Optimal Experiment Design for Coevolutionary Active Learning

Daniel Le Ly, *Student Member, IEEE,* and Hod Lipson, *Member, IEEE*

***Abstract*—This paper presents a policy for selecting the most informative individuals in a teacher-learner type coevolution. We propose the use of the surprisal of the mean, based on Shannon information theory, which best disambiguates a collection of arbitrary and competing models based solely on their predictions. This policy is demonstrated within an iterative coevolutionary framework consisting of symbolic regression for model inference and a genetic algorithm for optimal experiment design. Complex symbolic expressions are reliably inferred using fewer than 32 observations. The policy requires 21% fewer experiments for model inference compared to the baselines and is particularly effective in the presence of noise corruption, local information content as well as high dimensional systems. Furthermore, the policy was applied in a real-world setting to model concrete compression strength, where it was able to achieve 96.1% of the passive machine learning baseline performance with only 16.6% of the data.**

***Index Terms*—Active learning, competitive coevolution, optimal experiment design, Shannon information criterion.**

I. Introduction

ACTIVE machine learning is a growing field of research that focuses on leveraging the ability to interact with the environment to improve and accelerate the learning process. The traditional machine learning paradigm infers models by processing a fixed data set, which is assumed to comprehensively characterize the phenomena of interest. In contrast, an active approach is able to investigate the phenomena by tailoring its dynamic data set with new observations to guide the modeling process.

Although active learning is promising due to its ability to adapt, the fundamental challenge is to determine a method of data collection that results in superior models. Fine parameter sweeps in the input space may capture the required details for model inference, but this approach generates vast amounts of data, most of which is highly redundant. Furthermore, in many physical fields, measuring a single data point may be costly and difficult, making it infeasible to gather sufficiently comprehensive data sets. Thus, collecting just the right data needed to best advance the modeling task remains a challenge.

Optimal experiment design aims to provide a solution to this fundamental issue—new data is gathered according to its ability to create disagreement between arbitrary models. By investigating areas of predicted ambiguity, the information content in each observation is maximized and superior models can be distinguished based on these observations. Furthermore, a non-parametric approach provides a powerful and robust tool that allows the comparison of candidate models without requiring analysis of the model's underlying structure.

Teacher-learner type coevolutionary algorithms are a popular approach for implementing active learning. Rather than treat active learning as a monolithic problem, it is divided into two complementary subproblems: one population infers models using a dynamic data set while the second population adds to the data set by designing experiments that disambiguate the current candidate models. Each evolutionary algorithm leverages the advancements in its counterpart to achieve superior results in a unified active learning framework. Flavors of this coevolution active learning have been used in fields such as robotics [1] and automated science [2], [3].

This paper presents a policy for optimal experiment design in active learning algorithms based on a Shannon information criterion. The problem considered is as follows: given a collection of arbitrary and competing models, determine the next experiment that will best disambiguate the models based solely on their predictions. Shannon information is used as the underlying theory for the optimal experiment criterion—we show that traditional entropy is prohibitively computationally expensive and we propose using the closely related measure surprisal of the mean as the optimization criterion. This approach is applied to an iterative coevolutionary problem that uses symbolic regression for model inference and a genetic algorithm for experiment design. Complex symbolic expressions are reliably inferred using fewer than 32 data points. Furthermore, the surprisal of the mean policy outperforms all published baselines and requires 21% fewer experiments to achieve the same performance. The surprisal of the mean policy is found to be particularly effective in situations of noisy data, local information content, as well as high dimensional systems. Finally, for a real-world concrete compression strength modeling task, the surprisal of the mean policy is able to achieve 96.1% of the performance of a passive machine learning baseline with only 16.6% of the data.

II. Related Work

Although coevolution has been used to primarily investigate game-theoretic constructions, there is a growing interest in

Manuscript received July 5, 2012; revised December 17, 2012 and March 14, 2013; accepted June 17, 2013. Date of publication September 11, 2013; date of current version May 27, 2014.

D. L. Ly is with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: dll73@cornell.edu).

H. Lipson is with the Sibley School of Mechanical and Aerospace Engineering and also with the Faculty of Computing Information Science, Cornell University, Ithaca, NY 14853 USA (e-mail: hod.lipson@cornell.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2013.2281529

1089-778X © 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

leveraging the coevolutionary dynamics for active learning applications. In particular, the estimation and exploration algorithm (EEA) has shown impressive results of applications in coevolutionary active learning. The algorithm is divided into two phases: the estimation phase evolves predictive models while the exploration phase evolves experiments to gain more information. Through active learning, EEA has successfully inferred complex models using minimal trials, with applications that include inferring genetic networks [4], deterministic finite automata [5], non-linear systems [2], [6], and evolutionary robotics [1].

In addition to EEA, flavors of coevolutionary active learning have been applied to a wide variety of problems, including active sub-sampling for unbalanced data sets in GP classification [7], autonomous robot navigation [8], automatic system identification [9], coevolutionary evaluation [10], and coevolving fitness predictors [11]–[13]. Furthermore, coevolutionary active learning is often found in multiagent systems; a survey that includes summaries of numerous applications can be found in [14]. However, the previous work on coevolutionary active learning often focused on the evolutionary dynamics and used an experiment design optimization criterion that was based on heuristics or intuition. In comparison, our work focuses exclusively on the optimization criterion and develops a formal framework based on information theory for experiment design.

Within the machine learning community, active learning has also been an area of growing research. As recent surveys indicate, the vast majority of active learning is focused on classification problems and recommender systems [15], [16]. The primary problem of interest is, given a collection of unlabeled examples and the ability to request labels for specific examples, how can classifiers or recommenders be improved by selective labeling [17]–[19]. Although this field shares many information theory concepts, there is a fundamental difference in the subject—namely, classification or recommendation instead of regression.

Cohn *et al.* [20] introduced influential work on active learning for regression. They described two statistically-based learning architectures based on mixtures of Gaussians and locally weighted regression. However, they used function approximations in the form of one of the two statistical models, as opposed to leveraging competing models simultaneously.

Gaussian processes have also been applied as a tool for active learning for regression tasks [21], [22]. A Gaussian process model is fitted and areas of high variance or mutual information are used to collect data. However, Gaussian processes are dictated by a single kernel or covariance function, and thus, are a single stochastic model with hyperparameters.

Finally, the concept of optimal experiment design has been rigorously explored in the control systems and system identification community [23]. The work in this field is relatively comprehensive and includes topics such as parameterized Bayesian criterion [24], uncertainties in parameter estimation [25], correspondence with optimal control [26], least costly control identification [27], and robust system identification [28]. However, the control systems community generally begins with a model of the fundamental dynamics obtained from first principles and, consequently, focuses primarily on obtaining optimal estimates of unknown parameters in their models.

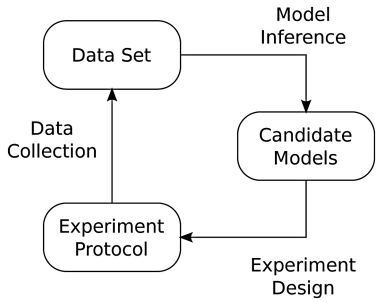


Fig. 1. Flowchart describing the relationship between the three processes of active learning.

III. Optimal Experiment Design via Information Theory

Active learning consists of three iterative processes (summarized in Fig. 1):

1) Model inference: the traditional learning process of inferring a model given a data set.
2) Experiment design: the process of determining the next experiment.
3) Data collection: the process of executing a desired experiment and inserting the measured results into the data set.

This paper presents a framework for experiment design according to a Shannon information criterion for a collection of arbitrary and competing models. This section begins with a formal problem definition and an introduction to the statistical framework. A discussion of entropy follows, along with the description of the proposed criterion: surprisal of the mean. This section concludes with a specific formulation for Gaussian models.

A. Problem Definition for Experiment Design

Consider the following problem: given a data set and a collection of arbitrary models, determine the experiment that best disambiguates the candidate models. A distinguishing feature of this problem is that the form of the underlying system is not known. Consequently, all models are plausible and a primary goal is to build a data set that aids in inferring which candidate model best represents the underlying system.

Consider the scenario depicted in Fig. 2: for a data set with five data points, there are three candidate models—a linear, cubic, and sinusoid model, fit with least squares regression. These models vary in their predictive accuracy, and more importantly, produce different predictions depending on the input. Thus, the problem is selecting the next input experiment x^* that optimally determines which of the three models best represents the underlying system.

Note that while this example uses a relatively simple collection of models, more complex models such as feedforward neural networks or support vector regression can be applied.

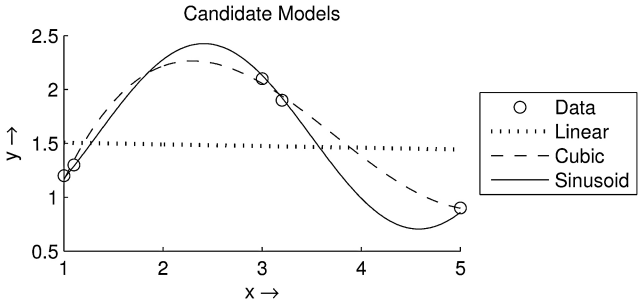


Fig. 2. Model inference problem example with five data points and three candidate models. The goal of experiment design is to determine the optimal input x^* to disambiguate the candidate models and accelerate the inference process.

The distinguishing feature of non-parametric approaches is that they are agnostic to the parameter space of the candidate models and only rely on model predictions for experiment design.

Formally, for a set of multidimensional inputs $x_n \in \mathbb{R}^m$ and corresponding output $y_n \in \mathbb{R}$, consider the underlying model $\mathcal{F}(x)$

$$y_n = \mathcal{F}(x_n) + \epsilon \tag{1}$$

where ϵ is an additive noise term. The noise can be drawn from any distribution $p(y|\mu, \theta)$ where μ is a location parameter and θ represents all the remaining model parameters.

The fundamental goal of learning algorithms is to infer the underlying model $\mathcal{F}(x)$ from the collection of data (x_n, y_n). Active learning is concerned with the complementary task of determining the next input experiment x^* that disambiguates which candidate best represents the underlying system, given a variety of candidate regression models $f_i(x)$.

B. Statistical Representation of Regression Models

This section introduces a statistical framework for a collection of regression models. Rather than interpreting noise as an additive term, a candidate regression model $f_i(x)$ can be represented statistically as a probability distribution of observing an output where the location parameter is the regression model. Thus, the regression problem (1) can be equivalently restated as the probability density of observing the random variable Y_i as output y_n given the i-th regression model $f_i(x_n)$

$$p_i(y_n) = p\left(y_n | f_i(x_n), \theta\right). \tag{2}$$

As a collection of regression models are presented, the collective predictions of the candidate models are determined via joint probabilities. Since the candidate regression models are independent by definition, the probability density of all the models observing the same output is simply

$$p(y_n) = \prod_i p_i(y_n | f_i(x_n), \theta). \tag{3}$$

Finally, each candidate regression model $f_i(x)$ is assumed to be locally optimal, which is obtained by optimizing the corresponding maximum likelihood estimator (4), or maximum log-likelihood estimator (5), for a parametric class $\check{f}_i(x)$

$$f_i(x) = \underset{\check{f}_i(x)}{\operatorname{argmax}} \frac{1}{N} \sum_n \mathcal{L}(\check{f}_i(x_n)|y_n, \theta) \tag{4}$$

$$= \underset{\check{f}_i(x)}{\operatorname{argmax}} \frac{1}{N} \sum_n \sum_i \log p_i(y_n|\check{f}_i(x_n), \theta). \tag{5}$$

Note that achieving optimality for the candidate models is part of the model inference process—the experiment design process is not concerned with how the models are obtained. Approaches such as least squared regression, backpropagation or heuristic methods are all valid as long as the candidate models are locally optimal. Furthermore, optimality is only required within the same parametric class. For example, linear models are not compared against sinusoidal models.

C. Shannon Entropy

Ideally, an optimality criterion for experiment design is Shannon entropy: a quantified measure of the uncertainty associated with a random variable. Conceptually, entropy measures the uncertainty of the collection of models—an experiment with high entropy is one where the candidate models collectively disagree while a low entropy experiment is one that is well modeled by all the candidates. Furthermore, the use of entropy is strongly motivated by related work [18], [22], [24], [28].

Before discussing entropy, we introduce the concept of surprisal or self-information of a random variable $I(Y)$. Surprisal measures the information content associated with the outcome of a random variable—the rarer the event, the more information or surprisal is obtained from observing the event. The surprisal of a random variable is

$$I(Y) = -\log p(y). \tag{6}$$

Entropy H is then defined as the expected value of surprisal

$$H(Y) = \mathbb{E}\left[I(Y)\right] \tag{7}$$

$$= -\int_{\mathbb{Y}} p(y) \log p(y)\, \mathrm{d}y \tag{8}$$

where (7) is the general definition of entropy, (8) is the definition of differential entropy for continuous random variables, and $\mathbb{Y}$ is the support of Y. Unlike its discrete counterpart, differential entropy is not invariant to a change of variables and it can be negative. However, for experimental design, only relative entropy is relevant as entropy is used as a optimality criterion and thus these properties are extraneous.

For the collection of models (3), the input experiment x_H^* that satisfies the entropy optimality criterion is

$$x_H^* = \underset{x}{\operatorname{argmax}} \left[H(Y)\right] \tag{9}$$

$$= \underset{x}{\operatorname{argmax}} \left[-\int_{\mathbb{Y}} p(y) \log p(y)\, \mathrm{d}y\right]$$

$$= \underset{x}{\operatorname{argmax}} \left[-\int_{\mathbb{Y}} \left(\prod_i p(y|f_i(x), \theta)\right) \left(\sum_i \log p(y|f_i(x), \theta)\right) \mathrm{d}y\right]. \tag{10}$$

For the majority of useful distributions, Shannon entropy for a collection of regression models is not a closed form expression. As a result, computing entropy values relies on numeric integration, which itself includes sum and product terms that scale according to the number of candidate models. Calculating entropy becomes extremely computationally expensive as the sum and product terms increase linearly with the number of candidate models, while the integration domain increases exponentially based on the dimensionality of the input space. Furthermore, the entropy of a collection of regression models is generally non-convex and optimizing such criteria often requires numerous evaluations, compounding the computational load.

D. Surprisal of the Mean

As Shannon entropy is prohibitively expensive to compute, we propose a similar measure as an optimality criterion for a collection of regression models. Since entropy is defined as the expected value of surprisal or the mean surprisal, we propose using the surprisal of the mean $\check{H}(Y)$ as the optimality criterion

$$\begin{aligned}\check{H}(Y) &= I(\mathbb{E}[Y]) \\ &= -\log p\left(\mathbb{E}[Y]\right). \end{aligned} \tag{11}$$

The subsequent optimality criterion $x^*_{\check{H}}$ is

$$\begin{aligned} x^*_{\check{H}} &= \underset{x}{\operatorname{argmax}} \left[-\log p\left(\mathbb{E}[Y]\right)\right] \\ &= \underset{x}{\operatorname{argmax}} \left[-\sum_i \log p\left(\mathbb{E}[Y]|f_i(x),\theta\right)\right]. \end{aligned} \tag{12}$$

While the surprisal of the mean, referenced herein as simply surprisal, may not be same as Shannon entropy, it does provide a coarse approximation that is well-suited for optimization. In fact, as an optimization criterion, the two measures are closely related and there is a bounded error between the optimal values of each measure (see Theorem 1). In addition to the bounded relationship to entropy, surprisal's equally important property is that it has a simple, closed-form expression that is computationally lightweight—no integration is required, the number of calculations does not depend on the dimensionality of the input space, and it scales linearly with the number of candidate models.

Theorem 1: For a bounded domain $x \in [a, b]$ and a logarithmically concave distribution $p(x)$, if $x^*_{\check{H}}$ satisfies the surprisal optimality criterion, $\check{H}(x^*_{\check{H}}) \geq \check{H}(x), \forall x \in [a, b]$, then $x^*_{\check{H}}$ is also the optimal Shannon entropy within a bounded error, $H(x^*_{\check{H}}) \geq H(x) - T_f(a, b), \forall x \in [a, b]$, where

$$\begin{aligned} T_f(a,b) = \max_{\lambda\in[0,1]} [&-\lambda\log p(a) - (1-\lambda)\log p(b) + \\ &\log p(\lambda a + (1-\lambda)b)]. \end{aligned} \tag{13}$$

Proof: For logarithmically concave distributions, Jensen's inequality defines an inequality between Shannon entropy and surprisal

$$\begin{aligned} -\log p\left(\mathbb{E}[x]\right) &\leq \mathbb{E}\left[-\log p(x)\right] \\ \check{H}(x) &\leq H(x). \end{aligned} \tag{14}$$

For $x \in [a, b]$, Simic [29] provides a global upper bound on Jensen's inequality, providing a second inequality between Shannon entropy and surprisal

$$\begin{aligned} -\log p\left(\mathbb{E}[x]\right) &\geq \mathbb{E}\left[-\log p(x)\right] - T_f(a,b) \\ \check{H}(x) &\geq H(x) - T_f(a,b) \end{aligned} \tag{15}$$

where $T_f(a, b)$ is defined in (13).

For two different inputs, x_1 and x_2, the optimization of surprisal is defined by the inequality $\check{H}(x_2) \leq \check{H}(x_1)$. Given the two relationships in (14)–(15) and the transitive properties of inequalities, the following relationship can be obtained

$$H(x_2) - T_f(a,b) \leq \check{H}(x_2) < \check{H}(x_1) \leq H(x_1) \tag{16}$$

which proves Theorem 1. ■

E. Optimal Experiment Design for Gaussian Distributions

This section describes the derivation of the surprisal-based active learning framework that assumes the noise, ϵ in (1), is drawn from zero-mean Gaussian distribution with variance σ_N^2

$$\epsilon \sim \mathcal{N}(y|0,\sigma_N^2) = \frac{1}{\sqrt{2\pi}\sigma_N} e^{\frac{x^2}{2\sigma_N^2}}. \tag{17}$$

Although the Gaussian distribution is a popular and ubiquitous noise model, our surprisal framework can be readily adapted to other distributions in a similar fashion outlined in this section.

For a collection of regression models with Gaussian noise, the probability of observing an output (3) is

$$p(y_n) = \prod_i \mathcal{N}_i(y_n|f_i(x_n),\sigma_i^2) \tag{18}$$

where $\sigma_i^2 = \frac{1}{N}\sum_n (f_i(x_n) - y_n)^2$ is the empirical sample variance of the residual error of the candidate model from the data set.

The optimal regression models are determined by optimizing the corresponding maximum likelihood estimator (5), resulting in minimizing the sum of squares

$$f_i(x) = \underset{\check{f}_i(x)}{\operatorname{argmin}} \sum_n \left(\check{f}_i(x_n) - y_n\right)^2. \tag{19}$$

The expected value of the joint distribution (18), is the weighted mean

$$\begin{aligned} \bar{f}(x) = E[Y] &= E\left[\prod_i \mathcal{N}_i(y_n|f_i(x),\sigma_i^2)\right] \\ &= \frac{\sum_i \frac{1}{\sigma_i^2} f_i(x)}{\sum_i \frac{1}{\sigma_i^2}}. \end{aligned} \tag{20}$$

Thus, the surprisal optimality criterion (12), is

$$\begin{aligned} x^*_{\check{H}} &= \underset{x}{\operatorname{argmax}} \left[-\sum_i \log \mathcal{N}_i(\bar{f}(x)|f_i(x_n),\sigma_i^2)\right] \\ &= \underset{x}{\operatorname{argmax}} \left[\sum_i \frac{\left(\bar{f}(x) - f_i(x)\right)^2}{\sigma_i^2}\right]. \end{aligned} \tag{21}$$

The surprisal criterion for a collection of regression models with Gaussian noise is a simple closed-form expression that is easy to calculate and computationally light. In comparison,

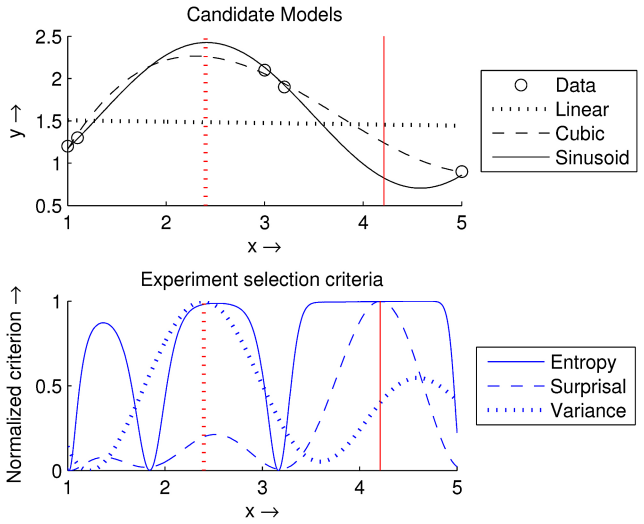


Fig. 3. Normalized measures of entropy, surprisal, and variance for the candidate model example. The optimal input experiment for each criterion is marked with red vertical curves.

the Shannon entropy for the same formulation (10) cannot be reduced to a closed-form expression and requires numerical integration

$$x_H^* = \underset{x}{\operatorname{argmax}} \left[\int_{\mathbb{Y}} \left(\prod_i \mathcal{N}_i(y_n | f_i(x), \sigma_i^2) \right) \left(\sum_i \log(\sqrt{2\pi}\sigma_i) + \frac{\left(\bar{f}(x) - f_i(x)\right)^2}{\sigma_i^2} \right) \mathrm{d}y \right]. \tag{22}$$

For a system with m models with an input space of n dimensions and an integration grid of p intervals, calculating surprisal (21) for a single data point has a computational complexity of $O(m)$. In comparison, Shannon entropy (22) requires $O((2mp)^n)$ calculations.

Furthermore, it is clear that surprisal is the variance of the weighted mean, with each term weighted by the empirical variance of the corresponding model. In comparison, the standard variance optimality criterion is

$$x_V^* = \underset{x}{\operatorname{argmax}} \left[\sum_i \left(\tilde{f}(x) - f_i(x) \right)^2 \right] \tag{23}$$

where $\tilde{f}(x) = \frac{1}{I} \sum_i f_i(x)$ is the unweighted mean. This agrees with the original motivation—the optimality criterion measures how well an experiment is expected to create disambiguation between the candidate models. However, unlike standard variance, surprisal leverages their predictive abilities to emphasize disagreement between highly predictive models while marginalizing the contributions of poor candidate models.

Returning to the three candidate model example in Section III-A, the measures of entropy (22), surprisal (21), and variance (23) are calculated and normalized between 0 and 1 for comparison in Fig. 3. First, analyzing the candidate models, the cubic and sinusoid models are accurate with small residuals, while the linear model has relatively large residuals. Thus, there is more information to gain from making the cubic and sinusoid models disagree since the linear model predicts that the data is dominated by noise.

Algorithm 1 Non-parametric optimal experiment design algorithm

```
input:  candidate models - f_i
        data set - (x_n, y_n)
output: optimal experiment - x

for each model:
  calculate empirical variance (18)

initialize experiment_population

until allotted computational effort is reached :
  divide population into random pairings
  for each pair in experiment_population :
    recombine parents to generate two offsprings
    mutate each offspring
    calculate models' prediction for each offspring (20)
    calculate surprisal for each offspring (21)
    determine which offspring is most similar to which parent
    for both similar(offspring, parent) pairs:
      if offspring.surprisal >= parent.surprisal :
        replace parent with offspring

optimal_experiment = highest_surprisal(experiment_population)

return optimal_experiment
```

Consequently, Fig. 3 shows that the relative maxima of the entropy and surprisal criteria state the optimal experiment is $x^* = 4.2$, as it best differentiates whether the cubic or sinusoid model is a better representation. On the other hand, the variance criteria states that $x^* = 2.4$ is optimal as that creates the greatest disagreement between the predictions of all three models, with no sensitivity to the accuracy of the models. Furthermore, surprisal is qualitatively similar to entropy with three local maxima and a similar shape, while the variance curve bears little resemblance to entropy.

The non-parametric optimal experiment design process is outlined in Algorithm 1. The algorithm is a genetic algorithm that maximizes surprisal and uses deterministic crowding for selection [30]. Note that maximizing surprisal is the primary goal and can be obtained by applying alternative evolutionary algorithms or optimization approaches.

IV. Coevolutionary Active Learning Testbench

This section describes the testbench used to measure the performance of the proposed experiment design within an active learning framework. The section is organized according to the processes outlined in Fig. 1; model inference, experiment design, and data collection are discussed in Sections IV-A, IV-B, and IV-C, respectively. The section is concluded with a discussion of the procedure.

The coevolution aspect of this approach is a product of how the active learning processes are implemented. The model inference process is implemented as a symbolic regression algorithm that creates genetic programs to model the dynamic data set, while the experiment design process is a genetic algorithm that finds new data which optimizes the surprisal given the current models. This type of implementation produces dynamics where the performances of both evolutionary algorithms are interdependent.

A. Symbolic Regression via Genetic Programming

For model inference, we used symbolic regression, a genetic programming algorithm 1. A process flowchart is shown in

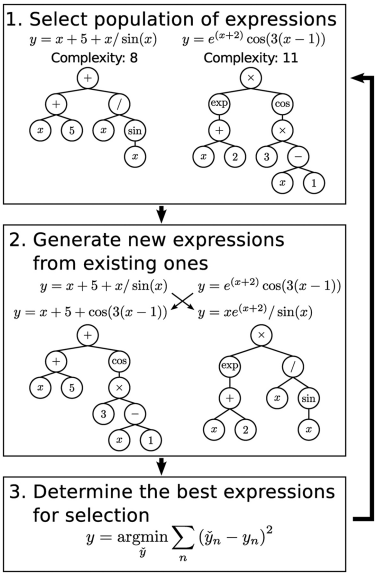


Fig. 4. Process flowchart for symbolic regression.

Fig. 4. Symbolic regression was chosen for variety of reasons. First, it is a heuristic algorithm and, consequently, highly sensitive to the data set. As surprisal works with arbitrary regression models, one that is more sensitive to data sets provides a better measure of robustness.

Next, symbolic regression provides an automated method for exploring various forms based on their predictive accuracy. In comparison, alternative inference algorithms focus on optimizing the parameters of a single form. As a multiple candidate models are required, manually selecting different model types creates a bias in comparing active learning algorithms. Symbolic regression is able to represent arbitrary expressions by arranging fundamental building blocks and generate new models in a heuristic fashion.

A continuously updated record of the best performing models is kept, where performance is measured according to a Pareto optimal approach based on accuracy and complexity—for every model in the record, no other models were found that are simultaneously more accurate, as determined by (19), and less complex, defined as the size of the expression. These Pareto optimal expressions are used as the candidate models.

A publicly available symbolic regression engine, Eureqa, was used for this testbench [32]. Symbolic regression was executed for 10^9 evaluations, a conservative estimate of the computational effort needed for convergence, which required approximately five core minutes on a 2.8 GHz Intel processor. The building block operators were $\{+, -, \times, /, \sin, \cos, \exp\}$ and the default parameters were used: population size of 64 solutions, crossover probability of 0.7, and mutation probability of 0.03. The performance of the algorithm was not found to be sensitive to these parameters.

B. Experiment Design

Four experiment design policies were compared; entropy was not included due to its prohibitively long computational demands. Two of the policies are model-free and independent of any input, while the other two policies are model-based and leverage the candidate models as part of the design process.

1) Binary recursive design [Recursive]—This is the optimal method for a model-free design policy. The input space is recursively subdivided in half along each dimension. For 1D input with a domain [0,1], the binary recursive policy generates the following sequence of experiments: $\{0, 1, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}, \frac{1}{16}, \dots\}$.
2) Random design [Random]—This is a model-free design policy that consists of randomly selecting an experiment with a uniform probability.
3) Variance design [Variance]—This is a model-based design policy based where the optimality criterion is (23). This policy was included due to its popularity in related work [16], [20], [25], [26].
4) Surprisal design [Surprisal]—This is the proposed model-based design policy where the optimality criterion is (21).

Experiments were selected within predefined bounds for the input space. Although the framework is suitable for continuous inputs, the experiment selection was discretized into 257 uniformly spaced grid points per input. The discretization was chosen as a result of the performance metric discussed in Section IV-D.

The model-based policies are generally non-convex, which presents an optimization challenge. A simple genetic algorithm (Algorithm 1) was used that evolved bit strings that corresponded to the grid points. The genetic algorithm had a population size of 8, crossover probability of 0.7, and a mutation probability of 0.05. Since this optimization problem was relatively simple, the algorithm ran for 10^3 evaluations, which required approximately five core seconds on a 2.8GHz Intel processor.

C. Data Collection

Three different underlying models were selected based on their difficulty for symbolic regression algorithms. The underlying models generate data according to a symbolic expression, and thus, a perfect model inference is possible with this approach. These underlying models were adapted from popular baselines and the difficulty of these models for symbolic regression is well established [11], [33].

1) *1-D Exponential Sinusoid:* The exponential sinusoid is a 1-D problem with the expression $y = e^{0.1x^2}\sin(5.5x)$ and a complexity of 11 nodes. A plot of the function evaluated at the possible 257 experiments is shown in Fig. 5.

This expression is designed to be a simple inference problem. It is a 1-D problem with lower complexity than its counterparts. Furthermore, the function has global information content—any experiment is likely to provide meaningful information for model inference since the probability of generating a pathologically poor data set, such as one that consists of only zero crossings, is highly unlikely.

2) *1-D Wavelet:* The wavelet is a 1-D problem with the expression $y = e^{-(x-2)^2}\sin(5x)$ and a complexity of 15 nodes. A plot of the function evaluated at each of the possible 257 experiments is shown in Fig. 6

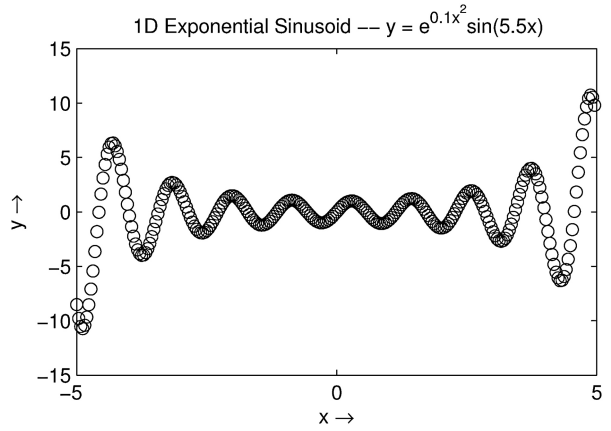


Fig. 5. Plot of the exponential sinusoid expression.

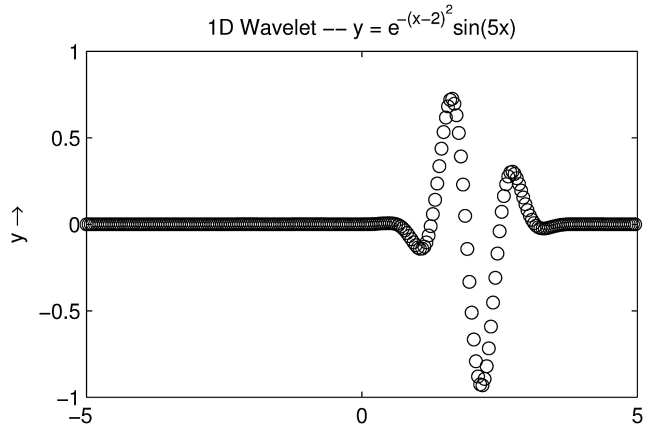


Fig. 6. Plot of the wavelet expression.

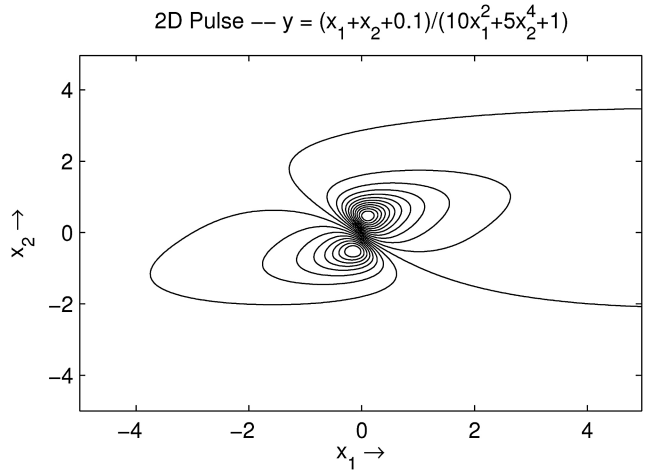


Fig. 7. Plot of the pulse expression.

Unlike the exponential sinusoid, the wavelet presents a significantly more challenging inference problem due to its local information content. The majority of the input space consists of near zero outputs, and thus it is trivial to obtain a pathologically poor data set. Much of the expression is characterized within a narrow input space, and thus a non-uniform sampling of the input space is required for efficient inference.

3) *2-D Pulse:* The pulse is a 2-D problem with the expression $y = \frac{x_1+x_2+0.1}{10x_1^2+5x_2^4+1}$ and a complexity of 24 nodes. A contour plot of the function evaluated at each of the possible 257×257 experiments is shown in Fig. 7.

Like the wavelet, the pulse has local information content and presents a challenge for experiment design to avoid deceptive data sets. The pulse also presents the additional problem of inferring a multidimensional function—the space of possible experiments grows exponentially with the number of inputs. This phenomenon is referred to as the curse of dimensionality and is often a crippling issue in learning algorithms.

Noise Corruption—Each of the data sets were corrupted with additive zero-mean Gaussian noise. Given that the sample variance of the total data set is σ_S^2 and the noise variance is σ_N^2, then the signal-to-noise ratio is defined as

$$SNR = 10\log_{10}\left(\frac{\sigma_S}{\sigma_N}\right). \tag{24}$$

Three treatments were considered: $\sigma_N^2 = 0$, or a noiseless output; $SNR = 20$dB, or when the signal is 100-fold stronger than the noise; and $SNR = 10$dB, or when the signal is 10-fold stronger than the noise.

D. Testbench Procedure

The testbench begins with three randomly-selected initial experiments. The active learning processes (see Fig. 1) are iteratively executed in a sequential order: first, models are inferred using the existing data set; this is followed by an experiment design using one of the four policies in Section IV-B; and finally, the experiment evaluated and the result is inserted into the data set. This computationally intensive algorithm is motivated by the scenario that building a data set is expensive, while computation is relatively cheap.

This iterative process is repeated 32 times, resulting in a final data set of 35 points. The number of experiments was chosen to illustrate the performance difference between the various experiment design policies, and was found to be a suitable balance between the asymptotic behaviors of perfect accuracy with infinite experiments and poor accuracy with zero experiments. The active learning algorithm is applied to the three expressions with three noise conditions each (see Section IV-C), resulting in nine treatments. Finally, for each treatment, the algorithm is repeated 12 times for statistical analysis with different initial experiments.

The performance is measured using the normalized mean squared error

$$E = \sum_n \frac{(f(x_n) - y_n)^2}{\sigma_S^2} \tag{25}$$

which effectively is the normalized discrete integrated error between the candidate model and ground truth. The normalized error is computed for all the candidate models and lowest error is reported.

V. Testbench Results

This section describes the results of the active learning testbench. First, the convergence of the design policies is analyzed, followed by a case study of designed data sets.

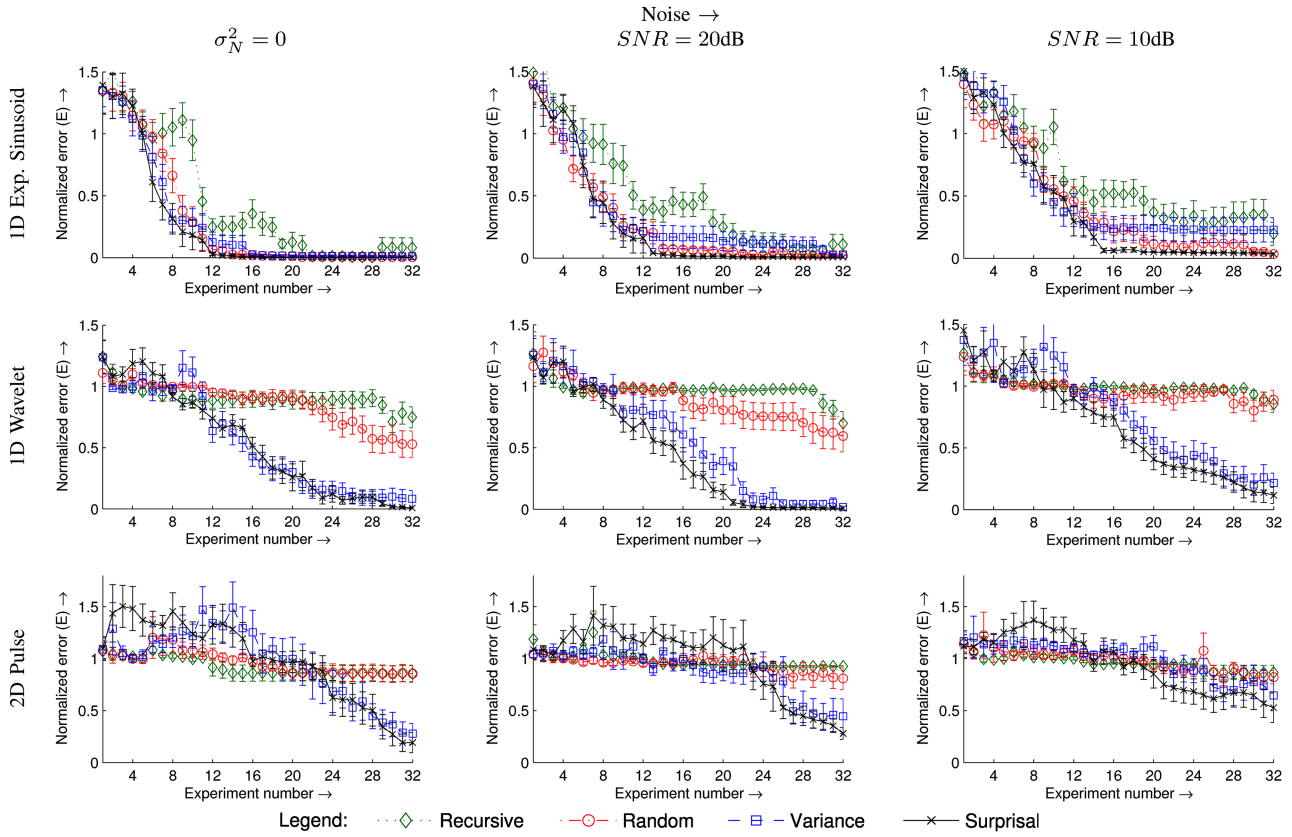


Fig. 8. Convergence plots for the three testbench expressions and the three noise levels. Error bars indicate standard error (n = 12).

A. Convergence Analysis

Convergence, or the normalized error compared to the size of the data set, is shown for all treatments, in Fig. 8. Overall, surprisal is the superior method as it converges at least as fast as any other policy for every treatment. In fact, on average, surprisal is able to achieve the same performance as the next best baseline with 6.8 fewer experiments, or a 21% reduction in the number of experiments. Furthermore, surprisal achieves significantly better convergence as the model inference problem increases in difficulty, either by increasing the noise corruption, increasing the input dimensions, or localizing the information content.

The noiseless exponential sinusoid expression was designed as the simplest non-trivial problem for every policy as it is a 1-D noiseless model with global information content. Surprisal performs just as well as any other policy, suggesting no penalty or overhead is incurred for simple problems. As the noise levels increase for the exponential sinusoid, surprisal provides significant improvements on convergence.

The wavelet expression compares the effect of information content. The best predictions in the model-free design policies, recursive and random, achieve a normalized error of approximately one, which is marginally better than predicting a constant mean or reducing the data to its first statistical moment. On the other hand, the model-based policies, variance and surprisal, are able to identify that the information content is not uniform and localize the data set for superior convergence. In addition, this difference is magnified with increasing noise levels.

The analysis of the pulse expression provides two key results. First, for this expression, the surprisal design policy is not severely affected the increase of dimensions. Although the convergence requires slightly more data than the 1-D examples, the increase in the number of experiments does not reflect the exponential growth in the input.

Next, an interesting phenomenon occurs for model-based policies on difficult inference problems: for small data sets, the normalized error suggests that policies obtain poor representations of the underlying expression. This phenomenon arises from the fundamental experiment design principle of determining experiments that cause disagreement in the candidate models. Consequently, this results in pathological sampling that promotes overfit models for small data sets.

However, in the long term perspective, this phenomenon is actually advantageous within the active learning paradigm. Overfitting occurs when an inferred model describes noise contributions instead of the underlying model, and thus, makes bad predictions with great certainty. As a consequence, overfit models are ideal for experiment design as they create large disagreements. These disagreements are then iteratively used to build better data sets and overfit models are quickly removed due to their inability to generalize to the new data. Thus, the model-based policies are able to leverage the existence of early overfit models to accelerate the data collection process, eventually resulting in superior models.

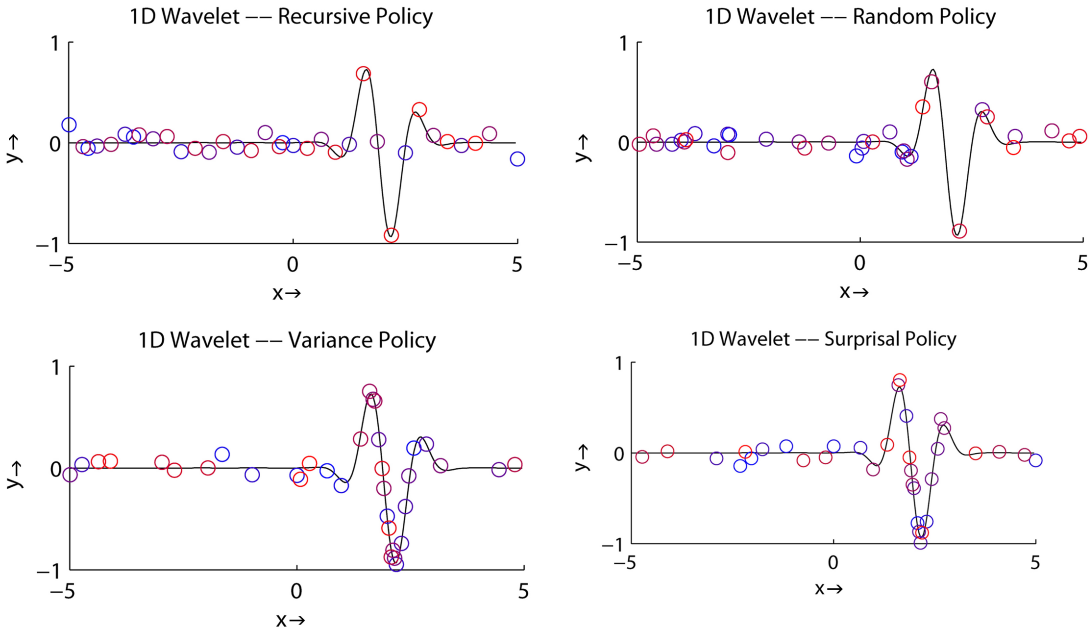


Fig. 9. High noise (SNR = 10dB) data set of the median performing run for each design policy overlaid on the ground truth expression. The final normalized error is indicated. Blue points indicate early experiments while red points indicate late experiments.

B. Case Study Analysis

In addition to analyzing convergence, the designed data sets themselves provide a significant resource for analysis. For the wavelet problem with the highest noise condition, SNR = 10dB, the data set of the median performing run for each policy is shown in Fig. 9.

The two model-free policies, unsurprisingly, generate data sets with a relatively uniform distribution; they are unable to focus on the characteristic region, resulting in poor performing model inferences.

Variance is able achieve an order of magnitude reduction in error by requesting a high density of experiments in the region that helps characterize the underlying model. In this run, note the limited noise corruption on the characteristic region; this suggests that despite the increased density, variance is still highly sensitive to noise.

In comparison, surprisal is able to achieve another order of magnitude reduction in error with a data set that is severely affected by noise, even in the characteristic region. Because the surprisal design policy accounts for the model prediction accuracy, it attempts to disambiguate overfit models, which is equivalent to determining the magnitude of noise. As such, it resamples at pivotal and specific locations, allowing it to deal with greater noise conditions than the variance design policy.

VI. Real-world Concrete Compression Strength Experiments

This section describes an application of coevolutionary active learning for a real-world problem. The goal is to infer a symbolic model of the concrete compression strength as a function of various mixing parameters [34], using experimental data that is publicly available in the UCI machine learning database [35]. The data set consists of 1030 measurements with 8 inputs, which describes the amount of mixture components as well as the settling time.

This example of a real-world setting illustrates the possible applications for coevolutionary active learning. For domains that are dominated by experimental testing such as materials engineering, gathering extensive data sets for traditional machine learning algorithms may be financially prohibitive. Active learning provides an alternative approach that is able to achieve similar performance levels without expensive parameter sweeps.

To simulate the active learning environment, the data is divided into three sets: 1) a potential training pool; 2) an active training set; and 3) a test set. The original 1030 point data set was reorganized to generate four unbiased samples: the original data set was randomly divided into four subsets and, in a cyclic fashion, one subset was designated as the test set while the remaining three formed the potential training pool. The experimental design and data collection processes then consisted of iterating through the potential training pool, finding the inputs that maximize the experiment design policy, and transferring that observation to the active training set for model inference.

Three treatments were selected for comparison. The first two treatments used active learning approaches with variance and surprisal as the experiment design policy, respectively. The protocol in Section IV was used and the initial active training set consisted of three data points selected randomly from the potential training pool. Model inference was executed via Eureqa with a computational limit of 10^9 evaluations and 128 experiments were designed. The third treatment used a passive machine learning approach that consisted only of model inference using all available 773 observations as the training set. The computational effort was extended to 3.2×10^{10} evaluations to compensate for the larger training

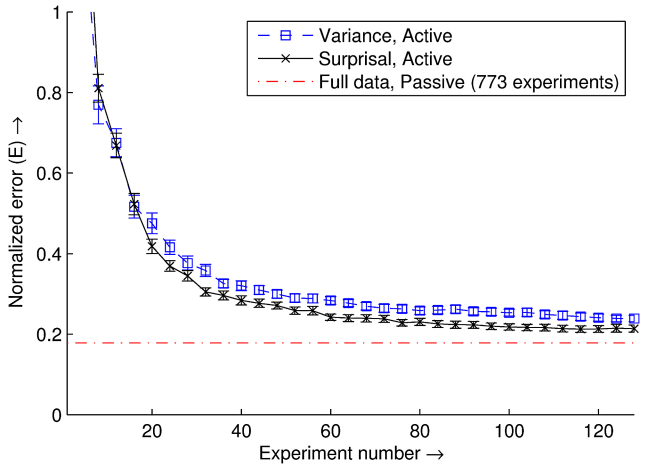


Fig. 10. Convergence plot for the three treatments. Note that the full data treatment is a passive learning approach and uses all available data for its training set while the variance and surprisal treatments dynamically add to their training set and the number of instances is equal to the experiment number plus three. Error bars indicate standard error (n = 32).

set. For added statistical analysis, each treatment was repeated eight times for each sample, resulting in 32 runs for each treatment.

The performance of the three treatments is shown in Fig. 10. It is clear that the surprisal experiment design policy is able to significantly outperform its variance-based counterpart. After 128 designed experiments, surprisal achieves a normalized error of $0.210 \pm .006$ while variance has an error of $0.234 \pm .007$. However, the metric more important than final error is the rate of learning—the surprisal policy is able to achieve the final error of the variance policy with just 59 experiments or with 53.8% less experiments. Surprisal is able to achieve this significant difference in performance because of its ability to account for the candidate models' accuracy, which is essential for complex, noisy systems.

Next, the surprisal design policy is able to asymptotically approach the error of the full data treatment, $0.178 \pm .003$. In fact, it is able to achieve 96.1% of the performance with just 16.6% of the data. For systems where data collection is limited, active learning is able to approach similar levels of performance with significantly fewer experiments.

However, all of the treatments fare noticeably worse than the neural network approach in the original work [34]. The difference in performance may be a result of three sources. First, it is possible that concrete compression strength is simply better suited for modeling by feedforward neural networks as opposed to symbolic expressions. Second, the active learning approach was simulated by restricting the experiment design to experiments that had recorded observations. It is possible that the active learning would be able to find better symbolic models if it was able to obtain specific observations of its choice. Finally, many of the important mixture properties, such as the water-binder ratio, are not drastically non-linear and have global information content [36]. From Section V, it is clear that active learning approaches, and surprisal in particular, perform best in local information content systems. Nonetheless, the surprisal learning approach does remarkably well compared to the more appropriate baseline of full data symbolic regression.

VII. Conclusion and Future Work

In this paper, we proposed using surprisal of the mean as an optimality criterion for non-parametric experiment design for coevolutionary active learning. As a coarse approximation to entropy, surprisal provides the primary benefit of being computationally lightweight. Compared on a testbench with three different underlying models and various noise conditions, surprisal outperforms the three baselines requiring 21% fewer experiments to achieve the same performance, particularly in situations where the underlying model has local information content, high noise conditions or high dimensionality. Furthermore, surprisal performs at least as well as any baseline. The resulting data sets are able to capture the underlying model for reliable model inference with as few as 32 data points. For the real-world concrete compression strength experiments, surprisal is able to achieve the same final error as variance with 53.8% less experiments and obtain 96.1% of the performance of the full data baseline with only 16.6% of the data. Future work will investigate how this framework can be extended to designing a sequence of experiments.

References

[1] J. Bongard, V. Zykov, and H. Lipson, "Resilient machines through continuous self-modeling," *Science*, vol. 314, no. 5802, pp. 1118–1121, 2006.
[2] J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," *Proc. Nat. Acad. Sci.*, vol. 104, no. 24, pp. 9943–9948, 2007.
[3] M. D. Schmidt and H. Lipson, "Distilling free-from natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, 2009.
[4] J. Bongard and H. Lipson, "Automating genetic network inference with minimal physical experimentation using coevolution," in *Proc. Genet. Evol. Comput. Conf.*, 2004, pp. 333–345.
[5] J. Bongard and H. Lipson, "Active coevolutionary learning of deterministic finite automata," *J. Mach. Learn. Res.*, vol. 6, pp. 1651–1678, Jan. 2005.
[6] J. Bongard and H. Lipson, "Nonlinear system identification using coevolution of models and tests," *IEEE Trans. Evol. Comput.*, vol. 9, no. 4, pp. 361–384, Aug. 2005.
[7] J. Doucette and M. I. Heywood, "GP Classification under imbalanced data sets: Active sub-sampling and AUC approximation," in *Proc. Eur. Conf. Genet. Programm.*, 2009, pp. 266–277.
[8] A. Berlanda, A. Sanchis, P. Isasi, and J. M. Molinda, "A general learning co-evolution method to generalize autonomous robot navigation behavior," in *Proc. IEEE Congr. Evol. Comput.*, 2000, pp. 769–776.
[9] S. Koos, J. B. Mouret, and S. Doncieux, "Automatic system identification based on coevolution of models and tests," in *Proc. IEEE Congr. Evol. Comput.*, 2009, pp. 560–567.
[10] E. D. de Jong and J. B. Pollack, "Ideal evaluation from coevolution," in *Proc. Evol. Comput.*, 2004, pp. 152–192.
[11] M. D. Schmidt and H. Lipson, "Coevolution of fitness predictors," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 736–749, Dec. 2008.
[12] M. D. Schmidt and H. Lipson, "Predicting solution rank to improve performance," in *Proc. Genet. Evol. Comput. Conf.*, 2010, pp. 949–956.
[13] D. L. Ly and H. Lipson, "Trainer selection strategies for coevolving rank predictors," in *Proc. IEEE Congr. Evol. Comput.*, 2011, pp. 2399–2406.
[14] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," in *Proc. Autonom. Agents Multi-agent Syst.*, 2005, pp. 387–434.
[15] B. Settles, "Active learning literature survey," Univ. Wisconsin-Madison, Madison, WI, USA, Comput. Sci. Tech. Rep. 1648, 2010,.

[16] D. Rubens, D. Kaplan, and M. Sugiyama, "Active learning in recommender systems," in *Recommender Systems Handbook*. Berlin, Germany: Springer, 2011.
[17] D. A. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Mach. Learn.*, vol. 15, no. 2, pp. 201–221, May 1994.
[18] C. H. Bryant and S. H. Muggleton, "Closed loop machine learning," Univ. York, Yorkshire, U.K., Tech. Rep. YCS 330, 2000.
[19] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *J. Mach. Learn. Res.*, vol. 5, pp. 225–291, Jan. 2004.
[20] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 129–145, 1996.
[21] M. Deisenroth, C. Rasmussen, and D. Fox, "Learning to control a low-cost manipulator using data-efficient reinforcement learning," in *Proc. Robot.: Sci. Syst.*, 2011.
[22] C. Guestrin, A. Krause, and A. Singh, "Near-optimal sensor placements in Gaussian processes," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 265–272.
[23] M. Gevers, "Identification for control: From the early achievements to the revival of experiment design," *Eur. J. Control*, vol. 11, pp. 1–18, 2005.
[24] P. Sebastiani and H. P. Wynn, "Maximum entropy sampling and optimal Bayesian experimental design," *J. Roy. Statist. Soc. B*, vol. 62, no. 1, pp. 145–157, 2000.
[25] A. F. Emery, A. V. Nenarokomov, and T. D. Fadale, "Uncertainties in parameter estimation: The optimal experiment design," *Int. J. Heat Mass Transf.*, vol. 43, no. 18, pp. 3331–3339, Sep. 2000.
[26] U. Forssell and L. Ljung, "Some results on optimal experiment design," *Automatica*, vol. 36, no. 5, pp. 749–756, May 2000.
[27] X. Bombois, G. Scorletti, M. Gevers, P. M. J. Van den Hof, and R. Hildebrand, "Least costly identification experiment for control," *Automatica*, vol. 42, no. 10, pp. 1651–1662, 2006.
[28] C. R. Rojas, J. Welsh, G. Goodwin, and A. Feur, "Robust optimal experiment design for system identification," *Automatica*, vol. 43, no. 6, pp. 993–1008, 2007.
[29] S. Simic, "Best possible global bounds for Jensen's inequality," in *Proc. Appl. Math. Comput.*, 2009, pp. 2224–2228.
[30] S. W. Mahfoud, *Niching methods for genetic algorithms*, Ph.D. dissertation, Dept. Comput. Sci., Univ. Illinois, Urbana-Champaign, IL, USA, 1995.
[31] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA, USA: MIT Press, 1992.
[32] M. D. Schmidt and H. Lipson. (2012). *Eureqa* [Online]. Available: http://creativemachines.cornell.edu/eureqa
[33] E. J. Vladislavleva, G. F. Smits, and D. den Hertog, "Order of nonlinearity as a complexity measure for models generated by symbolic regression via Pareto genetic programming," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 333–349, Apr. 2009.
[34] I. C. Yeh, "Modeling of strength of high performance concrete using artificial neural networks," *Cement Concrete Res.*, vol. 28, no. 12, pp. 1797–1808, 1998.
[35] A. Frank and A. Asuncion. (2010). *UCI machine learning repository* [Online]. Available: http://archive.ics.uci.edu/ml
[36] I. C. Yeh, "Analysis of strength of concrete using design of experiments and neural networks," *J. Mater. Civil Eng.*, vol. 18, no. 4, pp. 597–604, 2006.



Daniel Le Ly (S'09) received the B.A.Sc. (Hons.) degree in engineering science and the M.A.Sc. degree in computer engineering from the University of Toronto, Toronto, Canada, in 2008 and 2009, respectively, and the Ph.D. degree in mechanical engineering from Cornell University, Ithaca, NY, USA, in 2013.

His research interests include computational approaches for discovering underlying structure in complex natural and engineered systems, and their implications toward the understanding of cognitive processes.



Hod Lipson (M'98) received the B.Sc. and the Ph.D. degrees in mechanical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 1989 and 1998, respectively.

He is currently an Associate Professor with the Sibley School of Mechanical and Aerospace Engineering and Computing and Information Science, Cornell University, Ithaca, NY, USA. His research interests include computational methods to synthesize complex systems out of elementary building blocks, and the application of such methods to design automation and their implication toward understanding the evolution of complexity in nature and in engineering.