Learning Noise

Michael Schmidt
Computational Synthesis Laboratory
Cornell University
Ithaca NY, 14853, USA
mds47@cornell.edu

Hod Lipson
Computational Synthesis Laboratory
Cornell University
Ithaca, NY 14853, USA
hod.lipson@cornell.edu

ABSTRACT

In this paper we propose a genetic programming approach to learning stochastic models with unsymmetrical noise distributions. Most learning algorithms try to learn from noisy data by modeling the maximum likelihood output or least squared error, assuming that noise effects average out. While this process works well for data with symmetrical noise distributions (such as Gaussian observation noise), many real-life sources of noise are not symmetrically distributed, thus this approach does not hold. We suggest improved learning can be obtained by including noise sources explicitly in the model as a stochastic element. A stochastic element is a random sub-process or latent variable of a hidden system that can propagate nonlinear noise to the observable outputs. Stochastic elements can skew and distort output features making regression of analytical models particularly difficult and error minimizing approaches inhibiting. We introduce a new method to infer the analytical model of a system by decomposing non-uniform noise observed at the outputs into uniform stochastic elements appearing symbolically inside the system. Results demonstrate the ability to regress exact analytical models where stochastic elements are embedded inside nonlinear and polynomial hidden systems.

Categories and Subject Descriptors

I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms – *Algebraic algorithms, Analysis of algorithms, nonalgebraic algorithms*

General Terms

Algorithms

Keywords

Symbolic Regression, Dynamical Systems, Stochastic Elements

1. INTRODUCTION

Random noise is found in many natural and engineered systems, such as random diffusion, noisy actuators or sensors, and human input [4]. Most learning algorithms handle noise by fitting the maximum likelihood or least squares error of noisy data [2,4]. This approach works well when noise is distributed symmetrical about the true system output, such as white noise, Gaussian noise, and any zero mean noise superimposed over the output.

When noise exists internally in the system, it can be coupled with nonlinear components of the system. In other words symmetric internal noise can be scaled, offset, and in general transformed to produce non-symmetric noise distributions on the output. In these situations, the noise has deformed the maximum-likelihood output from the theoretical noiseless system, and the regressed models may no longer describe the analytical structure of the system.

We call this type of noise a *stochastic element* – a random process inherent to the system, affecting its behavior and observable output. Noise from stochastic elements can propagate nonlinearly to the system's output and produce non-uniform variation.

The most common approach to handling noise is to model its expectation, either through averaging or least-squares fitting [2,4]. While the expectation of a noisy system is valuable for finding a model with minimal error, it can be misleading when finding a descriptive analytical model of the system (e.g. symbolic regression). In the worst case, it can distort the observed output of the system, preventing the true system structure from being found [6].

In this paper, we aim to improve regression of a noisy system based on the notion that observed noise that is coupled to the system may itself provide additional information about the system's analytical structure. For example, if the output noise appears to grow quadratically, there is likely to be some quadratic structure in the system. Our approach is to use symbolic regression to model the output noise explicitly, decomposing noise as uniform stochastic elements inside the system to produce a noisy model. We then compare the noise observed in candidate models to the variation in the training data to calculate fitness. The final analytical model is obtained by removing the stochastic terminals used.

In the remaining sections, we discuss the distortion produced by stochastic elements, describe our approach in greater detail, show some simple results, and finish with concluding remarks.

2. BACKGROUND

2.1 Distortion from Stochastic Elements

Expected values of a noisy output can disguise and distort analytical structure when the system contains internal stochastic elements [5,6,9]. Noise can be multiplied into the system or pass

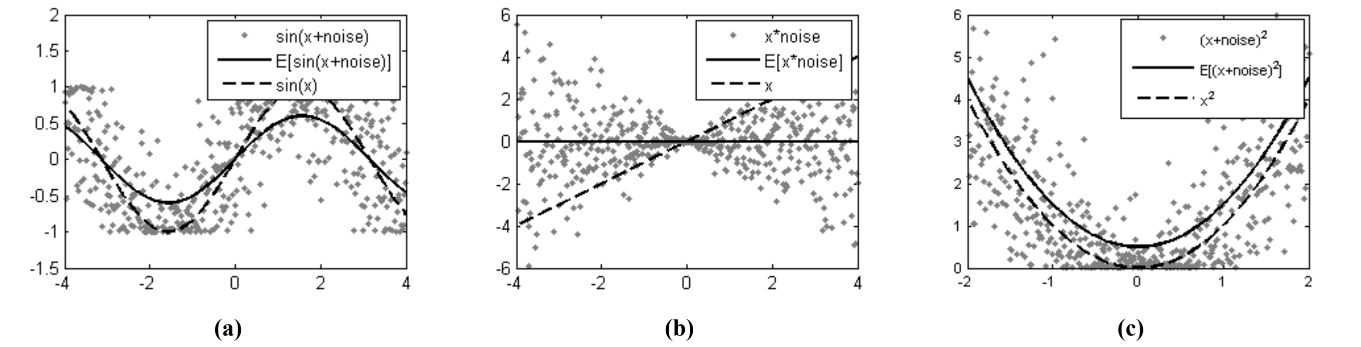


Figure 1. Three basic examples where a stochastic element hides or distorts analytical features of the system to different extents. Blue dots show the observed system output, the red line shows the expectation of the output, and the green line show the target analytical model with stochastic elements removed.

through a nonlinear operation to significantly change the expected output values. Figure 1 shows three simple examples where a stochastic element hides or distorts analytical features.

Figure 1 (a) shows a sine function, *f(x)* = sin*(x)* with a stochastic element giving rise to a random phase offset, *f(x)* = sin*(x + R)*. The noise does not change the magnitude of the sine wave but does shift data samples left or right. The expectation of the output shows a sine function with correct phase but with smaller amplitude than the target analytical model, *f(x) = A**sin*(x)*.

The system in Figure 1 (b) is a simple linear function, *f(x) = x*, multiplied by a stochastic element, *f(x) = x*R*. The multiplied noise completely hides the linear growth from the expectation. The expected output becomes simply *f(x)* = 0.

Figure 1 (c) is a quadratic function, $f(x) = x^2$, with noise added to the input, $f(x) = (x + R)^2$. This noise again shifts the data points left or right, but does not change the y-intercept. The expected output model however is quadratic with a y-offset, $f(x) = x^2 + A$.

Though these are simple examples, they give insight into how stochastic elements can distort expectation models from the exact analytical model, or even hide features. In the next section, we describe a simple approach to incorporating stochastic elements into models in order to recover exact analytical models despite this difficulty.

2.2 Regressing Noising Data

Noise is found in almost all experimental data and is a central focus in many areas of machine learning [13]. Here, we briefly overview how noise is traditionally handled in regression problems.

Often experimental data is pre-processed to remove outliers [10], remove white noise [9], and more generally, smooth features. Common techniques for preprocessing include convolving with a low-pass-filter (e.g. box or sliding window, Gaussian filter), local least-squares fitting, and spline fitting.

The aim of preprocessing is to transform the data set to be more representative of the expected outcome or maximum likelihood of the system through interpolation or statistical properties among neighboring data points. These processes make assumptions about the underlying system and its noise distribution but are still used frequently in practice to improve predictive performance.

In contrast, we are interested in exploiting the existence of nonlinear noise to reveal internal structure of the unknown system. In this sense, the goal is broader and removing noise coupled to the system could remove information.

2.3 Modeling Noise and Confidence

One is often interested in the confidence of predictions made by a regressed model. Accurate models predict the maximum-likelihood value, but the variance of outputs for this value may be large.

The most common non-parametric approach to measure confidence is to examine the residual errors of the model on the training set. This leads to a natural two-step procedure:

(1) Regress a best fit model

(2) Derive a statistical model of the residual error

In the case of white noise, residual errors appear uniformly distributed and can be modeled globally such as calculating its mean and variance.

If noise is coupled to the system by an internal stochastic element, the residual error may vary greatly over the input space. In this case, local statistical models are used to model confidence among neighboring inputs [11].

Deriving a statistical model of the residual error in this fashion requires assuming a noise distribution model, such as the normal distribution. In nonlinear regression, where an analytical model of the system is assumed, the noise distribution can be derived automatically from the model. Most commonly, confidence intervals are calculated on the model fitting parameters [12]. Parameter confidences then translate into nonlinear output confidence ranges on the model output.

In contrast, the method proposed in this paper models noise explicitly in the model parametrically without a predetermined model structure.

2.4 Symbolic Regression

Symbolic Regression is the problem of identifying the exact mathematical (analytical) description of a hidden system from experimental data [1,7,8]. Unlike polynomial regression or related machine learning methods which also fit data, symbolic regression is a system identification method which explicates behavior. Symbolic regression is an important problem because it is related to general machine learning problems but is an open-ended discrete problem that cannot be solved greedily, thus requiring non-standard methods.

For experiments in this paper, we represent algebraic expressions (candidate solutions) as a procedural list of algebraic operations on local variables [1], effectively a graph encoding. Empirically, the graph encoding has comparable performance to tree encodings but has significant computational advantages.

The operations can be unary operations such as *abs*, *exp*, and *log*, or binary operations such as *add*, *mult*, and *div*. If some a priori knowledge of the problem is known, the types of operations available can be narrowed ahead of time [1,8]. The terminal values available consist of the function's input variables and the function's evolved constant values.

Mutation in a symbolic expression can change an operation type (eg. change add to sub), change the arguments of an operation (eg. change x+1 to $x+x$), delete an operation (eg. change $x+x$ to x), or add an operation (eg. change $x+x$ to $x + (x*x)$).

```
Individual ind = { encoding E, stochastic elements S }
Input variables X

Function evalute()
    For each s in S
        s = random value [-1, 1]
    End
    ...
    val = evaluate ind normally
    ...
    Return val
End
```

```
Individual ind
Training data D of (x,y) pairs
Number of samples N

Function fitness()
    fitness = 0
    For each d in D
        $y_{in}$, $y_{max}$
        Repeat N times
            y = ind.evaluate(d.x)
            If ($y < y_{min}$) $y_{min} = y$
            If ($y > y_{max}$) $y_{max} = y$
        End
        If ($y_{min} < d.y < y_{max}$)
            fitness += $1/(y_{max} - y_{min})$
        Else
            fitness += $-\min(|d.y - y_{max}|, |d.y - y_{max}|)$
        End
    End
    Return fitness
End
```

Crossover of a symbolic expression exchanges sub-trees, or in the graph encoding used here two sub-graphs, from two parents. For example, crossing $f_1(x) = x^2 + 1$ and $f_2(x) = x^4 + \sin(x) + x$ could produce a child $f_3(x) = x^2 + \sin(x)$. In this example, the leaf node +1 was exchanged with the sin(x) term.

The fitness objective in symbolic regression, traditionally, is to minimize error on the training set [1,7,8]. Later in this paper however, we define a new objective geared specifically to reward candidate solutions with noise distributions that match the noise observed in the training set.

3. OUR APPROACH

The basic idea of our approach is to include behavior of stochastic elements inside the analytical model. Instead of using an error minimization objective, we attempt to find a model of stochastic elements with the simplest distribution explaining all features and noise in the training data. The final analytical model identifies the origin of noise as well as its effect on out observations.

Much research has been done on bounding noise error and modeling error distributions [3,11,12]. The distinction here is that we are modeling individual noise components explicitly inside a system. The analytical model is regressed from scratch, rather than relying on an assumed system model or distribution model.

We use symbolic regression to find an analytical model which incorporates uniform random variables to explain residual error parametrically in addition to finding a best fit. In the next two sections, we describe how we incorporate stochastic elements into candidate models and describe a new objective function to explain observed noise.

3.1 Decomposing Stochastic Elements

Our basic building block for a stochastic element is a uniform random variable with range [-1,1], that returns a random value every time it is read or evaluated by the model. Symbolic regression can incorporate this random variable anywhere in its models to help explain the noise distribution.

$$R() = \text{uniform random value } [-1, 1]$$

Nearly all types of random variables and distributions can be derived from this uniform random variable. Symbolic regression treats this variable like it would any other attribute variable, and can derive combinations and transformations to non-uniform distributions. For example, the Normal distribution can be derived

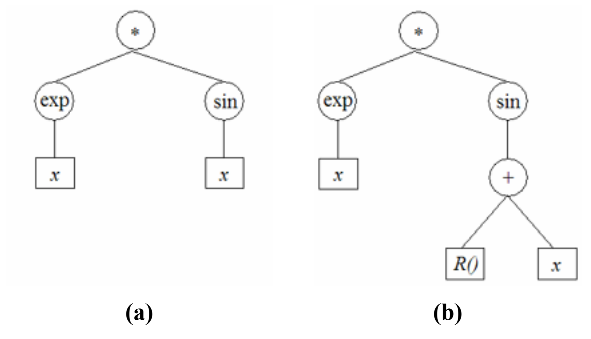
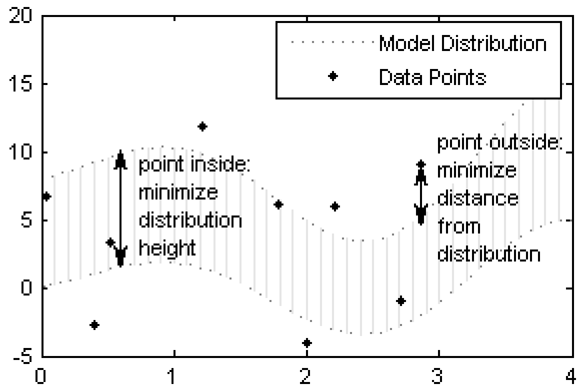
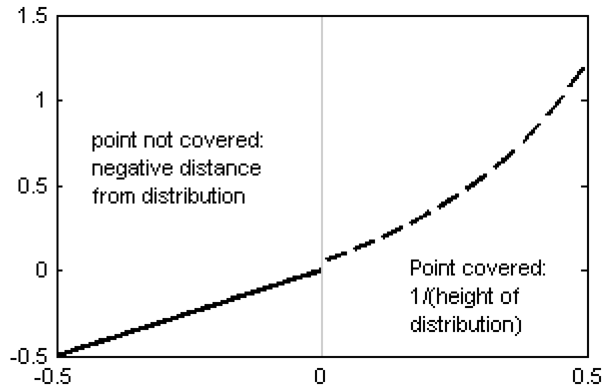


Figure 2. An example binary expression tree (a) for the function $f(x)=e^x\sin(x)$, and a similar tree modeling a stochastic element (b) for the function $f(x)=e^x\sin(x+R())$.



(a)



(b)

Figure 3. The fitness objective for explaining training data with a with model that has stochastic elements and output distribution. If a training point falls inside the model distribution, the objective is to minimize the height of the distribution. If the point falls outside, the objective is to minimize the distance of the point to the distribution.

from querying the uniform random variable twice:

$$Normal = -A \cdot \sqrt{\ln(2) - \ln(R() + 1)} \cdot \cos(R() \cdot \pi)$$

Symbolic regression most commonly represents candidate solutions as expression trees (Figure 3.a).

We treat stochastic elements as a new variable in the terminal set that can be used anywhere in the expression tree to model the noise in experimental data (Figure 3.b). The new terminal value is special however in that it is randomized every time it is evaluated, even when appearing multiple times in the same expression tree.

3.2 The Noise Distribution Objective

Now that candidate models can include random variables, their output predictions will have some distribution. Our goal for this distribution is to explain all variation found in the training data, and do so in the narrowest and simplest way.

A distribution explains a training data point if the data point falls inside the model's distribution at that point. For example if $f(x=10)$ has a distribution between [-9,-3], it explains the training data point if its value is -6, but not if it is 4.

We can approximate the distribution of a candidate model at a training point by sampling it. In our experiments, we find the range of output for a training input by storing the minimum and maximum output from 100 model evaluations.

Note however that a trivial solution would be a large (or perhaps infinite) distribution where all training data lies inside the distribution. Therefore, we must introduce a second objective to minimize the size of the distribution.

If a training data point lies inside the model's distribution, we want to minimize the height of the distribution at that point. If the point is not covered by the distribution, we want to minimize the distance of that point from the distribution. We can combine these two objectives into a single fitness criterion:

$$fitness(f) = \sum_{(x,y)} \begin{cases} -\min |y - range(f(x)| & \text{if } y \notin range(f(x)) \\ 1/range(f(x)) & \text{if } y \in range(f(x)) \end{cases}$$

This is a two-step fitness objective, summarized in Figure 3. The model must first cover the point with its distribution, and then it must minimize the area of its distribution. As shown in Figure 3 (b), training points not explained by the distribution contribute negatively to the fitness, and points that are explained contribute positively.

Psuedocode for evaluating a model that contains stochastic elements, and for evaluating the distribution fitness of a model is shown on Page 3.

4. EXPERIMENTS

We modify a symbolic regression algorithm [1] to include stochastic elements and regress based on distributions rather than error minimization. This algorithm utilizes adaptive sampling of the training set to reduce computational cost, which is particularly high for finding the output distribution of candidate models during regression.

Table I. Summary of Experiment Setup

Solution Population Size	64
Selection Method	Deterministic Crowding
P(mutation)	0.05
P(crossover)	0.75
Solution Encoding	Operation List (graph)
Operations	16
Local Variables	4
Evolved Constants	4
Inputs	1
Operator Set	{ +, -. *, /, *sin*, *cos* }
Terminal Set	{ x, c_1, c_2, c_3, c_4 }
Crossover	variable, single point
Fitness Sample Size	4
Distribution Samples	100

Parameters for all experiments are summarized in Table I. In deterministic crowding, offspring replace their most similar parent if they have equal or higher fitness and are discarded otherwise. Population size, mutation probability, and crossover probability have been tuned empirically. Crossover produces a higher fit child approximately 20% of the time with these setting on the operation list encoding.

The candidate solutions (algebraic expressions) are lists of operations on local variables. The number of operations and local variables were tuned for computational performance. The encoding size, terminal set, and operator set are over-represented (no experiments requires all for convergence). Single point crossover is used on the operation list at a variable offset.

To measure fitness, the output distribution is measured on four inputs from the training set, one hundred times. The minimum and maximum values are then used to calculate the fitness described earlier.

We test on three simple example systems each with a uniform stochastic element coupled in the system:

- $f_1(x) = 10 \sin(x + R)$
- $f_2(x) = x^2 \sin(x + R)$
- $f_3(x) = (x + R) - 1.5 x^3$

These experiments demonstrate the finding the exact structure and parameters of the system despite internal stochastic noise which offset the expected output.

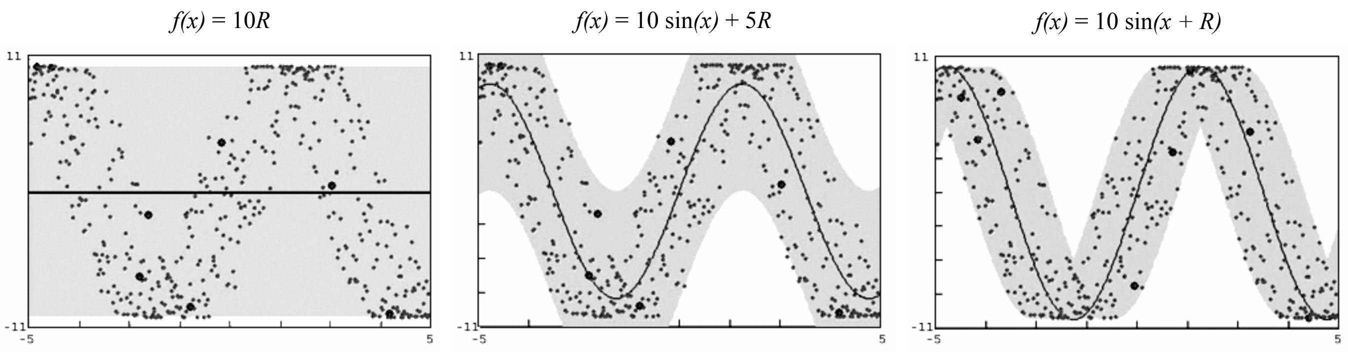
5. RESULTS

This section gives results on three simple examples of regressing stochastic elements embedded in a hidden system to demonstrate our approach. We show screen captures of different stages during regression to show the progress toward the analytical model.

The time to regress each system successfully ranged from one to five minutes. The primary computation time consists in computing the candidate model distribution at each training point. We use random sampling to determine the output ranges at each point, but a more intelligent sampling method could be used to scale the application to higher complexity systems.

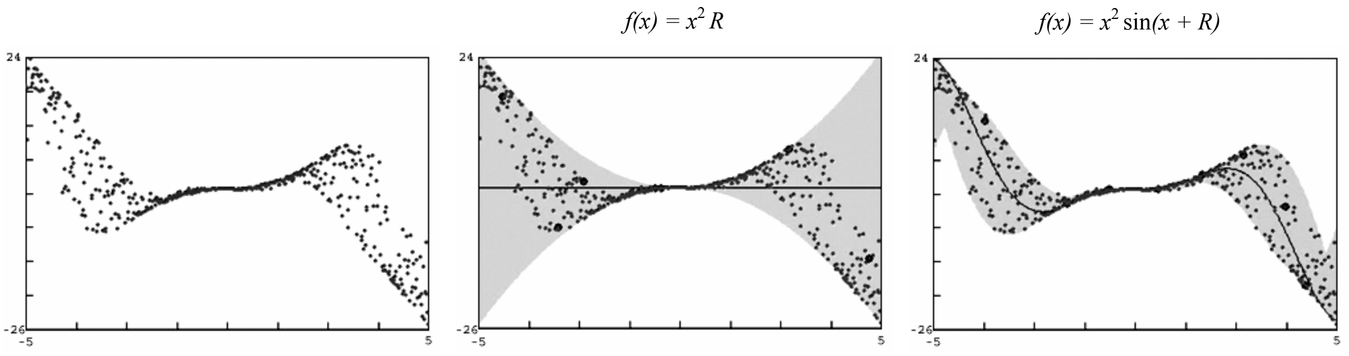
Figure 4 shows three stages during regression of the function $f(x)=10*\sin(x+R)$, where R is a stochastic element variable that returns a uniformly random number in the range [-1,1] each time it is read.

Early on, candidate solutions are linear with distributions that cover all the training points – shown in Figure 4 (a). In Figure 4 (b), the solutions have inferred the sine function in the system, but the noise distribution is just added linearly to the output. In the next stage, Figure 4 (c), the solution has converged on the sine



(a) **(b)** **(c)**

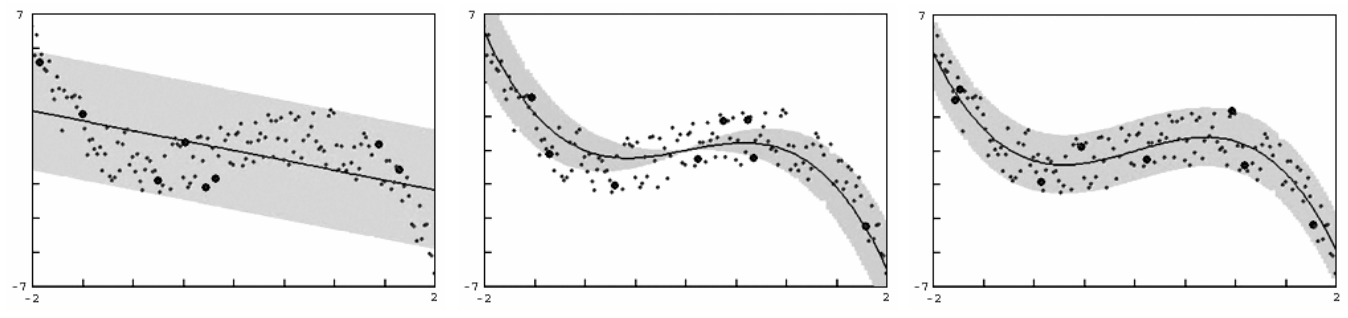
Figure 4. The best model found at three points during regression of $f(x) = 10 \sin(x + R)$. The green points are the training data, the grey area is the model's distribution, and the blue line is the analytical model with stochastic elements removed.



(a) **(b)** **(c)**

Figure 5. The best model found at three points during regression of $f(x) = x^2 \sin(x + R)$. The green points are the training data, the grey area is the model's distribution, and the blue line is the analytical model with stochastic elements removed.

f(x) = -1.5 *x* + 3*R* | *f(x)* = *x R* - 1.5 x^3 | *f(x)* = *(x + R)* - 1.5 x^3



(a) **(b)** **(c)**

Figure 6. The best model found at three points during regression of *f(x) = (x + R)* - 1.5 x^3. The green points are the training data, the grey area is the model's distribution, and the blue line is the analytical model with stochastic elements removed.

function with the stochastic element located inside the sine function.

Figure 5 shows the regression of the function $f(x)=x^2*\sin(x+R)$ which is similar to the first experiment but now has a variable amplitude sine wave. Candidate solutions converge on quadratic amplitude noise very quickly – Figure 5 (b). Shortly after, the sine function is found and the analytical model converges in Figure 5 (c).

The third experiment uses a polynomial function but with noise simply added linearly to the output. This is a case where the minimum error model is the same as the analytical model but it is important that we can differentiate this type of noise as well.

Figure 6 (a) shows early candidate solutions are linear with an additive noise range. In Figure 6 (b), the analytical model has been found but the noise distribution has not yet explained all data points. Figure 6 (c) shows the converged solution identifying the correct analytical model and its distribution.

6. CONCLUSIONS

Stochastic elements existing inside a hidden system can produce nonlinear and non-uniform noise at the observable outputs. There are many cases where the expected value output or minimum error regression can be deceiving toward finding an exact analytical model as done in symbolic regression.

We have presented a simple approach to model stochastic elements directly as uniform random features using symbolic regression. The objective for candidate models with stochastic elements is to explain (overlap) all training data points in its distribution and minimize the area of the distribution used.

Results show this approach can find the exact analytical model despite misleading nonlinear and non-uniform output noise. In three basic experiments, regression of the output distribution found the correct system structure and location of the stochastic elements with parameters existing in the hidden system.

7. ACKNOWLEDGEMENTS

This research was supported in part by the U.S. National Science Foundation grant number CMMI-0547376.

8. REFERENCES

[1] Schmidt, M., and Lipson, H. "Coevolution of Fitness Maximizers and Fitness Predictors", GECCO Late Breaking Paper, 2005.

[2] Rasmussen C. 1996. Evaluation of Gaussian Processes and Other Methods for Non-Linear Regression. PhD thesis, Department of Computer Science, University of Toronto.

[3] Llor`a, X., Goldberg, D.E. "Bounding the effect of noise in Multiobjective Learning Classifier Systems." Evolutionary Computation, 2003.

[4] V.G. Kulkarni, Modeling and Analysis of Stochastic Systems. Chapman Hall, 1995.

[5] Schafer, W, Ellner S, Kot M. "Effects of noise on some dynamical models in ecology." J Math Biology 1986.

[6] Theiler, J., Galdrikian, B., Longtin, A., Eubank, S. and farmer, J.D. "Detecting non-linear structure in time series." First Experimental Chaos Conference, 1991

[7] Koza, J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: The MIT Press, 1992.

[8] Augusto D. A. and Barbosa H. J. C. "Symbolic Regression via Genetic Programming," VI Brazilian Symposium on Neural Networks (SBRN'00), 01: 22-01, 2000.

[9] Kleijnen, Jack P.C., "White Noise Assumptions Revisited: Regression Models and Statistical Designs for Simulation Practice" CentER Discussion Paper No. 2006-50.

[10] Rousseeuw, P.J. and Leroy, A.M. "Robust Regression and Outlier Detection." New York: Wiley, 1987.

[11] Nix, D.A., and A.S. Weigend (1995). "Learning Local Error Bars for Nonlinear Regression." In Advances in Neural Information Processing Systems 7 (NIPS*94) p. 489--496. Cambridge, MA: MIT Press.

[12] Vugrin, K.W., Swiler, L., Roberts, R., et al. "Confidence Region Estimation Techniques for Nonlinear Regression: Three Case Studies." SAND Report, October 2005.

[13] Arnold D., "Evolution strategies in noisy environments–A survey of existing work," in Theoretical Aspects of Evolutionary Computing, 2001, pp. 239-24.