.M. Ghiringhelli, J. Vybiral, E. Ahmetcik, R. Ouyang, S.V. Levchenko, C. Draxl, and M. Scheffler: Learning physical descriptors for materials science by compressed sensing. *New J. Phys.* **19**, 023017 (2017).
77. R. Ouyang, S. Curtarolo, E. Ahmetcik, M. Scheffler, and L.M. Ghiringhelli: SISSO: a compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates. *Phys. Rev. Mater.* **2**, 083802 (2018).
78. G.N. Vanderplaats: Numerical Optimization Techniques for Engineering Design (Vanderplaats Research and Development, Inc., Novi, MI), 2005.
79. M. Shimada, H. Kokawa, Z.J. Wang, Y.S. Sato, and I. Karibe: Optimization of grain boundary character distribution for intergranular corrosion resistant 304 stainless steel by twin-induced grain boundary engineering. *Acta Mater.* **50**, 2331–2341 (2002).
80. B.F. Decker and D. Harker: Activation energy for recrystallization in rolled copper. *JOM* **2**, 887–890 (1950).
81. Trevor Stephens: Genetic Programming in Python, with a scikit-learn inspired API: gplearn, 2016.
82. G. Gou, I. Grinberg, A.M. Rappe, and J.M. Rondinelli: Lattice normal modes and electronic properties of the correlated metal $LaNiO_3$. *Phys. Rev. B* **84**, 144101 (2011).
83. H. Yu, J. Young, H. Wu, W. Zhang, J.M. Rondinelli, and P. Shiv Halasyamani: Electronic, crystal chemistry, and nonlinear optical property relationships in the dugganite $A_3B_3CD_2O_{14}$ family. *J. Am. Chem. Soc.* **138**, 4984–4989 (2016).

https://doi.org/10.1557/mrc.2019.85 Published online by Cambridge University Press