The ALAMO approach to machine learning

Zachary T. Wilson, Nikolaos V. Sahinidis

Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh PA 15213, USA

Abstract

ALAMO is a computational methodology for leaning algebraic functions from data. Given a data set, the approach begins by building a low-complexity, linear model composed of explicit non-linear transformations of the independent variables. Linear combinations of these non-linear transformations allow a linear model to better approximate complex behavior observed in real processes. The model is refined, as additional data are obtained in an adaptive fashion through error maximization sampling using derivative-free optimization. Models built using ALAMO can enforce constraints on the response variables to incorporate first-principles knowledge. The ability of ALAMO to generate simple and accurate models for a number of reaction problems is demonstrated. The error maximization sampling is compared with Latin hypercube designs to demonstrate its sampling efficiency. ALAMO's constrained regression methodology is used to further refine concentration models, resulting in models that perform better on validation data and satisfy upper and lower bounds placed on model outputs.

Keywords: Model selection, Parametric regression, Feature selection, Mixed-integer optimization

1. Introduction

The use of data science and machine learning techniques has become pervasive across disciplines of science and engineering to address problems associated with increasingly large and complex data sets. The abundance

---

*Corresponding author
Email address:* `sahinidis@cmu.edu` (Nikolaos V. Sahinidis)

of data available to analysts comes from a number of sources, including the prevalence of hard and soft sensors in process systems (Kadlec et al., 2009), the increased computational power available to perform complex simulations and store vast amounts of data, and the availability of modern social networking platforms to collect massive amounts of data from observational studies (Nayfeh et al., 2005; Wu et al., 2014). The problems presented to scientists and engineers often involve the optimization, prediction, control, and design of the system or process that generates a data set (Kleijnen & Sargent, 2000; Beck et al., 2016). Understanding the nature of the problem is critical in choosing an effective approach to accurately model the system at hand. Many problems in machine learning, including speech recognition, face recognition, and self-driving cars, are described by data sets that can have millions of samples or more and as many features (Chen et al., 2014; Nilsson, 2009). Accordingly, difficulty encountered in any analysis to follow can, in part, be ascribed to the pure volume of data (Friedman et al., 2013; Jacobs, 2009). Complexity is also inherent in trying to understand the system or process that generates data. In many cases, these systems may be well suited to the accurate prediction of an output given an input, but can be poorly suited to tasks involving optimization or design (April et al., 2003; Fu et al., 2005).

In order to accurately represent a system, any candidate model for the system must have the expressive power to represent the behavior of the observed response. The field of representation learning, sometimes called deep learning, is specifically concerned with generating candidate models that are fully capable of representing nonlinear relationships between inputs and outputs, modeling multi-scale effects, and even learning significant features in an unsupervised fashion; while still being able to train the prospective model to some training data efficiently (Bengio et al., 2013; LeCun et al., 2015). Recent efforts by companies including Google and Microsoft are specifically aimed at making break-through progress in the field of deep learning (Deng et al., 2013). Specifically, deep learning refers to a number of learning techniques that typically address *unsupervised* learning problems using different flavors of neural networks that contain many nodes in structured hidden layers. The learning problems being unsupervised simply means that the training data is not associated with any response vector. Instead, the goal of deep learning is to learn intrinsic relationships in the data set. This affords the opportunity to learn classes or aspects that are intrinsic to the data, and to make predictions about future data sets. As a result, deep learning methods can be generalized to both train a model using response vectors as well

as predict a response vector in the absence of one. Unsupervised learning problems almost exclusively consist of classification problems. Deep learning techniques have been made popular to a large audience due to their recent documented successes. However, their powerful unsupervised learning techniques can also be applied to other situations, such as fault detection in a chemical plant (Kulkarni et al., 2005).

Supervised learning techniques rely on a target vector of responses z to train a candidate model. When this response is an integer, the problem is called classification. It is common to reformulate any classification problems, regardless of the number of classes, to a number of binary classification problems. The remainder of this paper will focus on the regression setting, where $z \in \mathbb{R}^N$. Within this domain of supervised regression problems, approaches can be divided into either parametric or non-parametric regression techniques. *Non-parametric regression* methods include a number of different methodologies, including artificial neural networks (Haykin, 2009; Vapnik, 2013), support vector machines (Friedman et al., 2013; Scholkopf & Smola, 2001), and k-nearest neighbors (Altman, 1992) that function by placing different weights on each sample $i = 1, \ldots, N$, and predicting responses based on an appropriate averaging scheme. *Parametric regression* applies a different weight to each selected feature $j = 1, \ldots, p \leq k$, and predicts the responses as a function of the features. This function is a linear combination of various parameterizations of the inputs in the case of linear regression.

Although many inference techniques have recently been developed in the context of large data sets, they are routinely used for situations in which there is a dearth of both data points and number of features. In chemical kinetic parameter estimation, acquisition of data can be expensive (Hill, 1977; Buzzi-Ferraris & Manenti, 2010). As a result, parameter estimations are made in an attempt to identify unique parameter values from a minimum number of experimental data points. This places a high premium on the identification of simple models that do not overfit training data, the ability to incorporate information from first principles to bolster the model quality, and the intelligent acquisition of new data points if available. Different techniques have been developed to address these issues in the context of learning an appropriate regression or classification model.

Cozad et al. (2014) recently developed the ALAMO approach to machine learning, motivated by the need to obtain simple models from experimental and simulation data. Soon thereafter, the approach was extended to combine data-driven and theory-driven model building (Cozad et al., 2015). The pri-

mary purpose of this paper is to review, evaluate, and illustrate the ALAMO methodology and contrast it to existing techniques in the machine learning literature. Section 2 presents a review of machine learning approaches to model building. In Section 3, we discuss the core concepts of the ALAMO methodology. Subsection 3.1 describes the application of this methodology to learning simple models from a fixed data set. In Subsection 3.2, ALAMO's adaptive sampling methodology is reviewed. ALAMO's unique ability to enforce physical constraints on the response of the regression model is explored in Subsection 3.3. An illustrative example is presented in Section 4 to provide insights to the algorithms presented. Finally, extensive computations are presented on a set of 150 problems in Section 5, followed by conclusions in Section 6.

2. Review of current model building methodologies

In a typical linear least squares regression setting, a system's responses $z_i, i = 1, \ldots, N$, are approximated through linear combinations $\hat{z} = X\beta$ of a design matrix X, where $X_i = (x_{i1}, \ldots, x_{ip}), x_{ij} \in \mathbb{R}$ are the regressors and $\beta \in \mathbb{R}^p$ are the regression coefficients. Ordinary least squares (OLS) estimates for the regression coefficients can be obtained in closed form as $\hat{\beta}_{ols} = (X^t X)^{-1} X^t z$, and are obtained by minimizing the sum of squared residuals,

$$\text{SSR} = \sum_{i=1}^{N} \left(z_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

Linear models often incorporate an increasing number of regressors, either through acquisition of new independent variables or transformations of existing ones. As the basis set of regressors increases in size, the model will begin to describe noise and other aspects of the training data, a phenomenon commonly referred to as overfitting. Models that overfit data have a low bias, resulting from the increased degrees of freedom in the regression model, but high variance, meaning the predicted values can change dramatically with small changes in the training data (Friedman et al., 2013). Balancing the bias-variance trade-off and preventing overfitting in data-driven models is of fundamental importance across modeling techniques in order to produce models that will generalize to unseen data (Seber & Lee, 2012).

Avoiding overfitting is facilitated in practice by eliminating, or diminishing the effects of, superfluous regressors. A common method for accomplish-

4

ing this is the minimizing of an objective function composed of the sum of the SSR on a training set, and some q-norm of the regression coefficients, which quantifies the complexity of the linear model.

$$\text{BSS} = SSR + C\|\beta\|_q$$

$$\|\beta\|_q = \left(\sum_{j=1}^{p} \hat{\beta}_j^q\right)^{1/q}$$

If $q = 0$, this norm is called the l_0 (pseudo)norm, and is simply the number of nonzero regression coefficients $\|\beta\|_0 = \sum_{j=1}^{p} I(\hat{\beta}_j \neq 0)$, where $I()$ is the indicator function. The value of q determines a great deal about the properties of this problem; q values of 0, 1, and 2 are referred to as best subset selection, the lasso, and ridge regression respectively, and combinations of l_1 and l_2 norms is refereed to as elastic net regression (Friedman et al., 2010; Tibshirani, 1996; Friedman et al., 2013). The parameter C quantifies the relative trade-off between the complexity of a model and its error on a training set. Although minimization of the best subset problem has shown to produce more parsimonious models than the alternative problems, the use of the l_0 norm has been historically avoided due to the computational complexity associated with this combinatorial problem. However, integer optimization algorithmic advances combined with improved hardware have recently allowed for the best subset problem to be practically solved for problems with hundreds or thousands of regressors.

Accurate models can significantly reduce the expense associated with the design, optimization, and control of complex systems. For the purposes of developing accurate system models, standard multiple linear regression using the inputs to the system may be insufficient to develop accurate predictive models. A linear combination of the original inputs into the system can be inadequate to describe nonlinear relationships endemic in the underlying process. In this case, it is common to rely upon non-parametric techniques. These non-parametric techniques can build highly accurate representations of complicated nonlinear system responses (Cortes & Vapnik, 1995; Scholkopf & Smola, 2001); however, this accuracy comes at the expense of model complexity and a lack of interpretability. The larger problem is that of designing a hypothesis space of all possible model forms and parameter values that are capable of accurately representing the observed response. This fundamental question is intricately tied to the field of representation learning,

5

although some efforts to identify optimal nonlinear parametric transformations have been made (Breiman & Friedman, 1985).

The goal of building an inference-based model may not only be prediction accuracy, but also construction of a model that is informative of the underlying process. It is common for a significant amount of time and effort to be spent on the development of a nonlinear regression model from first principles. In the absence of insight or experience with a given domain, a standard set of parameterizations may be used as is the case in the standard polynomial regression to construct a new set of features. Although the new set of features will undoubtedly exhibit a high degree of correlation and may still not contain an adequate representation of the underlying process, performing model selection on the new set of features can produce models that are simple, interpretable, and highly accurate (Koza, 1994; Cozad et al., 2014). Even if predictive accuracy is of the utmost importance, a correctly specified parametric model will have a smaller variance than an optimal non-parametric model due to their flexibility (Shalizi, 2016). The expanded set of regressors can accurately represent nonlinear system responses, but will be prone to overfit the training data. This motivates the need to select the best subset of regressors in a multiple linear regression (Miller, 2002).

In linear models, preventing overfitting is often associated with finding the best subset of regressors. Best subset selection, also known as model selection or feature selection, has long been an active field in statistics (Hocking, 1976) and has received recent attention in the fields of machine learning and optimization (Bertsimas & Shioda, 2007; Liu & Motoda, 2007; Konno & Yamamoto, 2009; Cozad et al., 2014; Miyashiro & Takano, 2015; Bertsimas & King, 2015; Park & Klabjan, 2013). Exhaustive search algorithms, the most popular of which is Leaps-and-Bounds (Furnival & Wilson, 1974), were first proposed to address the subset selection problem. A major advantage of these exhaustive methods is that they can provide a number of viable models, all of which could perform adequately on unseen data. This small selection of models can then be validated on a test set, or through cross validation.

Due to the combinatorial nature of the search for the best subset, which is known to be NP-hard (Amaldi & Kann, 1998), exhaustive search methods do not scale well with the size of the problem. A popular implementation of this algorithm (Leaps and bounds package in R) is limited to only 32 regressors. This computational complexity results from exhaustive search algorithms as well as other embedded techniques, in which model selection

6

and parameter estimation are performed simultaneously (Guyon & Elisseeff, 2003). As a result, these algorithms are often passed over in favor of filter or wrapper methods. *Filter methods* involve a one-pass ranking of all features, and the discarding of any features that do not satisfy the chosen filter. *Wrapper methods* actively add and remove possibilities from the candidate model, often in a greedy fashion (John et al., 1994; Kohavi & John, 1997). Many wrapper methods can be generalized beyond the context of best subset selection in multiple linear regression, but are also directly applicable to this problem. Popular techniques include forward selection, backward elimination, and combinations of the two which fall under the umbrella term stepwise regression (Beck & Arnold, 1977; Miller, 2002). Stepwise heuristics for finding the best subset involve adding or removing regressors in an iterative fashion until a termination criterion is met. These stepwise heuristics typically produce good models, but offer no guarantee of optimality. Moreover, there exist a number of examples in the literature illustrating other shortcomings of these heuristics (Miller, 2002). Filter methods can also be applied to a number of model selection techniques including best subset selection. These methods are used to screen the features, routinely as part of a pre-processing procedure.

Regularization techniques also combat overfitting in linear models, and can facilitate model selection when regression coefficients are driven to zero (e.g., Tibshirani (1996)). These techniques rely on solving convex continuous optimization problems where the objective is a trade-off between error on the training set, and the magnitude of the regression coefficients:

$$\hat{\beta} = \text{argmin}_\beta \frac{1}{2}\|z - X\beta\|_2^2 + \lambda\alpha\|\beta\|_1 + (1-\alpha)\lambda\|\beta\|_2 \qquad (1)$$

where the parameter α determines the trade-off between the l_1 penalty and the l_2 penalty. If $\alpha = 1$, then Equation (1) is equivalent to the lasso problem (Tibshirani, 1996), and will facilitate subset selection as the l_1 penalty will drive regression coefficients toward zero. If $\alpha = 0$, the problem is equivalent to ridge regression (Hoerl & Kennard, 1970). For any value of α between 0 and 1, the problem is termed elastic net regression. Despite the fact that regularization does facilitate subset selection, Model (1) is not equivalent to the problem of best subset selection. Although regularization techniques do retain computational advantages, the use of mixed-integer programming (MIP) to facilitate subset selection has been shown to produce models that are smaller, and more accurate, than other techniques including

the lasso (Cozad et al., 2014).

A number of model fitness metrics have been developed from information theory and statistical theory to balance the bias-variance trade-off. Model fitness metrics that will be investigated in this paper include the corrected Akaike's information criterion (AICc) (Akaike, 1974), Mallows' Cp (Cp) (Mallows, 1973), the Bayesian information criterion (BIC) (Schwarz, 1978; Wang, 2009), the Hannan-Quinn information criterion (HQIC) (Quinn & Hannan, 1979), the risk inflation criterion (RIC) (Foster & George, 1994), and mean squared error (MSE) (Park & Klabjan, 2013). Different philosophies were followed in the derivation of each metric. For example, AIC seeks to minimize the Kullback-Leibler divergence between an underlying true distribution and the estimate from a candidate model, BIC attempts to maximize the posterior model probability, and Cp seeks to minimize the mean square error of prediction. The differing perspectives utilized in the derivations result in the identification of different best subsets of regressors. The statistical motivation of some model fitness metrics, specifically utilizing Kullback-Leibler divergence as a means of model selection, has been thoroughly investigated in the literature (Burnham & Anderson, 2001, 2002; Miller, 2002). All metrics account directly for the inclusion of an explanatory variable when quantifying model complexity, and this makes them all highly amenable to MIP formulations. Standard assumptions made for linear regression, namely the residuals being normally distributed with zero mean and a specified variance σ^2, result in similar functional forms across fitness metrics. These metrics are defined as follows.

$$\text{AIC}_c = N \ \log \left(\frac{1}{N} \sum_{i=1}^{N} (z_i - \mathbf{X}_i \beta)^2 \right) + 2p + \frac{2p(p+1)}{N-p-1} \tag{2}$$

$$\text{HQIC} = N \ \log \left(\frac{1}{N} \sum_{i=1}^{N} (z_i - \mathbf{X}_i \beta)^2 \right) + 2p \log(\log(N)) \tag{3}$$

$$\text{MSE} = \frac{\sum_{i=1}^{N} (z_i - \mathbf{X}_i \beta)^2}{N-p-1} \tag{4}$$

$$\text{C}_P = \frac{\sum_{i=1}^{N} (z_i - \mathbf{X}_i \beta)^2}{\hat{\sigma}^2} + 2p - N \tag{5}$$

$$\text{BIC} = \frac{\sum_{i=1}^{N} (z_i - \mathbf{X}_i \beta)^2}{\hat{\sigma}^2} + p \ \log(N) \tag{6}$$

$$\text{RIC} = \frac{\sum_{i=1}^{N}(z_i - \mathbf{X}_i\beta)^2}{\hat{\sigma}^2} + 2p \ \log(k) \tag{7}$$

In all model fitness metrics, N is the number of data points in the training set, $p \leq k$ denotes the number of nonzero regression coefficients, where k is the total number of regression coefficients. The parameter $\hat{\sigma}^2 = \frac{SSR_{ols}}{N-1}$ is an estimation of the residual variance, and is usually obtained from the mean square error of the full k-term model, which is an unbiased estimator. All metrics utilize the SSR, or l_2 error, to quantify fit on a training set, and penalize this fit directly based on the number of nonzero regression coefficients. Additionally, three of the six metrics are integer convex quadratic functions (5)–(7) and can be optimized directly through mixed-integer quadratic (MIQP) formulations. Direct optimization of the remaining metrics (2)–(4) would require mixed-integer nonlinear programming (MINLP). The derivation of the functional form of Bayes information criterion is obtained by utilizing the likelihood function $\hat{L} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(z-\hat{z})^2}{2\sigma^2}$ obtained by assuming the residuals of the model are normally distributed with zero mean, and variance estimated by $\hat{\sigma} = \frac{SSR}{N-1}$.

3. ALAMO

ALAMO is a regression and classification model learning methodology that builds simple, accurate surrogate models from experiments, simulations, or any other source of data using a minimal set of sample points. At the heart of ALAMO, is an integer-programming-based best subset technique that considers a large number of explicit transformations of the original input variables x. The model is then tested and improved using derivative-free optimization solvers that sample promising points in an adaptive fashion. The model building and adaptive sampling features constitute the core of ALAMO as seen in Figure 1. The process begins with an initial data set of N^{ini} points. A surrogate model is then built, and an adaptive sampling methodology referred to as error maximization sampling (EMS) is utilized to identify the next $i^{'} = N + 1 \cdots N^{ems}$ points. If the selected model satisfies some δ tolerance on the maximum observed error, then the algorithm is allowed to terminate; otherwise, the points identified by EMS are appended to the training set.

This section contains a summary of the core components of the computational implementation of the ALAMO methodology. In the usual setting, ALAMO can be cast into the domain of linear parametric regression, where
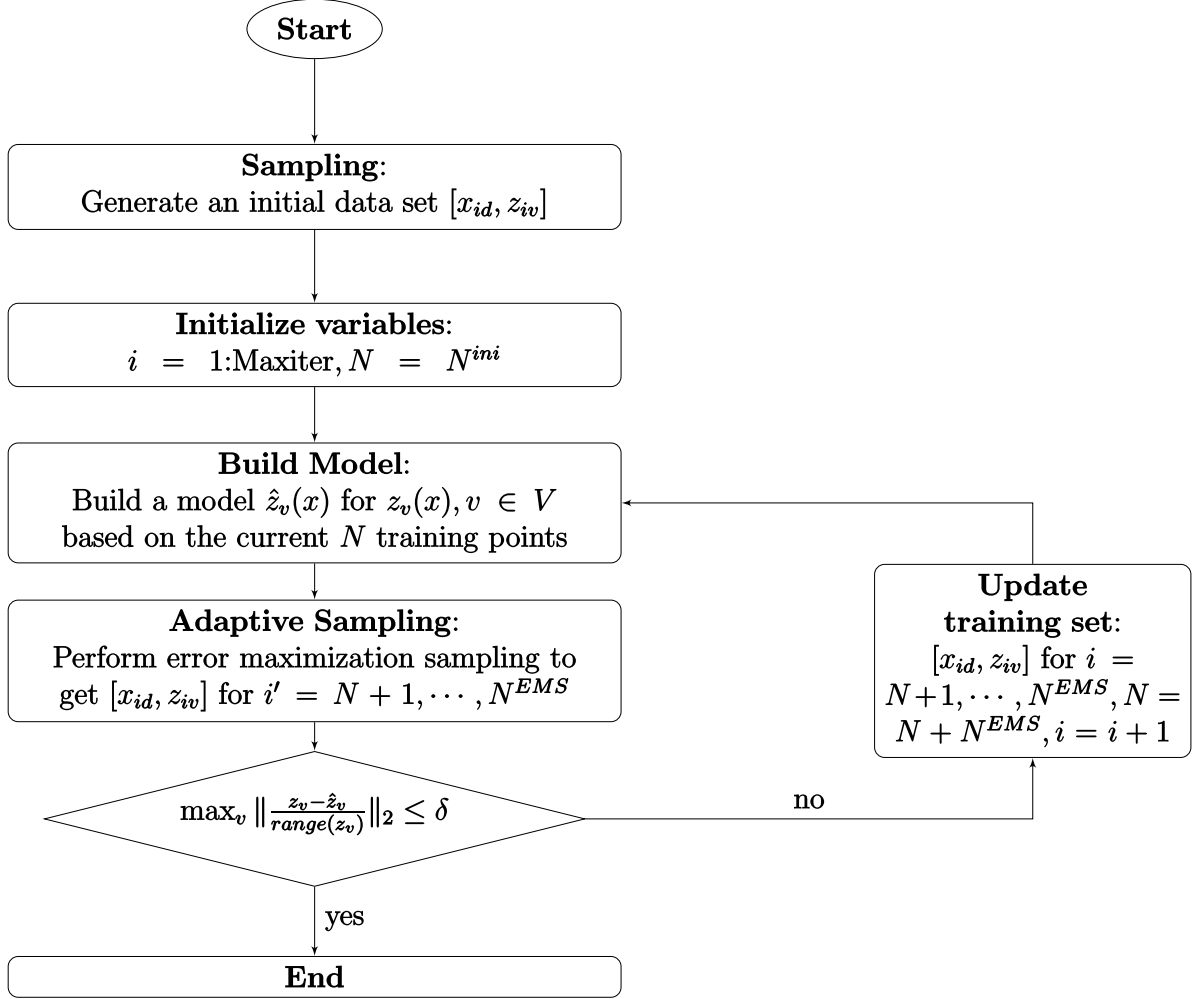
```
                          ┌─────────────┐
                          │    Start    │
                          └──────┬──────┘
                                 │
                                 ▼
              ┌──────────────────────────────────────┐
              │           Sampling:                  │
              │ Generate an initial data set [x_id, z_iv] │
              └──────────────────┬───────────────────┘
                                 │
                                 ▼
              ┌──────────────────────────────────────┐
              │       Initialize variables:          │
              │   i = 1:Maxiter, N = N^ini            │
              └──────────────────┬───────────────────┘
                                 │
                                 ▼
              ┌──────────────────────────────────────┐
              │           Build Model:               │
              │ Build a model ẑ_v(x) for z_v(x), v ∈ V│ ◄──────┐
              │ based on the current N training points│        │
              └──────────────────┬───────────────────┘        │
                                 │                             │
                                 ▼                             │
              ┌──────────────────────────────────────┐   ┌─────────────────┐
              │       Adaptive Sampling:             │   │     Update      │
              │ Perform error maximization sampling to│   │  training set:  │
              │ get [x_id, z_iv] for i' = N+1,···,N^EMS│   │ [x_id, z_iv] ...│
              └──────────────────┬───────────────────┘   └─────────────────┘
```


Figure 1: Flowchart of model building algorithm utilized by ALAMO

The decision diamond contains:

$$\max_v \left\| \frac{z_v - \hat{z}_v}{range(z_v)} \right\|_2 \leq \delta$$

with **no** branch leading to the Update box and **yes** branch leading to **End**.

The **Build Model** box:

Build Model:
Build a model $\hat{z}_v(x)$ for $z_v(x), v \in V$
based on the current N training points

The **Sampling** box:

Sampling:
Generate an initial data set $[x_{id}, z_{iv}]$

The **Initialize variables** box:

Initialize variables:
$i = 1$:Maxiter, $N = N^{ini}$

The **Adaptive Sampling** box:

Adaptive Sampling:
Perform error maximization sampling to
get $[x_{id}, z_{iv}]$ for $i' = N + 1, \cdots, N^{EMS}$

The **Update training set** box:

Update training set:
$[x_{id}, z_{iv}]$ for $i = N+1, \cdots, N^{EMS}, N = N + N^{EMS}, i = i + 1$

Start and **End** terminal nodes.

the columns of the design matrix are composed of explicit parameterizations of the original inputs into the system.

Any parametric regression methodology relies on explicit parameterizations of the system inputs to accurately describe any nonlinear behavior observed in the response data. A typical set of nonlinear transformations may be comprised of a number of transformations, including monomial, binomial, ratio, exponential, logarithmic, and trigonometric. Additionally, more complicated nonlinear functional forms such as a sigmoid, Arrhenius, and Gaussian relationships can be accommodated if the regression parameter in the nonlinear relationship is fixed and made to serve the role of a hyper parameter. Although this paper will focus on ALAMO's ability to produce parametric regression models, the addition of any distance function such as the Gaussian function will extend this methodology into the realm of non-parametric regression.

3.1. Learning simple surrogates

In this section, a MIP formulation for the optimization of the model fitness metrics discussed to facilitate best subset selection in linear regression is examined. As in the usual regression setting described before, we begin with a set of N training points; where each data point $(i = 1, \ldots, N)$ contains a set of input data $x_{id}, d \in D$, and a set of responses $z_{iv}, v \in V$. In the interest of simplifying the presentation, we will restrict our attention to the problem of a single response vector. A number of nonlinear transformations will be performed in order to populate a regression design matrix, $X_{ij}, j = 1, \ldots, k$.

In order to balance the bias-variance trade-off endemic to all data-driven modeling techniques, a MIP formulation will be used to optimize a model fitness metric. Despite nonlinearities present in some fitness metrics, they can all be optimized by parametrically minimizing the SSR subject to a maximum number of nonzero regression coefficients. In particular, any model fitness metric can be optimized through a series of cardinality constrained mixed-integer quadratic programs (CCMIQP) defined as follows:

$$\min_{r=1,\ldots,p} FM(r) \tag{8}$$

11

where

$$FM(r) = \min \sum_{i=1}^{N} (z_i - X_i\beta)^2 + C(r)$$

$$\text{s.t.} \sum_{j=1}^{p} y_j \leq r$$

$$-My_j \leq \beta_j \leq My_j, \quad j = 1, \ldots, p$$

$$y_j \in \{0, 1\}, \quad j = 1, \ldots, p$$

Above, FM is any model fitness metric (2)–(7), and $C(r)$ is a metric-dependent complexity penalty; $C(r)$ is constant for a specified model cardinality r. A cardinality constraint on the binary variables, y_j, ensures no more than r regressors are included in the model. The binary variables are used in conjunction with big-M constraints to model the inclusion or exclusion of a regressor. The value of the constant, M, in these big-M constraints is found using logic borrowed from the lasso, $(M = \sum_{j=1}^{p} |\hat{\beta}_{j,ols}|)$.

Although Formulation (8) can optimize all six model fitness metrics of interest, three metrics (5)–(7) will admit the following convex MIQP formulation for their direct optimization defined in the following formulation:

$$\left.\begin{array}{ll} \min & FM \\ \text{s.t.} & -My_j \leq \beta_j \leq My_j, \quad j = 1, \ldots, p \\ & y_j \in \{0, 1\}, \quad j = 1, \ldots, p \end{array}\right\} \tag{9}$$

Formulation (9) considers all 2^k possible regression models, including the null model, while Formulation (8) considers only $\binom{k}{r}$ models under the cardinality constraint. These two formulations are analogous, although by no means equivalent, to the penalized as well as constrained versions of the lasso problem. The constrained version of the lasso $\{\min_\beta SSR \| \beta \|_1 \leq T\}$, for some parameter T, may be more interpretable, as any $T < \|\hat{\beta}_{ols}\|_1$ will cause shrinkage in the regression coefficients. However, the use of the penalized version of the lasso $\{\min_\beta SSR + C\|\beta\|_1\}$, for some trade-off parameter C, is more common in practice due to a large number of computationally efficient implementations for solving this problem (Friedman et al., 2010). The issue of interpretability is easily avoided, as a maximum value for the parameter C where all regression coefficients are zero can be found directly from the KKT conditions of the lasso problem Friedman et al. (2010). It is common to start with this null solution, and find subsequent solutions for lower values of C

utilizing the previous solution, a process called warm-starting. Extending this back to the case of best subset selection, the CCMIQP Formulation (8) affords direct control on how many regression coefficients are nonzero, but may prove computationally inefficient if a large number of regressors are to be included in the optimal model. Likewise, the convex MIQP Formulation (9) may provide a computationally expedite solution of the same problem. However, unlike the penalized lasso, the fit-complexity trade-off found in (9) is informed from theory instead of exhaustively solved for using warm-starting.

3.2. Adaptive Sampling

Adaptive sampling, sometimes called active learning, refers to a means by which information about the underlying response surface can be acquired by querying the system at desired input levels. The ability to identify a data set that will perfectly train a model, or an optimal design of experiments (DOE), a priori does not exist. Instead, DOE methodologies could try to fill the space of the input variables in an optimal fashion. These techniques are referred to as single pass techniques and include factorial designs (Simpson et al., 2001), Latin hypercubes (McKay et al., 2000; Fang & Lin, 2003), and orthogonal arrays (Hinkelmann & Kempthorne, 2012). They generate a set of data points in the domain of the input variable, evaluate these points, and then move on to train the model. These methods are appealing in that they are often easy to implement and computationally efficient. However, the data set these methods produce may not be optimally suited to training a model. Since data is obtained in such a way that the input domain is filled, specific problem areas in the domain may be underrepresented, making the resultant models lack fidelity. Iterative techniques make use of the current model and data set to locate promising new sampling points. Exploration-based iterative methods seek to fill in the space of the input domain in a fashion similar to single pass methods. Exploitation-based methods utilize models in an adaptive fashion to locate data points near areas that are difficult to model. These may include regions of sharp nonlinearities or discontinuities.

The goal of adaptive sampling is to optimize the objective

$$\max_{x^l \leq x \leq x^u} \left(\frac{z(x) - \hat{z}(x)}{z(x)} \right)^2 \tag{10}$$

This formulation represents an error maximization sampling (EMS) approach. The algebraic form of $\hat{z}(x)$ is obtained through the use of ALAMO, or another surrogate modeling technique, but the algebraic form of $z(x)$ is not

known. The true black-box value of the objective is not known; therefore, a class of algorithms utilizing derivative-free optimization (DFO) must be used. A comparative study (Rios & Sahinidis, 2013) has investigated the use of a number of DFO solvers and discovered that SNOBFIT (Huyer & Neumaier, 2008) is currently a highly effective DFO solver for minimizing problems such as 10. The derivative-free solvers operate by estimating the error at new candidate points. If the estimated error indicates that the new point is in a region of model mismatch, then the point is added to the training set and a new model is built. The points to be added to the training set can also serve as a conservative estimate on the true value of (10). If this true error estimate is above the specified tolerance of δ the model is retrained using the updated data set as seen in Figure 1. If the true error estimate is below the tolerance, then the proposed model has converged on the final surrogate model of $z(x)$.

3.3. Constrained Regression

Data-driven models are routinely used when domain specific knowledge is available a priori that can be utilized to refine potential models. A fundamental form of restriction is composed of an upper and lower bound on the regression variables β. This leads to the usual constrained regression problem

$$\min_{\beta \in A} \; g(\beta; x, z) \tag{11}$$

This model determines the regression parameters $\beta \in \mathbb{R}^p$ that minimize a given loss function over an original set of regression constraints A. We wish to enforce an additional set of constraints

$$\Omega(\chi) := \{\beta \in \mathbb{R}^m : f[x, \hat{z}(x; \beta)] \leq 0, x \in \chi\} \tag{12}$$

where the function f is a constraint in the space of the modeled response(s) $\hat{z}(x)$ and the transformed predictors x, and χ is a nonempty subset of $\mathbb{R}^n$. The generality of Equation (12) extends beyond the ALAMO methodology. This equation can be used to reduce the feasible region for any regression analysis, including least squares, regularization techniques, best subset methods, and other nonparametric regression techniques. The general regression problem is now reformulated as

$$\min_{\beta \in A \cap \Omega(\chi)} \; g(\beta; x, z) \tag{13}$$

Previous work has employed a priori knowledge to reveal relationships between regression coefficients (Rao, 1965; Bard, 1974; Knopov & Korkhin,

14

2011). This may be of particular use in hierarchical regression tasks, where explicit rankings or importance of features may be known a priori. Inequality relationships between regression parameters have been used both in linear and nonlinear regression in engineering (Gibbons & McDonald, 1999), statistical (Judge & Takayama, 1966), as well as economic applications (Thomson, 1982). The sum of this previous work attempts to relate the regression coefficients through relationships obtained a priori. Instead, a more desirable constraint is one placed on the estimated response, that can be guaranteed across a domain of interest. The desire to constrain the response of the regression models leads to a complication in the realization that Eq. 12 is valid over the full domain of the inputs; requiring this equation to be enforced at infinitely many points, i.e., $\forall x \in \chi$. In the larger regression problem, this results in a problem with countably many (p) regression variables, but infinitely many possible constraints for every equation f that is applied to possible models. Semi-infinite programming (SIP) problems are optimization models that can be used to consider problems with a finite number of variables, but an infinite number of constraints. The core component of SIP involves solving $\max_{x \in \chi} f[x, \hat{z}(x; \beta)]$ to locate the point in the domain at which the maximum violation in the current surrogate model and the applied constraint occurs. This subproblem is significant because $\beta \in \Omega(\chi)$ if and only if $\max_{x \in \chi} f[x, \hat{z}(x; \beta)] \leq 0$. Therefore, this problem is in general nonlinear and nonconvex, thus requiring global optimization algorithms for solution.

Constraining a regression model's output in this fashion allows a modeler to: (a) restrict the bounds of the response, either individually or simultaneously restricting upper and lower bounds, (b) restrict the derivatives of the response to enforce model convexity or a monotonic relationship, and (c) enforce bounds over a domain outside of the training domain. Although regression-based models should not be used for extrapolation outside of the training domain, enforcing bounds on the response and certifying that a selected model satisfies these bounds can allow for a safer alternative to blind extrapolation. For a further discussion of ALAMO's constrained regression, see Cozad et al. (2015).

4. Illustrative example

A detailed look into one problem is provided to enable a more interpretable perspective on the models generated by ALAMO, and in order to

better facilitate the broader comparisons drawn later in the paper. There are two reactions in series $A \xrightarrow{k_a} B \xrightarrow{k_b} C$. We assume that the true system is governed by the following set of differential equations describing the transient mass balances of isothermal batch reactors:

$$\frac{dA}{dt} = -k_1 A$$
$$\frac{dB}{dt} = k_1 A - k_2 B$$
$$\frac{dC}{dt} = k_2 B$$

The true nonlinear functional forms of the concentration profiles of the three species, with associated kinetic rate constants $k_1 = 0.42$ and $k_2 = 0.97$, and initial conditions $A_0 = 1$, $B_0 = 0$, $C_0 = 0$ are as follows:

$$A = \exp(-0.42t)$$
$$B = \frac{0.42}{0.97 - 0.42} * (\exp(-0.42t) - \exp(-0.97t))$$
$$C = 1 - A - B$$

Three concentration profiles are generated from ALAMO using data corresponding to a LHS design of experiments of 20 data points. The three corresponding models generated by ALAMO, without utilizing the constrained regression feature, are as follows:

$$A_{alamo} = 0.44t + 0.00021 * t^{0.50} - 0.26t^2$$
$$+ 0.3 * t^{2.5} - 0.24 * t^3 + 0.101t^{3.5} - 0.021 * t^4 + 0.002t^{4.5}$$
$$B_{alamo} = 0.44t - 0.48t^2 + 0.3 * t^{2.5} - 0.08t^3 + 0.008t^{3.5}$$
$$C_{alamo} = 0.00021 * t^{0.50} + 0.22 * t^2 - 0.16 * t^3 + 0.093t^{3.5} - 0.021 * t^4 + 0.002t^{4.5}$$

The basis set utilized for this experiment is $X_j \in [t^{\pm[0.5,1,1.5,2,2.5,3,3.5,4,4.5,5]}$, $\log(t)$, $\exp(t)$, $1]$. As seen in Figure 2, this basis set is sufficient to obtain a perfect representation of all three concentration profiles.

In later comparisons, the time domain is sampled starting at later, nonzero, initial points. This is done for two primary reasons: first, it is often impractical to obtain concentration values from early process times due to time delays in the analytical process; secondly, the utility of constrained regression is more clearly demonstrated on a training set removed from 0, as the tendency of polynomial regression functions will be to have a sharp change in convexity in a region close, but outside, of the training set.

16
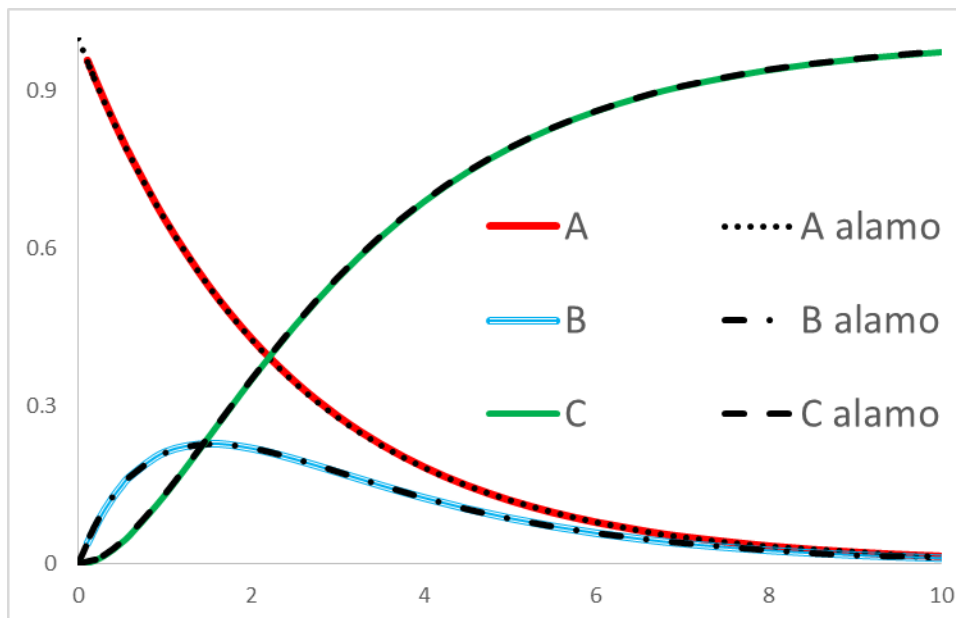
Figure 2: Concentration profiles of all three components for the two reactions in series

5. Computational experiments

The computational comparisons presented in this section are intended to demonstrate the following:

- The error maximization sampling (EMS) methodology is able to generate regression based models of a higher quality than those obtained from LHS.

- Constrained regression can be utilized in order to extend the range of regression-based models beyond the domain of the training data.

- For a given set of non-linear transformation, extending the range of the model comes at the expense of model accuracy on the training set; however, this problem is mitigated by the large number of candidate models available to ALAMO in the model building process.

In the following, the utility of the ALAMO methodology is demonstrated across a number of chemical reaction modeling problems. Data sets are generated from one of two possible reaction schemes. The concentration of

three species is modeled from a batch reactor with two first-order reactions in series:

$$A \xrightarrow{k_a} B \xrightarrow{k_b} C$$
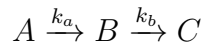
as well as a batch reactor with two competing first-order reactions:

$$A \xrightarrow{k_a} B$$

$$A \xrightarrow{k_b} C$$
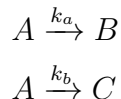
where the concentration of all species A, B, and C is a function of batch time t, where $0.6 \leq t \leq 10$, with the four associated kinetic rate constants k_a, where $0.4 \leq k_a \leq 3$ and k_b , where $0.9 \leq k_b \leq 2.1$ such that $k_a \neq k_b$. Five values are chosen for each parameter value via a uniform distribution and are exhaustively paired to yield 25 unique reactive systems. Boundary conditions for the reactors are available in the form of initial concentrations: $[A]_0 = 1, [B]_0 = 0, \text{and} [C]_0 = 0$. Training sets were sampled in one of two ways: using a Latin hypercube design of experiments for a specified number of points, or starting ALAMO with one random point over the domain $t_o \in [0.6, 10]$ and allowing EMS to sample until a tolerance δ was obtained. In each problem, nonlinear transformations were applied to the input variable t in order to create a basis set of 13 features, $X_j \in [t^{\pm[0.5,1,2,3,4]}, \log(t), \exp(t), 1]$, from which a surrogate model is generated. Details of the individual experiments are outlined in the appropriate subsection. As was demonstrated in 4 the simple kinetic schemes represented by the two reactions in series and parallel comprise a wide range of nonlinear behaviors to be represented with a simple set of nonlinear transformations of the single input variable t.

In order to facilitate a comparison between the model-building techniques detailed in the remainder of the paper, a basis for the comparison will be established. The model fitness metric Bayesian Information Criterion (BIC) in Eq. 6 will be used as the objective function of all subset selection problems. The model identified as optimal by this fitness metric will first be compared in terms of its performance on a validation set, quantified through the root mean squared error ($RMSE_{val}$). All validation data will consist of 1000 randomly chosen points across to domain of interest. A metric known as the error function (EF) will be used to facilitate the direct comparison of models tested on the same set of validation data. For model m, the error function is defined as:

$$EF_m = \frac{RMSE_{val,m}}{RMSE_{best}} \tag{14}$$

where $RMSE_{val,m}$ is the RMSE of model m applied to the validation set, and $RMSE_{best}$ is the lowest error obtained by any model in the comparison for that set of validation data. Consequently, the best modeling method will obtain $EF = 1$ while larger values are indicative of inferior solutions.

Two quality metrics will also be used to address the validity of the models in a practical sense. The root mean square error on the validation set of interest $RMSE_{val}$, as well as scaled comparison of the chosen model to an average of the training data. This metric is commonly known as R^2 when it is calculated on training data, and is sometimes called Q^2 when calculated on validation data. In order to remain consistent, we will adopt the nomenclature R^2_{val} and define the quantity as follows:

$$R^2_{val} = 1 - \frac{\sum_{i=1}^{n_{val}} \left(z_i - \hat{z}_i\right)^2}{\sum_{i=1}^{n_{val}} \left(z_i - \bar{z}\right)^2}$$

where $\bar{z}$ is the average of the training response data. Unlike the traditional R^2 value, the metric R^2_{val} is not bound below by 0 since a model that is optimal on a training set may generalize worse than the naive model to the validation set. Despite losing the lower bound of 0, R^2_{val} is still bound above by 1 as the error on the validation set can only be driven to 0. In the traditional R^2 metric, a value of 1 is often indicative of overfitting; however, measuring this metric on a set of independent validation data alleviates this concern. Instead, a value of 1 is only indicative of a substantially lower error on the validation set than $\bar{z}$, allowing it to be used as a check on model quality.

All simulations were performed on an Intel(R) Core(TM)2 Quad CPU Q9550 at 2.83 GHz. All mixed-integer and nonlinear optimization subproblems were solved using BARON 15.6 (Tawarmalani & Sahinidis, 2004; Sahinidis, 2015). In order to favor a tractable and parsimonious solution, ALAMO employs the use of a heuristic strategy to find an initially appealing linear model. Some problems may terminate in this heuristic phase if the found solution can be proven optimal, others move directly into the mixed-integer optimization of a chosen model fitness metric. In the following computations, ALAMO will utilize the fitness metric Bayesian information criterion, minimizing it by using Formulation (9). All ALAMO runs utilize Formulation (9) in order to guarantee the optimal model has been found.

5.1. Error maximization sampling in ALAMO

In this section, two experiments are performed to demonstrate the utility of the error maximization sampling methodology in ALAMO. First, ALAMO

is given one random point over the training interval, and the EMS methodology is allowed to proceed according to the flowchart in Figure 1 until a tolerance of $\delta = 0.0001$ is satisfied. The number of points required to satisfy this tolerance ranges from 14 to 46 across all 150 instances. The models generated by the EMS methodology are then compared to a number of models trained on data obtained from a Latin hypercube design of experiments (LHS). In the first comparison, a Latin hypercube design will be employed for exactly n_{EMS}, or the number of points required to satisfy the δ tolerance for each problem. Next, a Latin hypercube design will be used to generate data sets of the sizes $n \in \{20, 30, 40, 50\}$ (LHS-N). Comparing the models generated by the fixed sized training sets to those generated from training sets allowed n_{EMS} sample points serves to directly probe the quality of the solution obtained by the EMS methodology.

Figure 3 is a Dolan-Moreé performance profile (Dolan & Moré, 2002) that plots the error factor as defined in Equation (14) on the x axis against the fraction of the 150 problems that achieve the specified error factor. The best performance in this type of plot would be indicated by a vertical line at the origin, meaning one method outperformed all others in all problem instances. Figure 3 shows that the EMS methodology outperforms a LHS design consisting of the same number of points in the training set, n_{EMS}, indicated by the two unbroken lines, in 120 of 150 problems. This demonstrates the efficacy with which the EMS methodology is able to locate valuable new sample points that make a larger contribution towards the final model's predictive ability than those produced by an equivalently sized space-filling method. In the problems in which the LHS methodology outperforms the EMS methodology, two factors are usually present. Despite a difference in error calculated on a validation set, both models result in a high R^2_{val} indicating that the models are near similar in terms of predictive ability. Additionally, in some cases, the EMS methodology requires over 40 points to satisfy the δ tolerance. In these larger sized examples, the performance of the two methodologies is very similar due to the thorough sampling of the training domain. The exact number of points required by each problem can be seen in Figure 4, and the general statistics are contained in Table 1. As seen in this table, the EMS methodology requires a maximum of 46 sample points to converge with the specified $\delta-$tolerance, with 97% of problems requiring less than 30 samples to converge.

The results from these two methodologies are also compared against LHS designs of a fixed size across all problem instances. At face value, this might
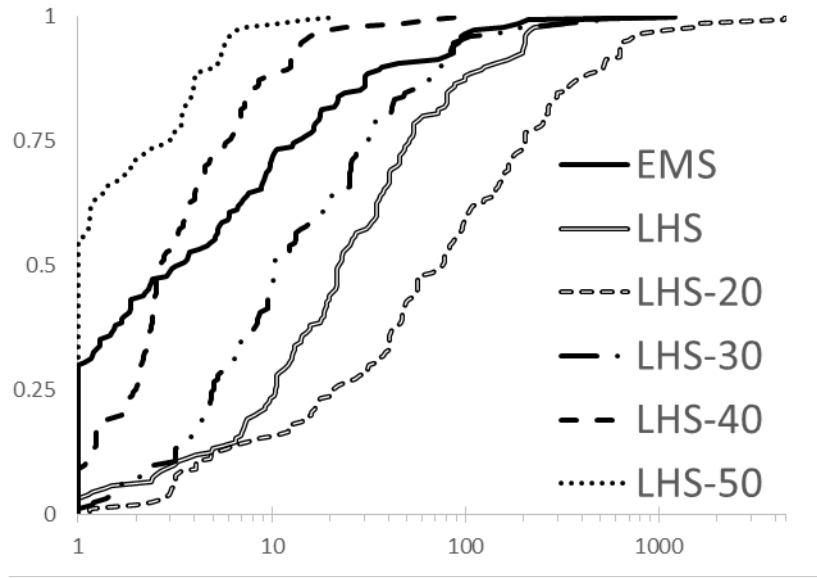
Figure 3: Fraction of problems solved vs. error factor

Table 1: Training set size statistics

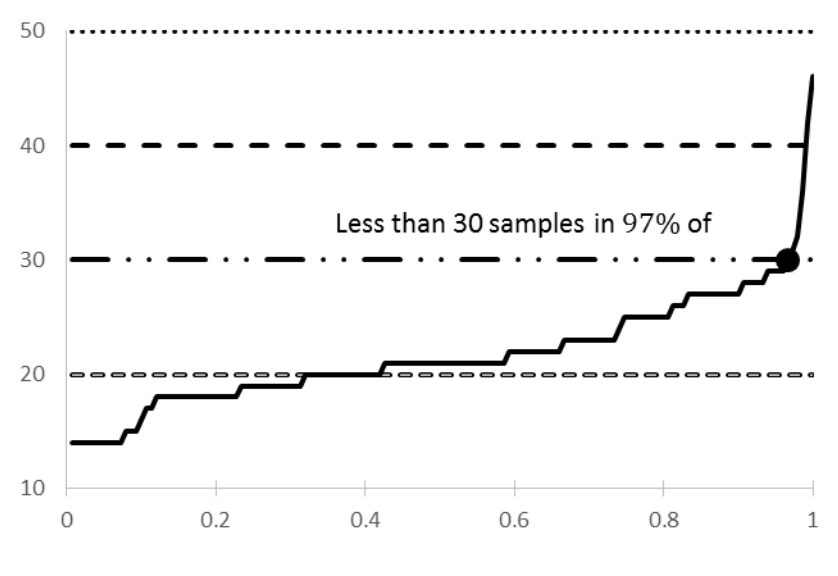| n_{EMS}				
Mean	Median	σ	Minimum	Maximum
21.7	21	4.9	14	46

Figure 4: Number of data points in training set vs. fraction of problems solve

seem like an unfair comparison as the static LHS methodologies routinely have access to many more data points than does the EMS methodology. However, the point of this comparison is to demonstrate the quality of the solution obtained by the EMS methodology versus those obtained by space-filling methodologies with access to many more training points on average. The two unbroken lines in Figure 3 provide a direct comparison of the EMS and LHS curves, while the dashed lines of Figure 3 serve to compare the quality of the solutions obtained by constant-sized LHS designs on much larger training sets.

The models chosen as optimal by ALAMO are compared first by their relative performance on a validation set consisting of 1000 uniformly distributed samples across the batch time domain. Figure 3 demonstrates that the EMS methodology achieves the best performing model in 45 out of 150 problems, only outperformed by the Latin hypercube design of 50 samples (LHS-50) which achieves the best performing model in 83 of 150 samples. Models trained on a design of 30 samples (LHS-30) achieve the best performance on the validation set in only 2 examples. The EMS methodology outperforms the LHS-30 methodology in 103 of 150 samples, despite utilizing less than 30 training points in 147 of 150 problems as seen in Figure 4.

The number of points in the final training set utilized by the EMS in

Table 2: Quality metrics calculated on validation set

Sampling method	R^2_{val}			RMSE_{val}		
	Mean	σ	Minimum	Mean	σ	Maximum
EMS	0.994	0.02	0.906	0.0012	0.004	0.038
LHS	0.988	0.04	0.547	0.0021	0.004	0.04
LHS-20	0.965	0.06	0.645	0.0063	0.01	0.083
LHS-30	0.998	0.002	0.988	0.0011	0.002	0.008
LHS-40	1	0	1	0.002	0.0003	0.001
LHS-50	1	0	1	0.00009	0.0001	0.0006

ALAMO can be seen in Figure 4. The average number of points in the final training set for all problems is $n_{EMS} = 21$; the number of points utilized by the Latin hypercube design remains constant across all problems. This adaptive sampling feature of ALAMO requires 8 iterations in the worst case of 46 data points sampled, and requires more than 40 samples in two problems and more than 30 samples in five problems out of 150.

The astute reader has observed that all sampling schemes compared in this section are implemented in ALAMO. In other words, ALAMO is capable of working exclusively with a user-specified data set, such as one obtained through LHS. Additionally, ALAMO is perfectly capable of generating a minimal sampling set through EMS or even combine LHS with EMS. The trend in the methodologies compared for given static data sets is clear and expected; more data in the training set results in a lower RMSE on a validation set. Therefore, it is important to remember that the solutions obtained by the EMS and LHS methodologies in Figure 3 are not on even footing in terms of the number of points in the training set. Instead, Figure 3 is intended to demonstrate the relative value of the data points obtained by the EMS methodology to those obtained by the space-filling EMS methodologies. Despite being out preformed by Latin hypercube designs consisting of more sampled points, Table 2 demonstrates the quality of the models produced regardless of the amount or quality of data provided to the wider ALAMO methodology. The lowest R^2_{val} achieved by any experimental design was $R^2_{val} = 0.547$ through a 14 points LHS design, and the lowest achieved by the EMS methodology was $R^2_{val} = 0.906$. Despite averaging only 21 points in the final data set, the EMS methodology was able to average a R^2_{val} of 0.994
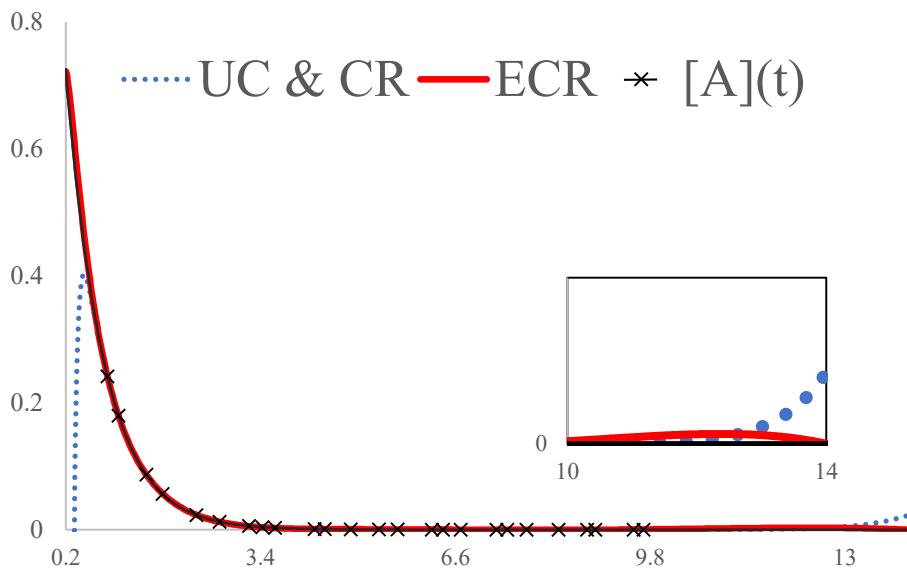
Figure 5: Visual comparison of the model produced through bound-extended constrained ALAMO and those produced by constrained and unconstrained ALAMO

across all 150 problem instances.

5.2. Constrained regression in ALAMO

This section will serve to demonstrate the utility of constrained regression in ALAMO. Three experimental settings will be examined: an unconstrained version of ALAMO (UC), a version of ALAMO in which the model output is bound below and above across the training domain (CR), and a bound-extended constrained version of ALAMO that seeks to extend the bound certification beyond the range of the training data (ECR). A non-negative lower bound, and an upper bound constrained by the initial concentration of species A are applied to all constrained regression problems. Training data for all 150 batch-reaction concentration problems will consist of 25 points from a Latin hypercube design of experiments. Figure 5 displays the results from the three modeling approaches on the concentration of species A, $[A](t)$, in the case of reactions occurring in parallel. In the illustrative example the CR methodology selected the same model as the UC methodology because it also satisfied the upper and lower bound restrictions over the training domain, $t \in [0.6, 10]$. However, this model did not satisfy the upper and lower bounds over the extended input range, $t \in [0.4, 14]$. Despite the lack
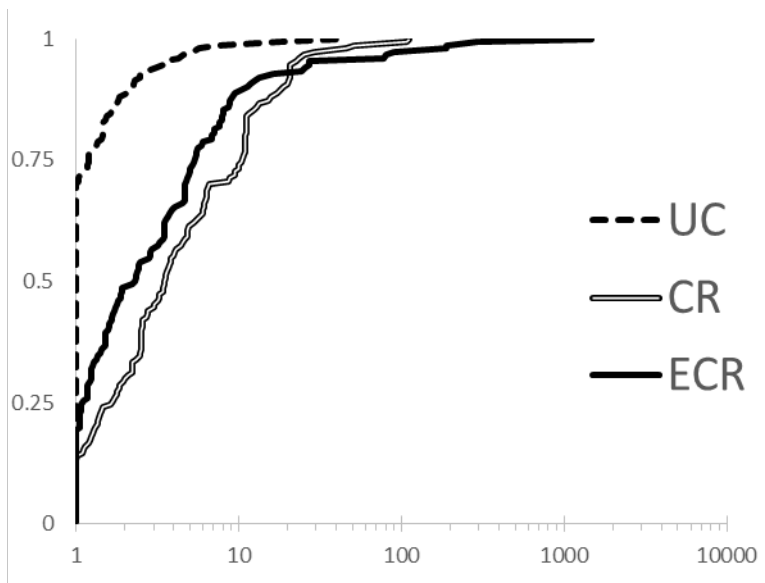
Figure 6: Fraction of problems solved vs. error factor, $t \in [0.6, 10]$

of training data present in the range of the extended bounds, a model that satisfies these constraints while maintaining accuracy on the original training domain is identified. The solution to either constrained regression problem will necessarily have a training error that is the same or larger than the solution to the unconstrained problem. However, this behavior cannot be generalized to any set of independent data, regardless of whether it is present inside or outside of the training domain. As such, the models produced by both CR and ECR may result in higher validation errors, even if the bounds they seek to maintain are appropriate and satisfied. This behavior, namely a decrease in validation error in the region outside the training set where CR is used to guarantee bounds at the expense of an increase in validation error in the original training domain, is observed throughout the synthetic reaction problem set, as seen in Figure 6 and Figurefig:crlb.

25
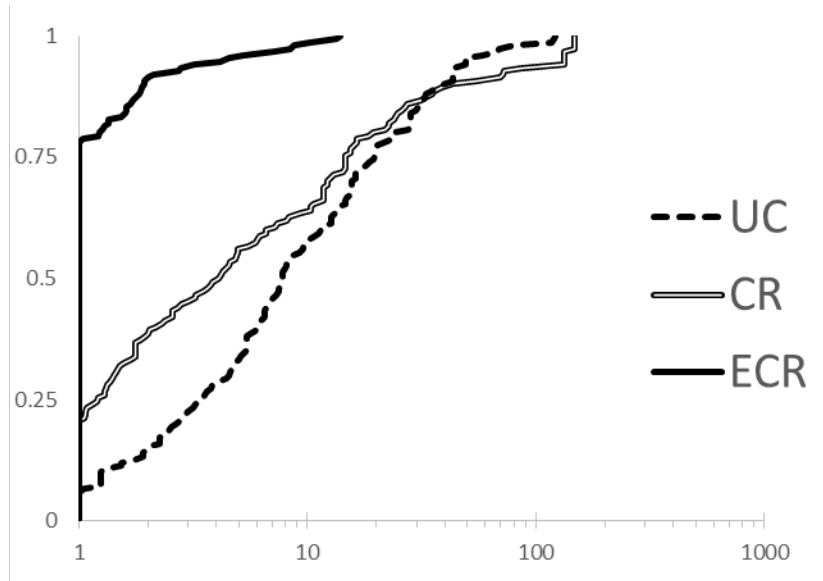
Figure 7: Fraction of problems solved vs. error factor, $t \in [0.4, 0.6)$

Table 3: Quality metrics calculated on validation set

		Lower extended domain			Training domain			Upper extended domain		
		UC	CR	ECR	UC	CR	ECR	UC	CR	ECR
RMSE_{val}	Mean	1.95	1.98	0.27	1.1E-03	3.0E-03	3.1E-02	4.4E-03	7.6E-03	4.1E-02
	σ	3.1	3.6	0.49	1.7E-03	4.3E-03	1.6E-01	7.1E-03	2.0E-02	1.7E-01
	Minimum	0.06	0.01	0.017	2.2E-05	9.4E-05	1.1E-04	4.4E-05	4.3E-07	9.3E-10
	Maximum	13.6	23	3.39	7.7E-03	2.8E-02	9.7E-01	3.9E-02	1.5E-01	1.0E+00
R^2_{val}	Mean	-111	-102	-5.2	0.99	0.99	0.93	0.99	0.99	0.91
	σ	326	382	24.8	0.0001	0.001	0.4	0.001	0.03	0.46
	Min	-2180	-3810	-203	0.999	0.995	-1.77	0.98	0.73	-2
	Max	0.983	0.996	0.998	1	1	1	1	1	1

In the training domain, $t \in [0.6, 10]$, as seen in Figure 6, models generated by UC have the lowest validation error in 103 problems, followed by ECR with 29, and CR performing best on 20 problems. Table 3 displays the quality metrics RMSE_{val} and R^2_{val} as applied to validation data obtained in the training domain. As expected, UC obtains the lowest average RMSE_{val} and highest average R^2_{val} values. Moreover, the high absolute values of R^2_{val} in this domain are indicative of high quality solutions in almost all cases.

Validation data with t values higher than those in the training domain demonstrate the utility of both constrained regression approaches. When the constrained regression feature is utilized to extend the upper bound of the time domain from 10 to 14, the UC approach and the CR approach perform best on 59 and 57 problems respectively, with the ECR approach achieving the lowest validation error on the remaining 36. Although it may seem counter intuitive that the models that obey physical constraints achieve a lower validation error, the satisfaction of the upper and lower bound constraints comes at the expense of a higher residual elsewhere in the model. This can be seen in a wide number of examples in Cozad et al. (2015) utilizing training data.

The utility of the ECR methodology can be seen most clearly in Figure 7, which shows it outperforming the other methodologies in 117 problems. The CR methodology performs best in 29 problems and the unconstrained models produced using the UC methodology perform best in the remaining 9. The quality metrics in Table 3 demonstrate the difficulty of generating an accurate model over this domain with the chosen 13 nonlinear transformations applied to the input variable t. However, Table 3 also demonstrates the ability of both the CR approach and ECR approach to generate accurate models in domains in which they have not been trained while satisfying physical constraints.

6. Conclusions

This paper presented ALAMO, a computational methodology and software developed to address the fundamental problem of leaning algebraic functions from data sets. A model fitness metric of choice, in this paper the Bayesian information criterion, is used to balance the bias-variance trade-off incumbent upon finding the best subset of a large number of explicit non-linear transformations of the process inputs used to construct a linear surrogate model. The model can be refined, as additional data are obtained in an adaptive fashion through the use of derivative-free optimization and the EMS methodology. This adaptive sampling methodology is able to make efficient use of small amounts of data, and outperforms space-filling models with larger training sets. An additional technique, referred to as constrained regression, where the model response is controlled through constraints on the regression parameters can be used additionally to enforce first-principles-based constraints on the model response. Forcing these models to obey physical constraints through constrained regression can result in models that are able to accurately predict data outside of the range in which the model was trained. The utility of the ALAMO methodology to generate simple algebraic models was demonstrated on a number of problems, and the additional performance imparted by using error maximization sampling as well as a priori domain knowledge through constrained regression was demonstrated.

7. Acknowledgments

or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*, 716–723.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor non-parametric regression. *The American Statistician*, *46*, 175–185.

Amaldi, E., & Kann, V. (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, *209*, 237–260.

April, J., Glover, F., Kelly, J. P., & Laguna, M. (2003). Practical introduction to simulation optimization. In *Simulation Conference, 2003. Proceedings of the 2003 Winter* (pp. 71–78). IEEE volume 1.

Bard, Y. (1974). *Nonlinear Parameter Estimation*. Cambridge, MA, USA: Academic Press.

Beck, D. A. C., Carothers, J. M., Subramanian, V., & Pfaendtner, J. (2016). Data science: Accelerating innovation and discovery in chemical engineering. *AIChE Journal*, *62*, 1402–1416.

Beck, J. V., & Arnold, K. J. (1977). *Parameter Estimation in Engineering and Science*. New York: John Wiley & Sons.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*, 1798–1828.

Bertsimas, D., & King, A. (2015). OR Forum—An algorithmic approach to linear regression. *Operations Research*, (pp. 1–15).

Bertsimas, D., & Shioda, R. (2007). Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, *43*, 1–22.

Breiman, L., & Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, *80*, 580–598.

Burnham, K. P., & Anderson, D. R. (2001). Kullback-Leibler information as a basis for strong inference in ecological studies. *Wildlife Research*, (pp. 111–119).

Burnham, K. P., & Anderson, D. R. (2002). *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. (2nd ed.). Springer-Verlag New York.

Buzzi-Ferraris, G., & Manenti, F. (2010). *Interpolation and Regression Models for the Chemical Engineer*. (1st ed.). Hoboken, NJ, USA: John Wiley and Sons.

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, *19*, 171–209.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273–297.

Cozad, A., Sahinidis, N. V., & Miller, D. C. (2014). Automatic learning of algebraic models for optimization. *AIChE Journal*, *60*, 2211–2227.

Cozad, A., Sahinidis, N. V., & Miller, D. C. (2015). A combined first-principles and data-driven approach to model building. *Computers and Chemical Engineering*, *73*, 116–127.

Deng, L., Li, J., Huang, J., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., Williams, J. et al. (2013). Recent advances in deep learning for speech research at Microsoft. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8604–8608). IEEE.

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, *91*, 201–213.

Fang, K. T., & Lin, D. K. (2003). Ch. 4. Uniform experimental designs and their applications in industry. *Handbook of Statistics*, *22*, 131–170.

Foster, D., & George, E. (1994). The risk inflation criterion for multiple regression. *The Annals of Statistics*, *22*, 1947–1975.

31

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*.

Friedman, J., Hastie, T., & Tibshirani, R. (2013). *The Elements of Statistical Learning*. (10th ed.). New York, NY, USA: Springer-Verlag.

Fu, M. C., Glover, F. W., & April, J. (2005). Simulation Optimization: A review, new developments, and applications. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, & J. A. Joines (Eds.), *Proceedings of the 2005 Winter Simulation Conference* (pp. 83–95).

Furnival, G. M., & Wilson, R. W. (1974). Regressions by leaps and bounds. *Technometrics*, *6*, 499–511.

Gibbons, D. I., & McDonald, G. C. (1999). Constrained regression estimates of technology effects on fuel economy. *Journal of Quality Technology*, *31*, 235–245.

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

Haykin, S. S. (2009). *Neural Networks and Learning Machines*. Pearson Upper Saddle River, NJ, USA.

Hill, C. G. (1977). *An Introduction to Chemical Engineering Kinetics and Reactor Design*. Hoboken, NJ, USA: John Wiley and Sons.

Hinkelmann, K., & Kempthorne, O. (2012). *Design and Analysis of Experiments, Special Designs and Applications*. Hoboken, NJ, USA: John Wiley and Sons.

Hocking, R. R. (1976). The analysis and selection of variables in linear regression. *Biometrika*, *32*, 1–49.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*, 55–67.

Huyer, W., & Neumaier, A. (2008). SNOBFIT–Stable noisy optimization by branch and fit. *ACM Transactions on Mathematical Software*, *35*, 1–25.

Jacobs, A. (2009). The pathologies of big data. *Communications of the ACM*, *52*, 36–44.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine learning: Proceedings of the eleventh international conference* (pp. 121–129).

Judge, G. G., & Takayama, T. (1966). Inequality restrictions in regression analysis. *Journal of the American Statistical Association*, *61*, 166–181.

Kadlec, P., Gabrys, B., & Strandt, S. (2009). Data-driven soft sensors in the process industry. *Computers and Chemical Engineering*, *33*, 795–814.

Kleijnen, J. P. C., & Sargent, R. G. (2000). A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research*, *120*, 14–29.

Knopov, P. S., & Korkhin, A. S. (2011). *Regression Analysis Under A Priori Parameter Restrictions*. New York, NY, USA: Springer-Verlag.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*, 273–3242.

Konno, H., & Yamamoto, R. (2009). Choosing the best set of variables in regression analysis. *Journal of Global Optimization*, *44*, 272–282.

Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, *4*, 87–112.

Kulkarni, A., Jayaramanan, V. K., & Kulkarni, B. D. (2005). Knowledge incorporated support vector machines to detect faults in Tennessee Eastman Process. *Computers and Chemical Engineering*, *29*, 2128–2133.

Leaps and bounds package in R (2016). Regression subset selection. Package 'leaps'. https://cran.r-project.org/web/packages/leapsl.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.

Liu, H., & Motoda, H. (2007). *Computational methods of feature selection*. London, England, UK: Chapman and Hall.

Mallows, C. L. (1973). Some comments on Cp. *Technometrics*, *15*, 661–675.

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *42*, 55–61.

Miller, A. (2002). *Subset selection in regression*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability (2nd ed.). London, England, UK: Chapman and Hall.

Miyashiro, R., & Takano, Y. (2015). Subset selection by Mallows' Cp: A mixed interer programming approach. *Expert Systems with Applications*, *1*, 325–331.

Nayfeh, A. H., Younis, M. I., & Abdel-Rahman, E. M. (2005). Reduced-order models for MEMS applications. *Nonlinear Dynamics*, *41*, 211–236.

Nilsson, N. J. (2009). *The Quest for Artificial Intelligence*. Cambridge, England, UK: Cambridge University Press.

Park, Y. W., & Klabjan, D. (2013). Subset selection for multiple linear regression via optimization. *Submitted for publication*, . http://dynresmanagement.com/uploads/3/3/2/9/3329212/regression_subset_selection_via_optimization.pdf.

Quinn, B. G., & Hannan, E. J. (1979). The determination of the order of an autoregression. *Journal of the Royal Statistical Socierty, Series B*, *41*, 190–195.

Rao, C. R. (1965). *Linear Statistical Inference and its Applications*. (2nd ed.). Hoboken, NJ, USA: John Wiley and Sons.

Rios, L. M., & Sahinidis, N. V. (2013). Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, *56*, 1247–1293.

Sahinidis, N. V. (2015). *BARON User Manual v. 15.6.5*. Available at http://www.minlp.com/downloads/docs/baron_manual.pdf.

Scholkopf, B., & Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*, 461–464.

Seber, G. A. F., & Lee, A. J. (2012). *Linear Regression Analysis*. (2nd ed.). Hoboken, NJ, USA: John Wiley and Sons.

Shalizi, C. R. (2016). *Advanced Data Analysis from an Elementary Point of View*. Cambridge, England, UK: Cambridge University Press.

Simpson, T. W., Poplinski, J. D., Koch, P. N., & Allen, J. K. (2001). Meta-models for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, *17*, 129–150.

Tawarmalani, M., & Sahinidis, N. V. (2004). Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, *99*, 563–591.

Thomson, M. (1982). Some results on the statistical properties of an inequality constrained least squares estimator in a linear model with two regressors. *Journal of Econometrics*, *19*, 215–231.

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

Vapnik, V. (2013). *The Nature of Statistical Learning Theory*. Springer Science & Business Media.

Wang, L. (2009). Wilcoxon-type generalized Bayesian information criterion. *Biometrika*, *1*, 163–173.

Wu, X., Zhu, X., Wu, G., & Ding, W. (2014). Data mining with big data. *IEEE transactions on knowledge and data engineering*, *26*, 97–107.