

On the use of supervised learning techniques to speed up the design of aeronautics components.

Gianluca Bontempi, Olivier Caelen
Machine Learning Group
Département d'Informatique
Université Libre de Bruxelles
Bruxelles, Belgium
email: {gbonte, ocaelen}@ulb.ac.be

Stephane Pierret, Cécile Goffaux
Numerical Methods and Optimization Group
CENAERO
Gosselies, Belgium
email: {stephane.pierret, cecile.goffaux}@cenaero.be

ABSTRACT

A crucial issue in the design of complex systems is the evaluation of a large number of potential alternative designs. A too expensive evaluation procedure can consequently slow down the search for good configurations, mainly in the case of high dimensional parameter spaces.

This paper discusses the use of machine learning techniques for speeding up the evaluation and the exploration of large design spaces. In particular, two supervised learning techniques, feedforward neural networks and lazy learning, are assessed and compared in the task of accelerating the design of a heat-pipe, a cooling device commonly used in aeronautics and electronics.

KEY WORDS

Design space exploration, machine learning, lazy learning

1 Introduction

The design of a complex product can be seen as a search problem in the space of design parameters which aims to maximize the quality of the product, measured by some given design objectives [12, 3].

This optimization problem is usually multidimensional and multicriteria, given the high number of design parameters and design objectives to be taken into account. The difficulty of the design problem is often made heavier by the fact that the optimizer has to rely on time consuming simulators to calculate the cost function to be optimized (Figure 1). This is often the case in aeronautics applications where the relation between the design parameters and the quality criteria is modeled by time consuming simulators.

Research in automatic design space exploration aims at defining efficient methods for exploring large design spaces. This can be achieved by

1. reducing the number of designs to be evaluated
2. reducing the time required to evaluate each design configuration.

This paper will address the second problem by exploring the use of supervised machine learning techniques [14] to speed up the evaluation of design configurations.

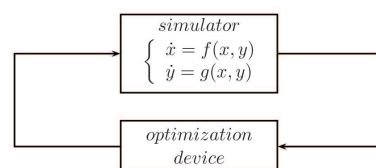


Figure 1: Design space exploration with physical simulator

Supervised learning techniques aim to model, on the basis of a finite set of observations, the relation between a set of *input* variables and one or more *output* variables, which are considered somewhat dependent on the input.

Our idea is to use machine learning techniques to model the relation between design parameters (i.e. the inputs) and design quality criteria (i.e. the outputs) (Figure 2).

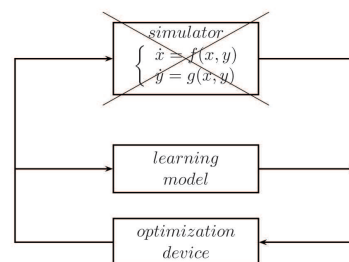


Figure 2: Design space exploration with machine learning approximator

The relation between the design parameter and the design quality criteria can be seen as an unknown multi-input multi-output mapping from the space of parameters to the space of quality criteria. This mapping is usually given by physical or expert-based models implemented by numerical simulators [19]. If we want to speed up the design exploration, it may be convenient to approximate the physical model by an input/output black box model, trained on a sufficient number of examples.

The approach we propose consists in (i) using the available simulator to assess the behavior of the system to be designed in some parameter configurations (ii) building a black-box model on the basis of the collected samples (iii) replace the physical model by our fast approximate simulator for design space exploration purposes.

The traditional approach to supervised learning is *global* modeling which describes the relationship between the input and the output with an analytical function over the whole input domain. Example of global models are regression linear models [16] and feedforward neural networks [6]. What makes global modeling appealing is the nice property that even for huge datasets, a parametric model can be stored in a small memory. Also, the evaluation of the parametric model requires a short program that can be executed in a reduced amount of time. Nevertheless, modeling complex input/output relations often requires the adoption of global nonlinear models, whose learning procedures are typically slow and analytically intractable. In particular, tasks like validation, feature selection and adaptive learning, which demand the identification of a large number of different models, appear to be computationally prohibitive in global nonlinear approaches [7].

For these reasons, in recent years, interest has grown in pursuing alternatives to global modeling techniques. A demonstration is the popularity which approaches based on the *divide-and-conquer* strategy have gained in the research community. The divide-and-conquer strategy consists in attacking a complex problem by dividing it into simpler problems whose solutions can be combined to yield a solution to the original problem.

Instances of the divide-and-conquer approach are *neuro-fuzzy inference systems* [4] and *local modeling* techniques [2]. For a comparison of these two approaches see [8]. This paper will focus on local modeling techniques whose main feature is not to return a global fit of the available dataset but perform the prediction of the output for specific test input values, also called *queries*. For that purpose, the database of observed input/output data is always kept in memory and the output prediction is obtained by interpolating the samples in the neighborhood of the query point.

This paper will present some preliminary results obtained by using a global learning technique, a feedforward neural network and a local modeling technique, called *Lazy Learning* [1, 7] for the approximate simulation of an aeronautic device: the heat pipe. A heat pipe (Figure 3) is a device for evacuating heat which is increasingly used in aeronautic systems (eg satellites) as well as in electronics (eg to cool the CPU in laptops and other embedded devices).

The experimental session of this paper will assess and compare the prediction accuracy of the two learning methods trained with a finite number of input/output samples returned by a heat-pipe simulator, jointly developed by CE-NAERO¹ (Centre of Excellence forin Aeronautical Re-

search)² and Euro Heat Pipes³

Also, we will address the problem of dimensionality reduction, that is the problem of detecting on the basis of data which design parameters are the most relevant for predicting the value of the design objectives. This problem can be formulated in a machine learning setting as a problem of feature selection, (for an up-to-date state of the art on feature selection see [13]). In particular we will study the effectiveness of combining local learning techniques with racing methods [15].

Related work on the use of machine learning techniques for accelerating design space exploration can be found in [20], where the MARS regression spline algorithm is used in a system simulation framework for embedded systems and in [19, 18] where feedforward neural networks are used to approximate a Navier-Stokes solver in a turbomachinery application.

Previous applications of lazy learning to design problems in embedded systems can be found in [10, 11]

2 The Heat Pipe simulator

Since a couple of decades, the heat pipe technology has proven its efficiency in the thermal control of highly dissipative equipments such as the electronic component of satellites. A heat pipe is a closed thermodynamic system in which a liquid evaporates in the vicinity of a dissipative source and condenses in contact with a cold region. To insure its passive working in a microgravity environment, the heat pipe is composed of a vapour duct surrounded by a capillary structure. This structure allows for the fluid to return from the cold zone to the heat source. By using a large latent heat fluid, the heat pipe absorbs an important energy quantity during the phase change process, inducing a very high thermal transport capacity for weak variations in temperature.

The grooved heat pipe simulator is developed by CE-NAERO and Euro Heat Pipes. Based on an hydraulic approach, an iterative scheme proceeds to the balance between the global friction losses acting on the heat pipe fluid and the capillary pressure in order to estimate the maximal heat transport capacity.

3 Machine Learning and system design

The design process of complex systems (e.g. a turbine) can be schematically represented by the following elements

- A vector $x \in \mathcal{X} \subset \mathbb{R}^n$ of design parameters (also design configuration), where \mathcal{X} is called the design space. In the case of a turbine, design parameters define the blade geometry.

¹www.cenaero.be

²www.cenaero.be

³www.ehp.be

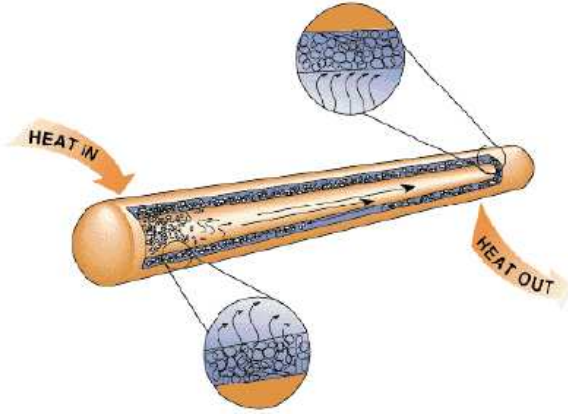


Figure 3: A heat pipe

- A vector $y \in \mathcal{Y} \subset \mathbb{R}^m$ of design objectives which are used to assess the quality of a design. In the turbine example, a design objective is the outlet flow angle.
- An evaluator function $E : \mathcal{X} \rightarrow \mathcal{Y}$ which maps the design space into the objective space. This could be a Navier-Stokes simulator in our example
- A search algorithm which explores the design space in order to find good or optimal configurations x according to the evaluation vector $E(x)$.

Suppose that a number N of input/output pairs (x_i, y_i) , $i = 1, \dots, N$, is obtained by running the simulator, which implements the evaluator function, for N different design configurations. This set can play the role of a training set for a supervised learning technique.

A learning technique aims to find a suitable function $f : \mathcal{X} \rightarrow \mathcal{Y}$ such that the output variable can be accurately represented by a model in the form

$$y = f(x) + \varepsilon \quad (1)$$

where ε is usually thought as the term including modelling error, disturbances and noise. If conventional assumptions of normality and whiteness are made on ε , it follows that a reliable prediction of the output, given the inputs, is returned by

$$\hat{y} = f(x) \quad (2)$$

Traditional approaches address linear formulation of the model $f(\cdot)$

$$\hat{f}(x) = x^T \beta \quad (3)$$

where the vector of parameters is estimated on the basis of the observed dataset by using conventional least-squares techniques. Recent advances in machine learning and data mining have proposed a set of powerful approximators of non-linear relationships.

The idea of modelling a process in a black-box fashion, i.e. from a limited amount of observed data, has been the object of several disciplines from non-linear regression to machine learning and system identification. In

the literature dealing with this problem, two main opposing paradigms have emerged: local memory-based versus global methods. Global modelling builds a single functional model of the dataset. This has traditionally been the approach taken in neural networks [6] and other form of non-linear statistical regression. The available dataset is used by a learning algorithm to produce a model of the mapping and then the dataset is discarded and only the model is kept. Local algorithms defer processing of the dataset until they receive request for information (e.g. prediction or local modelling). The classical nearest neighbour method is the original approach to local modelling. A database of observed input-output data is always kept and the estimate for a new operating point is derived from an interpolation based on a neighbourhood of the query point. The experiments in this paper will compare conventional feedforward neural networks and a local modelling technique, called Lazy Learning (LL), which proved to be successful in many problems of non-linear data analysis and time series prediction [9, 7].

4 Lazy Learning

Given two variables $x \in \mathbb{R}^n$ and $y \in \mathbb{R}$, let us consider the mapping $f: \mathbb{R}^n \rightarrow \mathbb{R}$, known only through a set of N examples $\{(x_i, y_i)\}_{i=1}^N$ obtained as follows:

$$y_i = f(x_i) + \varepsilon_i, \quad (4)$$

where $\forall i$, ε_i is a random variable such that $E[\varepsilon_i] = 0$ and $E[\varepsilon_i \varepsilon_j] = 0, \forall j \neq i$.

Given a query point x_q , the parameter β of a local first-degree polynomial approximating $f(\cdot)$ in a neighborhood of x_q , can be obtained solving the local polynomial regression:

$$\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^N \left\{ (y_i - x_i^T \beta)^2 K \left(\frac{D(x_i, x_q)}{h} \right) \right\}, \quad (5)$$

where, given a metric on the space \mathbb{R}^n , $D(x_i, x_q)$ is the distance from the query point to the i^{th} example, $K(\cdot)$ is a weight function, h is the bandwidth, and where the vectors x_i have been obtained by pre-appending a constant value 1 to each vector x_i in order to consider a constant term in the regression.

Once obtained the local first-degree polynomial approximation, a prediction of $y_q = f(x_q)$, is finally given by:

$$\hat{y}_q = x_q^T \hat{\beta} \quad (6)$$

By exploiting the linearity in the parameters of the local approximator, a leave-one-out cross-validation estimation of the mean squared error $E[(y_q - \hat{y}_q)^2]$ can be obtained without any significant overload. In fact, using the PRESS statistic [16], it is possible to calculate the leave-one-out error $e_j^{cv} = y_j - x_j^T \hat{\beta}^{-j}$, without explicitly identifying the regression parameters $\hat{\beta}^{-j}$ with the j^{th} case set aside.

If a rectangular weight function $K(\cdot)$ is adopted, the optimization of the parameter h can be conveniently reduced to the optimization of the number k of neighbors to which a unitary weight is assigned in the local regression evaluation. In other words, we reduce the problem of bandwidth selection to a search in the space of $h(k) = D(x(k), x_q)$, where $x(k)$ is the k^{th} nearest neighbor of the query point.

The main advantage deriving from the adoption of an indicator weight function, is that, simply by updating the parameter $\hat{\beta}(k)$ of the model identified using the k nearest neighbors, it is straightforward and inexpensive to obtain $\hat{\beta}(k+1)$. The recursive algorithm described in [5, 9] returns for a given query point x_q , a set of predictions $\hat{y}_q(k) = x_q^T \hat{\beta}(k)$, together with a set of associated leave-one-out error vectors $e^{cv}(k)$.

On the basis of this information a final prediction \hat{y}_q of the value of the regression function can be obtained in two different ways: the first is based on the selection of the *best* approximator according to a given criterion, while the second returns a prediction as a *combination* of more local models [17].

This paper illustrates and validates the use of Lazy learning for the task of black-box approximation of physical models for design. The authors deem that this algorithm present a set of specific features which make of it a promising tool for black-box and adaptive system identification in system design. In particular, we consider as relevant:

The reduced number of assumptions Lazy Learning assumes no a priori knowledge on the process underlying the data. For example, it makes no assumption on the existence of a global function describing the data and no assumptions on the properties of the noise. The only available information is represented by a finite set of input/output observations. This feature is particularly relevant in real datasets where problems of missing features, non stationarity and measurement errors make appealing a data-driven and assumption-free approach .

On-line learning capability. The Lazy Learning method can easily deal with on-line learning tasks where the number of training samples increases with time. In this case, the adaptiveness of the method is obtained by simply adding new points to the stored dataset. This make such techniques particularly suitable for design problems where the number of available samples increases all along the exploration of the design space.

Effective feature selection: The usefulness of a local modeling approach for reducing the cost of feature selection was first presented in [15]. The idea consists in assessing a large number of feature subsets by performing cross-validation only on a reduced test set. On the basis of well-known statistical results, it is possible to show that families of good feature subsets can

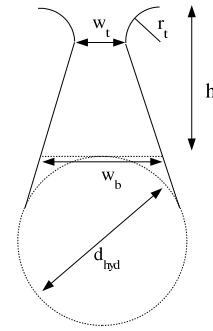


Figure 4: Diagram representing a groove of a heat pipe.

be rapidly found by quickly discarding the bad subsets and concentrating the computational effort on the better ones. This model selection technique was called the *Hoeffding race* by Maron and Moore [15], with reference to Hoeffding's formula which puts a bound on the accuracy of a sampled mean of N observations as an estimator of the expected value.

5 Some experimental results

The prediction accuracy of two learning algorithms, a feed-forward neural network and Lazy Learning are assessed on the basis of input/output samples generated by the simulator described in Section 2.

The feedforward neural network has two layers, the first (hidden) based on a sigmoidal transfer function and the second (output) based on a linear transfer function. The number of nodes in the hidden layer is determined using an empirical relation which is a function of the number of training data, the number of inputs and the number of outputs. For more details on the technique, we refer the reader to [19, 18]

Note that, in the case of LL, we train a different input/output model for each output: this means that the multi-input multi-output problem is treated as a series of multi-input single-output problems.

Two sessions of experiments are carried out, the first with no feature selection, the second with a preliminary selection of the relevant design parameters.

The experiments use two datasets, described in the following section, which differ for the number of design parameters and the number of samples.

5.1 The two datasets

The first dataset, henceafter $D1$, is composed of $N = 1260$ samples. It has $n = 3$ inputs and $m = 2$ outputs. The input design parameters are (see Figures 4 and 5):

- the internal diameter of the heat pipe.
- the diameter of the groove (d_{hyd}).

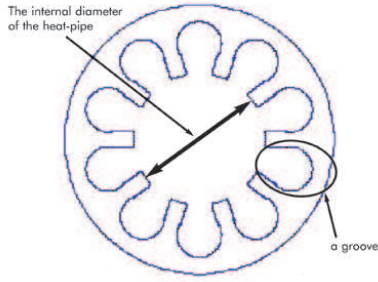


Figure 5: A section of a heat-pipe

- the inclination angle of the heat pipe.

The output design criteria are:

- The power (in Watt) released by the heat pipe.
- The external diameter of the heat pipe.

The second dataset, henceafter $D2$, is composed of $N = 820$ samples. It has $n = 6$ input variables and $m = 2$ output variables. The input design parameters are (see Figures 4 and 5):

- The internal diameter of the heat pipe.
- The number of groove in the heat-pipe.
- The diameter of the groove (d_{hyd}).
- The width of the bottom of the grooves (w_b).
- The width of the top of the grooves (w_t).
- The depth of the grooves (h).

The two output design criteria are the same as in $D1$.

5.2 The validation procedure

We adopt the following procedure in order to assess the prediction accuracy.

The total amount of the samples is randomly divided into two halves; the first half is used for the training and the second half is used for the validation. The accuracy is returned by the average of the squared prediction errors of the learned models in the validation set.

The results are reported in Table 1 and in Table 2 for $D1$ and $D2$, respectively.

5.3 Feature selection

We applied the racing feature selection algorithm [15] to reduce the dimensionality of the datasets $D1$ and $D2$.

Learner	Output 1	Output 2
LL	2.9e-04	2.2e-05
NN	5.0e-03	1.2e-04

Table 1: Before feature selection: Mean square prediction error for the two outputs of $D1$

Learner	Output 1	Output 2
LL	1.4e-02	1.0e-05
NN	2.2e-02	4.8e-05

Table 2: Before feature selection: Mean square prediction error for the two outputs of $D2$

Dataset 1: As far the first output variable of $D1$ is concerned, no reduced input dimensionality was found.

As far the second output variable of $D1$ is concerned, the feature selection algorithm returned an input set composed of $n = 2$ variables, where the third design parameter is not considered.

We report in Table 3 the mean square prediction errors of the two learning methods for the second output.

Dataset 2: the feature selection algorithm reduced the dimensionality to $n = 5$ both for the first output variable (last design parameter excluded) and the second output variable (second design parameter excluded). The mean square prediction error for the first and the second output are reported in Table 4 and 5, respectively.

6 Conclusion and future work

The need for repetitive evaluations of the quality of a system during the design process asks for rapid and precise predictors to be employed in the exploration of the design space

The preliminary results presented in this paper show that machine learning techniques can return very precise estimation of the quality objectives on the basis of a limited amount of observations. In particular the Lazy Learning technique appears competitive with more conventional machine learning techniques, like feedforward neural networks.

This appears still more relevant if we consider that Lazy Learning features some nice properties for the prob-

Learner	variable 2
LL	7.0e-07
NN	2.6e-05

Table 3: After feature selection: Mean square prediction error for the second output of $D1$

Learner	variable 1
LL	5.7 e-03
NN	1.3 e-02

Table 4: After feature selection: Mean square prediction error for the first output of $D2$

Learner	variable 2
LL	8.8e-06
NN	1.5e-04

Table 5: After feature selection: Mean square prediction error for the second output of $D2$

lem of multidimensional system design, like rapid tuning, fast adaptation and effective feature selection.

Future works will focus on (i) extending the experiments to a larger number of design parameters and quality objectives and (ii) integrating the black-box approximators in the optimization process.

References

- [1] D. W. Aha. Editorial of special issue on lazy learning. *Artificial Intelligence Review*, 11(1–5):1–6, 1997.
- [2] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1–5):11–73, 1997.
- [3] F. Balarin, M. Chiodo, P. Giusto, H. Hsieh, A. Jurcska, L. Lavagno, C. Passerone, A. Sangiovanni-Vincentelli, E. Sentovich, K. Suzuki, and B. Tabbara. *Hardware-Software Co-Design of Embedded Systems: The Polis Approach*. Kluwer Academic Press, 1997.
- [4] H. Bersini and G. Bontempi. Now comes the time to defuzzify the neuro-fuzzy models. *Fuzzy Sets and Systems*, 90(2):161–170, 1997.
- [5] M. Birattari, G. Bontempi, and H. Bersini. Lazy learning meets the recursive least-squares algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *NIPS 11*, pages 375–381, Cambridge, 1999. MIT Press.
- [6] C. M. Bishop. *Neural Networks for Statistical Pattern Recognition*. Oxford University Press, Oxford, UK, 1994.
- [7] G. Bontempi. *Local Learning Techniques for Modeling, Prediction and Control*. PhD thesis, IRIDIA-Université Libre de Bruxelles, 1999.
- [8] G. Bontempi, H. Bersini, and M. Birattari. The local paradigm for modeling and control: From neuro-fuzzy to lazy learning. *Fuzzy Sets and Systems*, 121(1):59–72, 2001.
- [9] G. Bontempi, M. Birattari, and H. Bersini. A model selection approach for local learning. *Artificial Intelligence Communications*, 121(1), 2000.
- [10] G. Bontempi and W. Kruijtzter. The use of intelligent data analysis techniques for system-level design: a software estimation example. *Soft Computing*, 8(7):477–490, 2004.
- [11] G. Bontempi and G. Lafruit. Enabling multimedia qos control with black-box modeling. In D. Bustard, W. Liu, and R. Sterritt, editors, *Soft-Ware 2002: Computing in an Imperfect World, Lecture Notes in Computer Science*, pages 46–59, 2002.
- [12] D.D. Gajski, F. Vahid, S. Narayan, and J. Gong. *Specification and design of embedded systems*. Prentice Hall, 1994.
- [13] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [14] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, 2001.
- [15] O. Maron and A. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11(1–5):193–225, 1997.
- [16] R. H. Myers. *Classical and Modern Regression with Applications*. PWS-KENT Publishing Company, Boston, MA, second edition, 1994.
- [17] M. P. Perrone and L. N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In R. J. Mammone, editor, *Artificial Neural Networks for Speech and Vision*, pages 126–142. Chapman and Hall, 1993.
- [18] S. Pierret, A. Demeulenaere, B. Gouverneur, C. Hirsch, and R. Van den Braembussche. Designing turbomachinery blades with the function approximation concept and the navier-stokes equations. In *Proceeding of the 8th AIAA/NASA/USAF/ISSMO Symposium on MDO*, 2000.
- [19] S. Pierret and R.A. Van den Braembussche. Turbomachinery blade design using a navier-stokes solver and artificial neural network. *ASME Trans., Journal of Turbomachinery*, 121(2):326–332, 1999.
- [20] P. van den Hamer, W.P.M. van der Linden, P. Bingley, and N.W. Schellingerhout. A system simulation framework. In *Proceedings of the 37th Conference on Design Automation*, pages 699–704, 2000.