A Deterministic and Symbolic Regression Hybrid Applied to Resting-State fMRI data

Ilknur Icke[1,3], Nicholas A. Allgaier[2,3], Christopher M. Danforth[2,3], Robert A. Whelan[4], Hugh P. Garavan[4], Joshua C. Bongard[1,3] and IMAGEN Consortium[5]

Abstract Symbolic regression (SR) is one the most popular applications of genetic programming (GP) and an attractive alternative to the standard deterministic regression approaches due to its flexibility in generating free-form mathematical models from observed data without any domain knowledge. However, GP suffers from various issues hindering the applicability of the technique to real-life problems. In this paper, we show that a hybrid deterministic regression (DR) / genetic programming based symbolic regression (GP-SR) algorithm outperforms GP-SR alone on a brain imaging dataset.

Key words: symbolic regression, hybrid algorithms, regularization, resting-state fMRI

1 Introduction

Despite various success stories (Koza (2010); Dubcakova (2011)) and claims that they will one day *replace scientists* (ACM (2011)), GP-SR applications (or any evolutionary computation based approach in general) have not yet been widely accepted as standard tools for data science. The resistance is not without a cause: GP suffers from various issues that hinder its applicability to many real-world machine learning problems (O'Neill et al (2010)). The theoretical foundations of GP are not as well developed as many of the standard machine learning (ML) algorithms. Moreover, scalability is a very challenging problem due to the trial-and-error nature of GP. Some efforts to make GP more scalable via GPUs and cloud computing have

[1] Department of Computer Science, University of Vermont
[2] Department of Mathematics and Statistics, University of Vermont
[3]Vermont Complex Systems Center
[4]Department of Psychiatry, University of Vermont
[5]http://www.imagen-europe.com/en/consortium.php

been reported (such as in GPG (2013) and Sherry et al (2011)). It is our belief that, if GP-SR is to be a trustable data science tool, it needs to take advantage of the recent developments in ML as well as parallel and distributed computing techniques.

The idea of studying evolutionary computation techniques from a ML perspective has previously been explored. The behavior of GP has been studied in terms of learning theory in Amil et al (2009) and Castelli et al (2011) among others. The learnable evolution model (LEM) proposed in Michalski (2000) is a technique to guide evolutionary processes with ML by creating hypotheses characterizing the differences between high performing and low performing individuals in the population.

Recently, it has been suggested that GP might not be the best option for SR and that stochasticity was not necessarily a virtue (McConaghy (2011)). The author proposed a deterministic basis function expansion method used in conjunction with a state-of-the-art deterministic ML regression algorithm as an alternative to GP-SR. This algorithm, Fast Function Extraction (FFX), was reported to outperform GP-SR on a number of real-world regression problems with dimensionalities ranging from 13 to 1468. This paper shares the same basic ideology: SR should not stray from the well-established techniques of ML. However, we argue that abandoning GP altogether may be premature. Instead, we propose to hybridize the two approaches.

Our long term hypothesis is that a hybrid algorithm combining symbolic regression with state-of-the-art deterministic regression out-competes both symbolic regression alone and deterministic regression alone. Our short-term hypothesis, which is covered in this paper, suggests that hybridizing symbolic regression with state-of-the-art deterministic regression out-competes symbolic regression (GP-SR) alone. In this paper, we explore one way to incorporate a deterministic ML method into GP-SR in order to improve GP-SR and demonstrate the utility of this hybrid algorithm on a brain imaging dataset.

The functional magnetic resonance imaging (fMRI) is a non-invasive way of monitoring the activation of various brain regions while the subject lays in the MRI scanner. In the case of resting-state fMRI, the subject is not given any external stimuli. It is possible to model the activation in one brain region in terms of activations of other brain regions using a regression algorithm. By identifying the most predictive regions for each target region, one can build a functional brain connectivity network (Poldrack and Nichols (2011)). For successful identification of the brain connectivity network, the performance of the regression algorithm is very important, especially when a large number of brain regions are to be modeled simultaneously. In this paper, we focus on the first step, namely the regression algorithm that identifies the concurrent relationships between the brain regions. The subsequent steps of building functional connectivity networks from the fMRI resting-state data will not be discussed here.

The organization of this paper is as follows: sections 2, 3 and 4 discuss the problem domain, technical background and related work respectively. Our proposed algorithm to hybridize the GP-SR and deterministic ML approaches is detailed in section 5. Experimental results are presented and discussed in section 6. Finally, section 7 discusses our conclusions and future work.

2 Resting-state fMRI

Direct measurement of electrical neuron activity is not easily accomplished. Electroencephalograms (EEGs) can be recorded by placing electrodes on the surface of subject's head, and are adequate for measuring an averaged whole-brain electrical signal. However, targeting specific regions of the brain, especially at the resolution of a neuron, is not practical with current technology. This difficulty in direct measurement is the motivation for neuroimaging.

One of the most recent methods of neuroimaging, developed about 20 years ago, is functional MRI. Time series of 3-D images of the brain captured by magnetic field indicate changes in blood flow, called the *hemodynamic response*, that correlate with neuronal activity. The reason for this correlation is that active neurons require more glucose and oxygen than inactive ones, and thus neuronal activity is said to be blood-oxygen-level dependent, giving rise to the name BOLD for the fMRI signal. It is important to note that there is some danger in using the BOLD signal as a proxy for neuronal activity. Many other factors contribute to increased blood flow in the brain, including contraction/dilation of vasculature, the structure of that vasculature (which can vary considerably with demographic), and even subject-specific general health and behavioral habits. Some techniques for handling their conflation in the BOLD signal were introduced, however, the discussion of these techniques is beyond the scope of this overview.

Very high image resolutions are attainable with fMRI. For this work, the data collected fall on a grid with voxels (3D pixels) approximately 3mm on a side, corresponding to about 50,000 voxels in a single whole-brain image. Typically, a region of interest (ROI) in the brain is represented by an averaged BOLD signal over some number of contiguous voxels, in order to smooth the noise present in a single voxel. Each of the 52 ROI's chosen for this study consists of a roughly spherical group of about 100 voxels, (with a radius of 3 voxels, or almost 1cm). Their locations were chosen based on work from Laird et al (2011), in which statistical analysis across thousands of previous imaging studies (both stimulus/task-based and resting-state) was used to identify networks of brain regions that tend to activate together. Figure 1 shows one such network, and spheres chosen to represent the ROI's in that network. Figure 2 shows a selection of equally spaced axial cross sections (top panel) that includes most of the 52 ROI's.

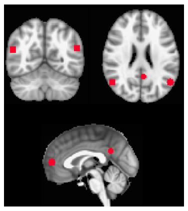

Fig. 1 Four selected ROI spheres for a resting-state network with centers corresponding to the highest z-statistics indicating the likelihood of co-activation across all studies (derived from independent component analysis) reported in Laird et al (2011)

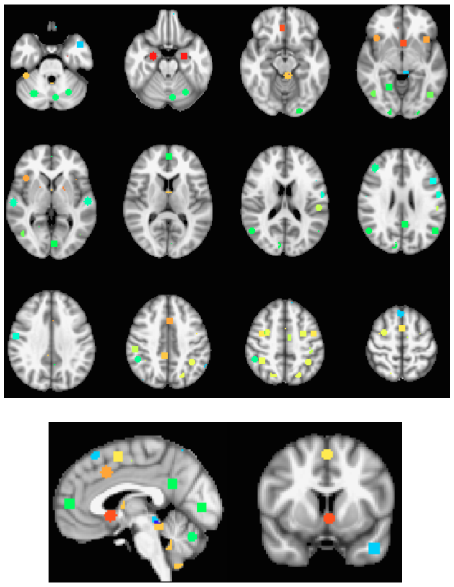

Fig. 2 (Top panel) Axial cross sections showing the majority of the 52 ROI's, colored by network, where the top left image is from a slice near the bottom of the brain, and the bottom right is near the top of the brain. (Bottom panel) Shown are a central sagittal cross section (left) and a coronal section from the front half of the brain (right)

The data for the present study come from fMRI scans of a group of adolescents who were asked to to keep their eyes closed while in the scanner, but were presented with no other task or stimulus. The term used for this type of data is "resting-state", though the brain is in no way at rest. In fact, the regions that activate during various task and stimulus-based studies also activate intermittently during the resting state, which suggests that some meaningful interactions are occurring and are ripe for analysis. Resting-state scans pose a challenge to standard fMRI analysis techniques, which typically involve regression of the BOLD signal in terms of a stimulus or task signal, (e.g. a square wave representing when the stimulus was being presented, or when the task was being performed). The application of deterministic regression, GP-SR, or a hybrid of the two, to model the activity in ROI's as functions *of one another* represents a potentially fruitful analytical strategy. Ideally, selection of ROI's would be avoided as well, and the input to the algorithm would be a much larger set of randomly selected regions within the brain. The very high dimensionality in this case provides an impetus for the hybrid technique, given the difficulty in applying GP-SR alone to such problems.

3 Feature Extraction and the Regularization Approach

Data dimensionality poses a great challenge for both the numeric and symbolic regression algorithm. As the number of predictors increases, it becomes more difficult to identify the informative predictors and to build accurate models. This problem has been well-studied in ML. The task of seeking the best representation for a given dataset in order to optimize the performance of a ML algorithm is known as *feature extraction* (Guyon and Elisseeff (2003)). Feature extraction can be seen as a sequential process, where new features are first constructed from the input variables and then the most informative ones are selected among the constructed features (figure 3). *Feature construction* may or may not increase data dimensionality. If the input variables are suspected to have interactions, it is generally the practice to create additional features via non-linear expansion (such as $x_1 * x_2$).

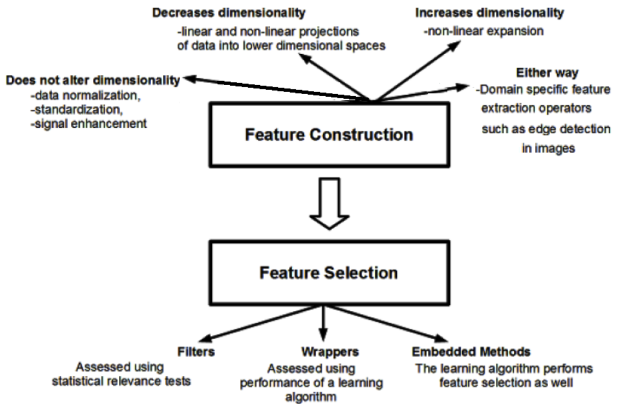
Decreases dimensionality
-linear and non-linear projections of data into lower dimensional spaces
Increases dimensionality
-non-linear expansion
Does not alter dimensionality
-data normalization,
-standardization,
-signal enhancement
Either way
-Domain specific feature extraction operators such as edge detection in images
Feature Construction
Feature Selection
Filters
Assessed using statistical relevance tests
Wrappers
Assessed using performance of a learning algorithm
Embedded Methods
The learning algorithm performs feature selection as well

Fig. 3 Feature extraction as a sequential process of creating features from the input variables and then selecting the most informative features.

As for *feature selection*, the simplest approach is to rank the features with respect to how well they correlate with the predicted variable. This method risks eliminating such features that might not be informative by themselves but may be very informative together. Subset selection methods aim to address this issue by considering a subset of features together. These techniques are divided into three main groups: filters, wrappers and embedded methods. *Filters* are pre-processing techniques that, independent of the learning algorithm, select a subset of variables with respect to some criteria such as mutual information. The *wrapper* techniques consider the learning algorithm as a black-box and select the set of features that optimize the performance of the learner. The feature subsets are generated either by forward selection, which gradually adds features or backward elimination, which starts with the whole set of features and eliminates least informative ones. The wrapper approach is computationally expensive as the learning algorithm needs to be executed many times. The *embedded* methods incorporate feature selection into the learning algorithm itself. The decision tree algorithms are the earliest examples of embedded methods. More recent embedded methods utilize the *regularization* technique, a version of which will be used in this work.

Within the context of the linear regression problem, regularization imposes additional constraints on the coefficients in order to reduce overfitting. In linear regression, given a multivariate dataset $\mathbf{X}_{[M \times N]} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}\}$, a matrix of observations, the response variable $\mathbf{Y}$ is defined as:

$$Y = f(X) = \beta_0 + \sum_{j=1}^{N} \beta_j * \mathbf{x_j}$$

The coefficients are computed via the least squares estimation by minimizing the residual sum of squares:

$$RSS = min_\beta (\sum_{i=1}^{M} y_i - \beta_0 - \sum_{j=1}^{N} \beta_j * x_{ij})^2$$

Since the parameters are computed on the training data, overfitting may occur. Overfitting manifests itself as extreme coefficient values. Therefore, an additional constraint on the coefficients is imposed in order to tame the coefficients ($\sum_{j=1}^{N} ||\beta_j||_1 \leq t$). This algorithm, known as lasso (least absolute shrinkage and selection operator), shrinks the coefficients and also performs feature elimination since the l_1-norm promotes sparsity. Therefore, the coefficients of uninformative features will be close to 0. An l_2-norm constraint is also possible and it is called ridge regression. Ridge regression has the effect of grouping the correlated variables so that they are included in the model together (Tibshirani (1996); Murphy (2012)). The elastic net approach (Zou and Hastie (2005); Jerome Friedman and Tibshirani (2010)) is a hybrid of lasso and ridge regression and is formulated as:

$$Y = f(X) = \beta_0 + \sum_{j=1}^{N} \beta_j * \mathbf{x_j} + \lambda_2 ||\beta||_2^2 + \lambda_1 ||\beta||_1$$

Generally, λ_1, λ_2 are balanced by defining one single parameter ($0 \leq \alpha \leq 1$) that is called the mixing parameter. At the extreme values of α, the elastic net behaves like purely lasso or purely ridge regression. A very large value of λ forces all βs to be 0. As λ is relaxed, the coefficients start to take nonzero values. This sweep of λ values can be visualized as a regularization path (figure 4). The algorithm is named an elastic net since the "regularization path is like a stretchable net that retains all the big fish" (Zou and Hastie (2005)).

This basic linear regression algorithm applies to generalized linear models (GLM) of the form:

$$Y = f(X) = \beta_0 + \sum_{j=1}^{N} \beta_j * b_j(X)$$

where $b_j(X)$ are nonlinear basis functions applied to the input variables in order to construct new features.

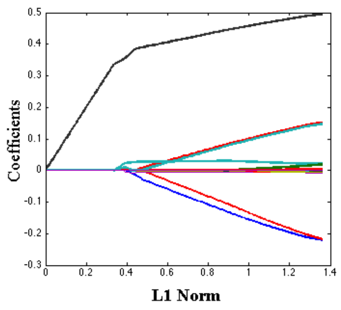

Fig. 4 Regularization path for an elastic net on a 10-dimensional dataset. For each λ, the l_1-norm of the coefficients vector versus individual coefficient values are shown. Each line traces the change of coefficient values for one variable. At the beginning, no features are selected. Gradually, more features are incorporated into the models (coefficients > 0).

4 Related Work

GP-SR inherently performs feature selection when it finds sufficiently accurate models; any feature that does not appear in the evolved expression can be considered uninformative. However, when data dimensionality is high, the search space grows exponentially, making it difficult for GP-SR to find good solutions. The issue of feature selection in GP-SR algorithms has been studied by various researchers. Koza's automatically defined functions (ADF) (Koza (1992)) can be seen as a feature extraction method within the context of SR. A pareto-GP based variable selection scheme was proposed in Smits et al (2005). In McRee (2010), permuation tests were introduced in GP-SR to discriminate between informative and uninformative variables. In Stijven et al (2011), feature selection capabilities of GP-SR and random forests were compared. The authors report that when it finds an accurate model, GP-SR captures the important features that are missed by the random forests algorithm.

The regularization approach has been applied to GP-SR in Nikolaev and Iba (2001) for polynomial functional form discovery. The authors incorporate a function smoothness term into the fitness function as a way to decrease overfitting.

The Fast Function Extraction (FFX) algorithm reported in McConaghy (2011) employs a nonlinear basis function expansion method that creates new features via unary and binary interactions of the input variables. The algorithm does not employ GP-SR to construct the features or the models. The new features are created in a deterministic manner and passed to the elastic net algorithm for model building.

The algorithm generates multiple models for the λs on the regularization path. The non-dominated set of these models with respect to accuracy versus complexity are identified as the final models.

The difference of our proposed technique is that we perform feature extraction using an efficient deterministic ML algorithm such as in FFX, and pass the features to GP-SR for model building. By taking advantage of the state-of-the-art deterministic regression methods, our algorithm aims to ease the burden of feature extraction on GP-SR and help it excel in model building, especially when higher order interactions between the variables exist. We have shown on a synthetic data suite with up to fourth order variable interactions that our technique significantly improves the performance of GP-SR as the data dimensionality increases (Icke and Bongard (2013)).

5 Improving GP-SR using Deterministic Machine Learning

The technique we propose in this paper has been largely inspired by the FFX algorithm reported in McConaghy (2011). The author mainly suggested eliminating GP for the symbolic regression problem in favor of a deterministic method for augmenting the dataset with polynomial features and then using a state-of-the-art machine learning algorithm (elastic net) for model building. In this paper, we propose to hybridize GP with the deterministic ML techniques so as to take advantage of the strengths of both approaches to solve symbolic regression problems more accurately and efficiently in comparison to either technique alone. The outline of the general idea behind FFX is presented in algorithm 1 (for a detailed description of the FFX algorithm, see McConaghy (2011)).

Algorithm 1: Our variant of the basic FFX algorithm

```
   Input: V={v_1,v_2, ..., v_N}
   Output: The set of non-dominated evolved models based on validation data error-model
           complexity (number of bases) trade-off
1  [ bases, expandedTrainingDataset] = basisFunctionExtraction(trainingDataset)
2  models={}
3  foreach $\alpha \in (0, 0.05, 0.1, ..., 1)$ do
4      models = models $\bigcup$ glmnetfit(variables,trainingDataset)
5      nonDominatedModels =
6      extractParetoFrontier(models,expandedValidationDataset)
7      models=nonDominatedModels
8      models = models $\bigcup$ glmnetfit(bases,expandedTrainingDataset)
9      nonDominatedModels =
10     extractParetoFrontier(models,expandedValidationDataset)
       models=nonDominatedModels
11 end
```

The algorithm consists of three stages: feature construction, model building and model selection. The feature construction stage creates new features by applying binary nonlinear interactions (basis functions) and augmenting the original dataset (algorithm 2). It is clear that, with only the unary and binary features, the FFX algorithm will not accurately model data generated by a system with third or higher order interactions. Instead, it will find a quadratic approximation based on the unary and binary features which might increase the complexity of the expression in terms of the number of bases. It is possible to expand FFX beyond binary interactions; however, this would increase the number of constructed features exponentially. In this paper, we considered only unary and binary features as in McConaghy (2011).

The model building stage utilizes the coordinate descent elastic net algorithm (glmnet, line 4 of algorithm 1) that was proposed in Jerome Friedman and Tibshirani (2010). As it is shown in figure 4, for each value of λ, one can build an expression using the corresponding coefficients. Therefore, the model building stage returns multiple expressions containing different numbers of basis functions. Note that the models are built on the training data. At the model selection stage, the non-dominated set of models with respect to error on validation data versus expression complexity (the number of basis functions or bases for short) are identified.

Algorithm 2: *basisFunctionExtraction* : Polynomial basis function generation as new features form the observed data

```
Input: V={$v_1$,$v_2$, ..., $v_N$}
Output: Expanded Dataset: $V_e$={$v_{e1}$,$v_{e2}$, ..., $v_{eM}$ }
//Generate unary bases
foreach $v_1,v_2, ..., v_N$ do
    unaryBases = unaryBases $\bigcup$ $v_i$
    foreach $exp_j$ do
        unaryBases = unaryBases $\bigcup$ $v_i{}^{exp_j}$
    end
    foreach $unaryOperator_k$ do
        unaryBases = unaryBases $\bigcup$ unaryOperator$_k$($v_i$)
    end
end
//Generate binary bases
foreach $u_i \in unaryBases$ do
    foreach $u_j \in unaryBases$ do
        binaryBases= binaryBases $\bigcup$ $u_i$*$u_j$
    end
end
```

Our hybrid approach takes advantage of the basic FFX idea in order to identify the most important building blocks that might help GP-SR build accurate models. The method is outlined in algorithm 3.

After one run of the FFX algorithm, the most useful basis functions are extracted from the set of models by computing a histogram of how frequently each basis function uniquely appears in the expressions (figure 5). The number of basis functions to

Algorithm 3: The hybrid FFX/GP-SR algorithm

```
Input: V={v_1,v_2, ..., v_N}
Output: One best model with respect to the validation data error and complexity
1 nonDominatedModels = ffx(trainingDataset)
2 bases = identifyUsefulBasisFunctions (nonDominatedModels,
3                                       validationDataset)
4 newDataset=createNewDataset(bases)
5 bestModel = GP-SR(newDataset)
```

be selected can be determined in a number of ways. In this paper, we chose the top N basis functions, where N is the number of original input variables. This choice was made in order to provide both the GP-SR and the hybrid FFX/GP-SR algorithms the same amount of data in terms of rows and columns. Before the hybrid algorithm runs, a new dataset with at most N features are computed using the selected basis functions and GP-SR is run on this new dataset instead of the original dataset.

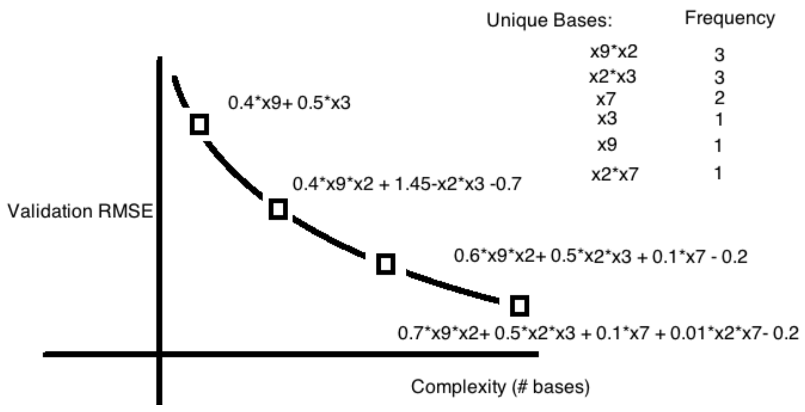
Unique Bases:
Frequency
x9*x2 3
x2*x3 3
x7 2
x3 1
x9 1
x2*x7 1
0.4*x9+ 0.5*x3
0.4*x9*x2 + 1.45-x2*x3 -0.7
0.6*x9*x2+ 0.5*x2*x3 + 0.1*x7 - 0.2
0.7*x9*x2+ 0.5*x2*x3 + 0.1*x7 + 0.01*x2*x7- 0.2
Validation RMSE
Complexity (# bases)

Fig. 5 Determination of the most useful features using validation data based on the models built by the elastic net on training data

6 Experimental Results

In this section, we first outline the procedure we followed in preparing the datasets and the parameters for each algorithm. Then, we compare the performance of our proposed algorithm to the plain GP-SR implementation on the resting-state fMRI data from a number of subjects. For each subject, the dataset contains 187 observations (rows) and 52 brain regions (columns). The data points for each brain region were divided into three non-overlapping and interleaving partitions for training (113

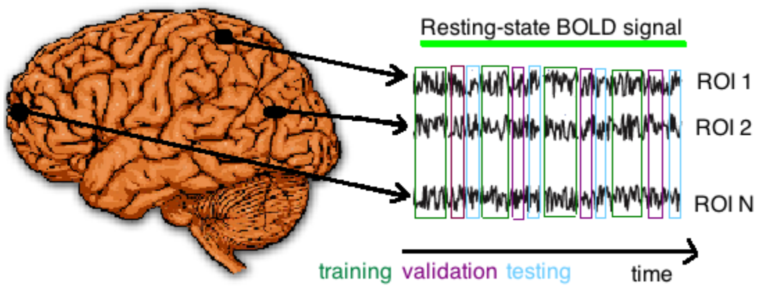

Fig. 6 Data partitioning: For each region of interest (ROI), the data points are partitioned into non-overlapping sets of training, validation and test points. Every fifth point is in the test partition and every fourth point is in the validation partition.

observations), validation (37 observations) and testing (37 observations) purposes (figure 6). The hybrid FFX/GP-SR algorithm performs the initial modeling on the training partition, and the best models are selected based on validation set performance. Then, the set of most useful basis functions are selected as described in the previous section. The test partitions are never used in model generation or selection. Finally, performances of the GP-SR and FFX/GP-SR hybrid are reported based on the errors on the test partitions.

We implemented our GP-based Symbolic Regression application using the GPTIPS Matlab package downloaded from Searson (2013). Our version of the FFX algorithm and FFX/GP-SR algorithms were also implemented in Matlab using the glmnet package downloaded from Jiang (2013) and the GPTIPS package. All experiments were run on a cluster computing environment. Unless otherwise specified, all FFX and GP-SR runs were performed using the default parameters specified in tables 1 and 2.

Parameter	Value
Basis Function Expansion	Exponents : 1 Interactions : Unary, Binary Operators : { }
Elastic Net	$\alpha : \{0, 0.05, 0.1, ..., 1\}$ λ : 100 λ values calculated by glmfit based on α Maximum basis functions allowed : 250
Model Selection	Non-dominated models with respect to validation data error versus number of bases

Table 1 Default FFX-variant parameters

For each subject, we ran the FFX algorithm for each region (column) as the target variable and all other columns as the predictors. For the GP-SR algorithm, we ran the GP-SR thirty times for each region as the target variable. Each GP-SR run was given a run-time budget of 5 minutes. For the FFX/GP-SR hybrid, we ran the FFX algorithm once for each region, extracted basis functions to recreate the dataset and ran the GP-SR thirty times for each region for a run-time budget of 5 minutes.

Parameter	Value
Representation	GPTIPS Searson (2013) Multigene syntax tree Number of genes: 1 Maximum tree depth: initially 3, 12 afterwards
Population Size Runtime Budget	2000 5 minutes
Selection Tournament Size	Lexicographic tournament selection 7
Crossover Operator Crossover Probability	Subtree crossover 0.85
Mutation Operator Mutation Probability	Subtree mutation 0.1
Reproduction Probability	0.05
Building Blocks	Operators: $\{+, -, *, protected/\}$ Terminal Symbols: $\{x_1, ..., x_N\}$
Fitness	RMSE: $\sqrt{\frac{1}{N}\sum(y-\hat{y})^2}$
Elitism	Keep 1 best individual

Table 2 Default GP-SR parameters

The comparisons between GP-SR and our hybrid FFX/GP-SR methods on subject 1 are presented in figure 7. For each region, we compare the test set errors of the best models found by the thirty runs of each algorithm. The winning algorithm for each region is shown on top of each plot in parentheses. The winner is determined based on a Wilcoxon rank sum left-tailed test, $\alpha = 0.05$. In the cases where no winner is specified, the algorithms were not found to be statistically different in terms of test set performance.

We repeated the experiments for a total of 7 subjects. The summary of the results are provided in table 3. Based on 364 distinct regression tasks, the results clearly indicate that the hybrid algorithm has a significantly better chance than random coin flipping in performing at least as well as the GP-SR algorithm. Moreover, the hybrid algorithm is around 2 times or more likely to outperform the plain GP-SR than to be outperformed by it.

Subject	%Hybrid wins	%GP-SR wins	%No difference	%Hybrid at least as good as GP-SR (100-GP-SR wins)
1	42.3%	23%	34.7%	77%
2	61.5%	15.4%	23.1%	84.6 %
3	46.1%	23.1%	30. 8%	76.9 %
4	32.7%	17.3%	50%	82.7%
5	36.6%	19.2%	44.2%	80.8%
6	44.2%	9.6%	46.2 %	90.4%
7	44.2%	19.2%	36.6%	80.8%
Overall	44%	18.1%	37.9%	81.9%

Table 3 Summary of results on a number of subjects (over all 52 regions per subject). The probability of the hybrid algorithm not being worse than the GP-SR is significantly higher than random chance (approximately 82%) over 52*7=364 regression runs

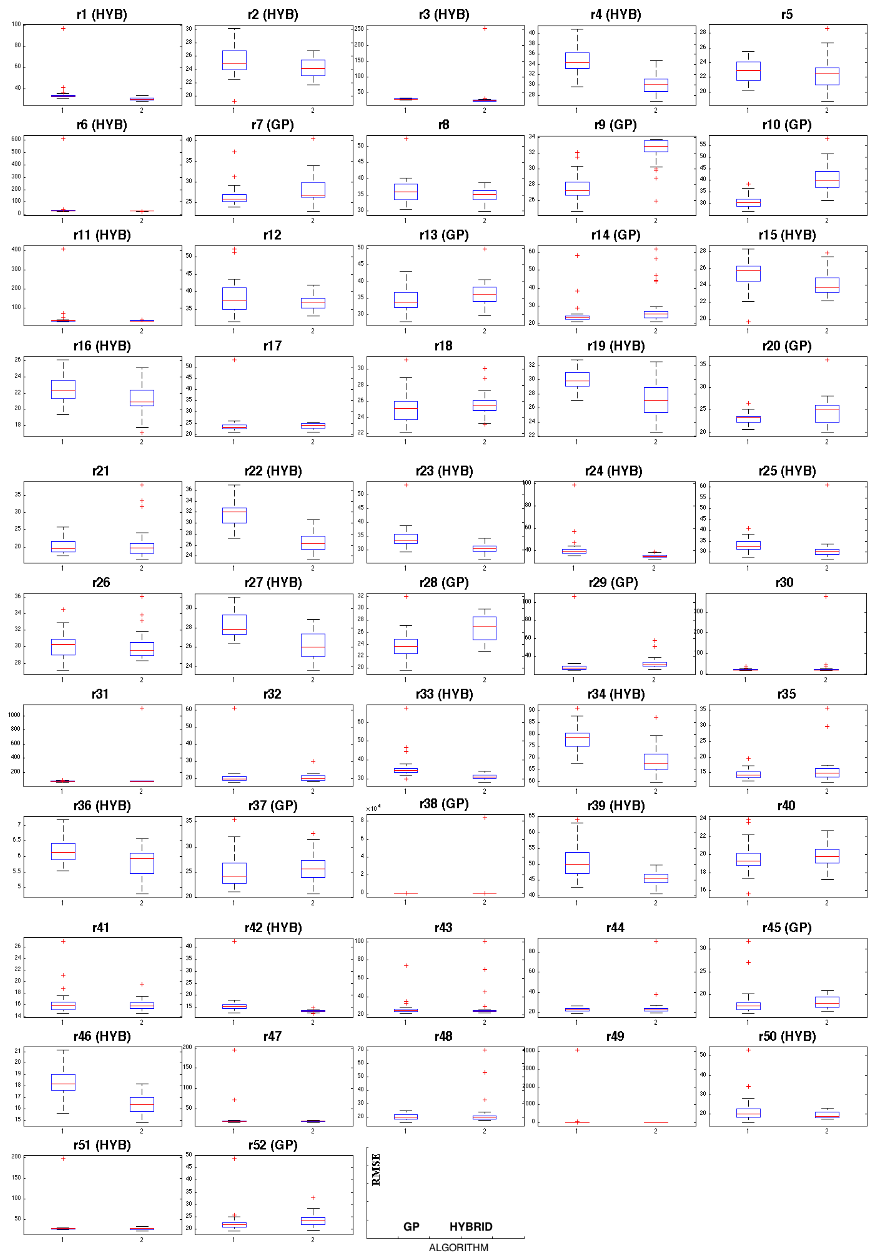

Fig. 7 Comparison of the hybrid and GP-SR on subject 1 based on testing partition RMSE. Hybrid wins 22 out of the 52 regions (42.3%). GP wins 12/52 (23%), and no performance difference in 18/52 (34.7%) regions. At the expense of an additional 90 seconds for the one-time run of the FFX-variant, the hybrid algorithm has a 77% chance of performing at least as good as the GP-SR.

We also compared all three approaches in order to see if it would just suffice to utilize FFX and not resort to the GP-SR techniques at all. Figures 8, 9 show the non-dominated sets of models generated by all three algorithms combined into one single Pareto front. For each brain region, we apply the final models from each algorithm to the testing partition and record error versus expression complexity (number of nodes) and then compute the non-dominated set. For the GP-SR and hybrid methods, there are 30 final models each, while for the FFX-variant, the number of final models varies. Each plot shows the non-dominated models marked with respect to the algorithms that generated them. As it is evident, the FFX-variant does not dominate the GP-SR based approaches. Specifically, for some brain regions, the solutions found by the FFX-variant are significantly more complex for a relatively small decrease in error. Therefore, we are convinced that we should not completely give up on GP based approaches for symbolic regression.

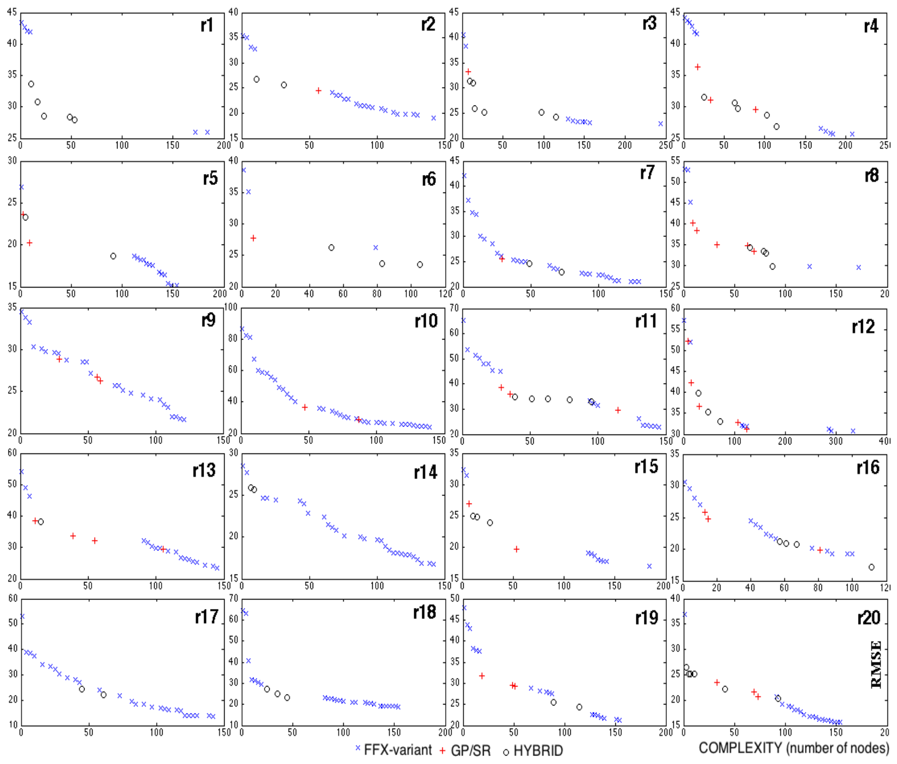

Fig. 8 Comparison of all 3 approaches on subject 1 for regions 1-20. The plots show combined non-dominated models with respect to complexity versus error on testing data.

Finally, we inspected those brain regions for which GP-SR outperformed the hybrid approach. For instance, for subject 1, region 9 (figure 7), we identified that one region that was found to be predictive of region 9 by the GP-SR algorithm

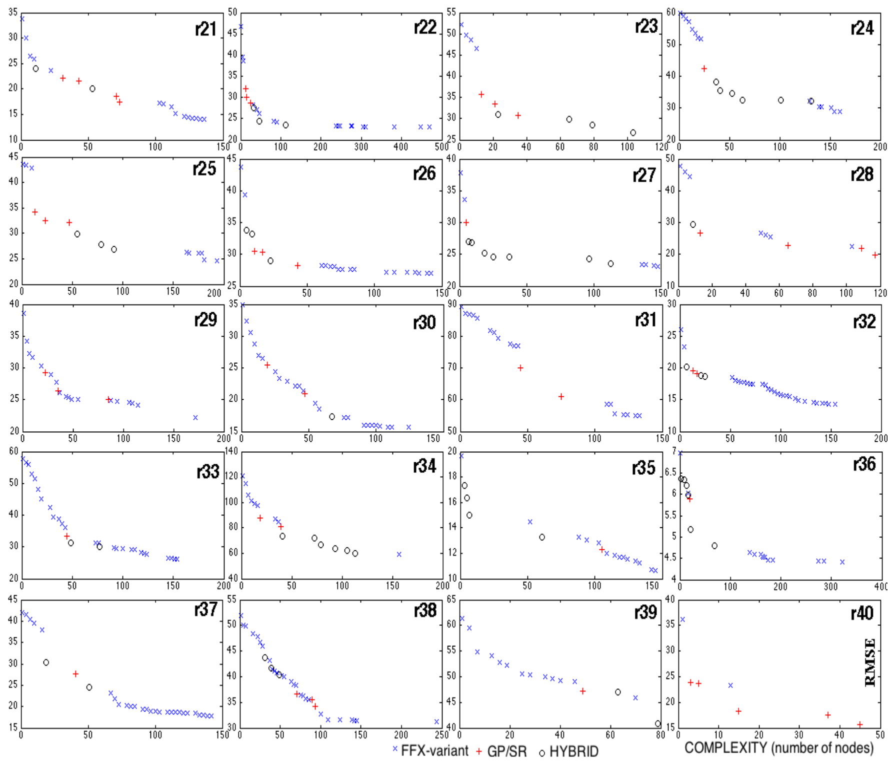

Fig. 9 Comparison of all 3 approaches on subject 1 for regions 21-40. The plots show combined non-dominated models with respect to complexity versus error on testing data.

was totally omitted from the models built by the FFX-variant. Therefore, it was not possible for the hybrid algorithm to incorporate that predictive variable into the models since the input to the hybrid algorithm is determined by the outcome of the FFX-variant.

In summary, our experimental results show that hybridizing GP-SR with a state-of-the-art deterministic ML technique is significantly more likely to improve the performance of GP-SR. We also show evidence that the hybrid approach works better than the deterministic ML technique on a considerable number of cases strengthening our claim that abandoning GP altogether for SR is premature.

7 Conclusions

Despite all the various efforts to improve it, GP-SR is yet to be accepted as a standard data analysis tool. In this paper, we argued that the resistance from the ML community was not totally unfounded. First of all, theoretical underpinnings of

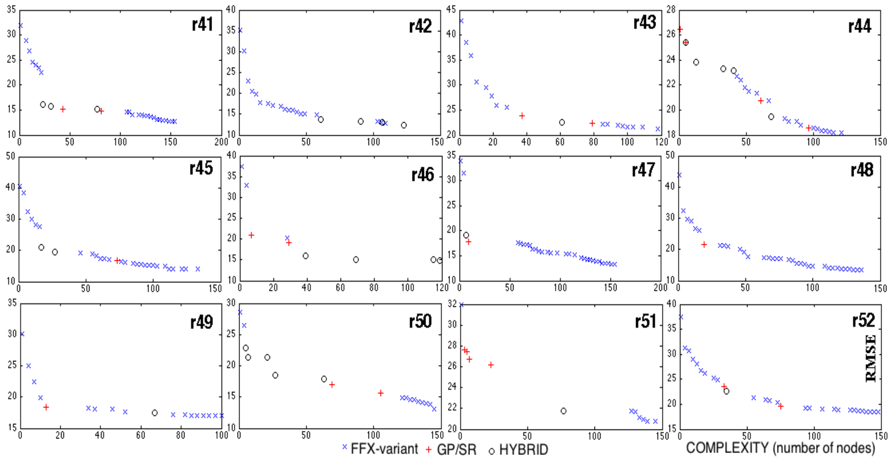

Fig. 10 Comparison of all 3 approaches on subject 1 for regions 41-52. The plots show combined non-dominated models with respect to complexity versus error on testing data.

the GP-SR such as convergence proofs are not well established. GP-SR is computationally more expensive compared to most deterministic ML algorithms due to its stochastic nature. Even though many intuitive strategies might be built into the algorithm, there is no guarantee that optimal data models will emerge at the end of the run. More importantly, despite all the success stories, GP-SR techniques do not necessarily outperform the state-of-the-art deterministic ML methods, especially on high dimensional problems. On the other hand, the strength of GP-SR is in its model-free nature which makes enables the algorithm to discover more accurate, more interpretable and novel models in comparison to a deterministic approach. In summary, it is our belief that stochasticity can be a virtue for SR if it is directed intelligently. To this end, we showed here that it was possible to create synergy between the deterministic and GP-SR approaches by hybridizing them. The technique presented in this paper is just one way out of many possible options to combine the GP-SR and deterministic techniques for regression problems. Here, we incorporated building blocks extracted by the deterministic algorithm into the GP-SR algorithm. Application of the algorithm to a real-world dataset from the neuroscience field gave encouraging results. Our current work focuses on improving this hybrid algorithm and validating it on other datasets. We are also working on devising a more rigorous scheme to compare all three approaches (the FFX-variant, GP-SR and the hybrid) and investigate the situations where each method would succeed or fail.

Acknowledgements This research was partially supported by a DARPA grant #FA8650-11-1-7155. We thank the IMAGEN consortium for providing us with the resting-state fMRI dataset. The authors also acknowledge 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.

References

(2011) Robot biologist solves complex problem from scratch. http://cacm.acm.org/careers/136345-robot-biologist-solves-complex-problem-from-scratch/fulltext

(2013) Genetic programming on general purpose graphics processing units. http://www.gpgpgpu.com/

Amil NM, Bredeche N, Gagné C, Gelly S, Schoenauer M, Teytaud O (2009) A statistical learning perspective of genetic programming. In: Vanneschi L, Gustafson S, Moraglio A, De Falco I, Ebner M (eds) Proceedings of the 12th European Conference on Genetic Programming, EuroGP 2009, Springer, Tuebingen, LNCS, vol 5481, pp 327–338, DOI doi:10.1007/978-3-642-01181-8$_2$8

Castelli M, Manzoni L, Silva S, Vanneschi L (2011) A quantitative study of learning and generalization in genetic programming. In: Silva S, Foster J, Nicolau M, Machado P, Giacobini M (eds) Genetic Programming, Lecture Notes in Computer Science, vol 6621, Springer Berlin Heidelberg, pp 25–36

Dubcakova R (2011) Eureqa: software review. Genetic Programming and Evolvable Machines 12(2):173–178, DOI doi:10.1007/s10710-010-9124-z

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. Journal of Machine Learning Research 3:1157–1182

Icke I, Bongard J (2013) Improving genetic programming based symbolic regression using machine learning. IEEE Congress on Evolutionary Computation, CEC 2013

Jerome Friedman TH, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33(1)

Jiang H (2013) Glmnet for matlab. http://www-stat.stanford.edu/ tibs/glmnet-matlab/

Koza JR (1992) Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA

Koza JR (2010) Human-competitive results produced by genetic programming. Genetic Programming and Evolvable Machines 11(3/4):251–284, DOI doi:10.1007/s10710-010-9112-3, tenth Anniversary Issue: Progress in Genetic Programming and Evolvable Machines

Laird AR, Fox PM, Eickhoff SB, Turner JA, Ray KL, McKay DR, Glahn DC, Beckmann CF, Smith SM, Fox PT (2011) Behavioral interpretations of intrinsic connectivity networks. J Cognitive Neuroscience 23(12):4022–4037

McConaghy T (2011) FFX: Fast, scalable, deterministic symbolic regression technology. In: Riolo R, Vladislavleva E, Moore JH (eds) Genetic Programming Theory and Practice IX, Genetic and Evolutionary Computation, Springer, Ann Arbor, USA, chap 13, pp 235–260, DOI doi:10.1007/978-1-4614-1770-5$_1$3$, URLhttp: //trent.st/content/2011-GPTP-FFX-paper.pdf$

McRee RK (2010) Symbolic regression using nearest neighbor indexing. In: Gustafson S, Kotanchek M (eds) GECCO 2010 Symbolic regression workshop, ACM, Portland, Oregon, USA, pp 1983–1990, DOI doi:10.1145/1830761.1830841

Michalski RS (2000) Learnable evolution model: Evolutionary processes guided by machine learning. Machine Learning 38:9–40

Murphy KP (2012) Machine Learning A Probabilistic Perspective. The MIT Press

Nikolaev NY, Iba H (2001) Regularization approach to inductive genetic programming. IEEE Transactions on Evolutionary Computing 54(4):359–375, DOI doi:10.1109/4235.942530, URL http://ieeexplore.ieee.org/iel5/4235/20398/00942530.pdf?isNumber=20398

O'Neill M, Vanneschi L, Gustafson S, Banzhaf W (2010) Open issues in genetic programming. Genetic Programming and Evolvable Machines 11(3-4):339–363, DOI 10.1007/s10710-010-9113-2, URL http://dx.doi.org/10.1007/s10710-010-9113-2

Poldrack RA, Nichols TE (2011) Handbook of functional MRI data analysis. Cambridge University Press

Searson D (2013) Gptips for matlab. https://sites.google.com/site/gptips4matlab/

Sherry D, Veeramachaneni K, McDermott J, O'Reilly UM (2011) Flex-GP: Genetic programming on the cloud. In: Di Chio C, Agapitos A, Cagnoni S, Cotta C, Fernandez de Vega F, Di Caro GA, Drechsler R, Ekart A, Esparcia-Alcazar AI, Farooq M, Langdon WB, Merelo JJ, Preuss M, Richter H, Silva S, Simoes A, Squillero G, Tarantino E, Tettamanzi AGB, Togelius J, Urquhart N, Uyar AS, Yannakakis GN (eds) Applications of Evolutionary Computing, EvoApplications 2012: EvoCOMNET, EvoCOMPLEX, EvoFIN, EvoGAMES, EvoHOT, EvoIASP, EvoNUM, EvoPAR, EvoRISK, EvoSTIM, EvoSTOC, Springer Verlag, Malaga, Spain, LNCS, vol 7248, pp 477–486, DOI doi:10.1007/978-3-642-29178-4_48

Smits G, Kordon A, Vladislavleva K, Jordaan E, Kotanchek M (2005) Variable selection in industrial datasets using pareto genetic programming. In: Yu T, Riolo RL, Worzel B (eds) Genetic Programming Theory and Practice III, Genetic Programming, vol 9, Springer, Ann Arbor, chap 6, pp 79–92, DOI doi:10.1007/0-387-28111-8_6

Stijven S, Minnebo W, Vladislavleva K (2011) Separating the wheat from the chaff: on feature selection and feature importance in regression random forests and symbolic regression. In: Gustafson S, Vladislavleva E (eds) 3rd symbolic regression and modeling workshop for GECCO 2011, ACM, Dublin, Ireland, pp 623–630, DOI doi:10.1145/2001858.2002059

Tibshirani R (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society (Series B) 58:267–288

Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society Series B 67(2):301–320

Index