HAL
open science

Towards the Optimal Design of Numerical Experiments

Stéphane Gazut, Jean-Marc Martinez, Gerard Dreyfus, Yacine Oussar

To cite this version:

Stéphane Gazut, Jean-Marc Martinez, Gerard Dreyfus, Yacine Oussar. Towards the Optimal Design of Numerical Experiments. IEEE Transactions on Neural Networks, 2008, 19 (5), pp.874-882. 10.1109/TNN.2007.915111 . cea-02511993

HAL Id: cea-02511993

https://cea.hal.science/cea-02511993v1

Submitted on 19 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards the Optimal Design of Numerical Experiments

Stéphane Gazut, Jean-Marc Martinez, Gérard Dreyfus, *Senior Member, IEEE*, and Yacine Oussar

***Abstract*—This paper addresses the problem of the optimal design of numerical experiments for the construction of nonlinear surrogate models. We describe a new method, called learner disagreement from experiment resampling (LDR), which borrows ideas from active learning and from resampling methods: the analysis of the divergence of the predictions provided by a population of models, constructed by resampling, allows an iterative determination of the point of input space, where a numerical experiment should be performed in order to improve the accuracy of the predictor. The LDR method is illustrated on neural network models with bootstrap resampling, and on orthogonal polynomials with leave-one-out resampling. Other methods of experimental design such as random selection and D-optimal selection are investigated on the same benchmark problems.**

***Index Terms*—Active learning, bagging, bootstrap, D-optimality, neural networks.**

I. Introduction

ALTHOUGH numerical simulation tends to be ubiquitous in today's engineering, computation time often limits its use, despite the ever increasing power of computers. A common technique for circumventing that limitation is the design of *surrogate models*, i.e., analytical functions that approximate the input–output mapping performed by the simulation model. Still, the estimation of the parameters of the surrogate models requires the availability of results obtained by the simulation model that it is intended to approximate; therefore, whenever numerical experiments are costly, it is important to select them as efficiently as possible in order to minimize their number. In statistics, the selection of experiments is known as optimal experimental design (OED), see, for instance, [1] and [2], while it is known as *active learning* in the machine learning literature.

Optimal experimental design has been widely developed for models that are linear in their parameters, such as polynomials. The observations of a given quantity are assumed to be realizations of a random variable that is the sum of a deterministic function (the regression function, assumed to be linear in its parameters) and of a random variable with zero mean. By contrast, in this work, the existence of the latter random variable is not assumed: in other words, repeated experiments will provide identical results. Such is the case when the data to be modeled is generated by a deterministic computer simulation. Moreover, we relax the assumption that the model is linear in its parameters. Therefore, we describe a generic alternative approach to experimental design, based on resampling techniques.

Sections II and III are intended to put optimal experimental design and active learning into the perspective of the present work. The two subsequent sections describe two variants of the method that we advocate in the context of numerical experiments. Finally, those approaches are compared on classical benchmark problems.

Manuscript received September 12, 2006; revised May 22, 2007; accepted August 28, 2007. This work was supported in part by the CEA Grant. This work was performed in part within the framework of the NeuroPex project on experimental planning for neural network models. Among the partners were the Netral company, the CEA (DAM/DP2I) and the CEA (DEN/DM2S).

S. Gazut and J.-M. Martinez are with the DM2S/SFME Centre d'Etudes de Saclay, 91191 Gif sur Yvette, France (e-mail: stephane.gazut@cea.fr; jean-marc.martinez@cea.fr).

G. Dreyfus and Y. Oussar are with the Ecole Suprieure de Physique et de Chimie Industrielles de la ville de Paris (ESPCI-Paristech), Laboratoire d'Electronique (UMR CNRS 7084), 75231 Paris Cedex 05, France (e-mail: gerard.dreyfus@espci.fr; yacine.oussar@espci.fr).

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

Digital Object Identifier 10.1109/TNN.2007.915111

II. Background: D-Optimality in Experimental Design

The mainstream development of optimal experimental design dealt with linear-in-their-parameters models, starting with the work of Kiefer [3], Kiefer and Wolfowitz [4], Fedorov [1], or Wynn [5]. Vila [6], MacKay [7], Cohn [8], and more recently Issanchou and Gauchi [9] and Witczak [10], described applications of optimal experimental design techniques to the training of neural networks.

A. Training Models From Data

We consider an unknown function $y(\mathbf{x})$ in a domain $U \subset R^d$. We denote by $\mathcal{L} = \{(y_k, \mathbf{x}_k), k = 1, \ldots, n\}$ a finite set of observations, where $\mathbf{x}$ is drawn from a probability distribution $p(\mathbf{x})$, and where $y_k = y(\mathbf{x}_k)$.

We consider a family of parameterized functions $f(\mathbf{x}, \boldsymbol{\theta})$, within which we seek the "best" approximation of the unknown function $y(\mathbf{x})$, given the available data $\mathcal{L}$. To that effect, the loss function $l(y(\mathbf{x}), f(\mathbf{x}, \boldsymbol{\theta})) = ||y(\mathbf{x}) - f(\mathbf{x}, \boldsymbol{\theta})||^2$ is defined, which expresses the discrepancy between function $y(\mathbf{x})$ and its approximation $f(\mathbf{x}, \boldsymbol{\theta})$. The parameters of the model are estimated by minimizing a cost function which is the sum, over all examples of a data set called training set, of $l(y_k, f(\mathbf{x}_k, \boldsymbol{\theta}))$; we denote by $\boldsymbol{\theta}_{\mathcal{L}}$ the vector of parameters for which the cost function is minimum

$$\boldsymbol{\theta}_{\mathcal{L}} = \arg\min_{\boldsymbol{\theta}} \sum_{(y_k, \mathbf{x}_k) \in \mathcal{L}} l\left(y_k, f(\mathbf{x}_k, \boldsymbol{\theta})\right). \tag{1}$$

B. Linear Framework

In the linear framework, D-optimal experimental design consists in organizing the experiments in order to minimize the variance of the estimated parameters, by maximizing the Fisher matrix determinant $(\det(\mathbf{X}'\mathbf{X}))$, where $\mathbf{X}$ is the experimental ma-

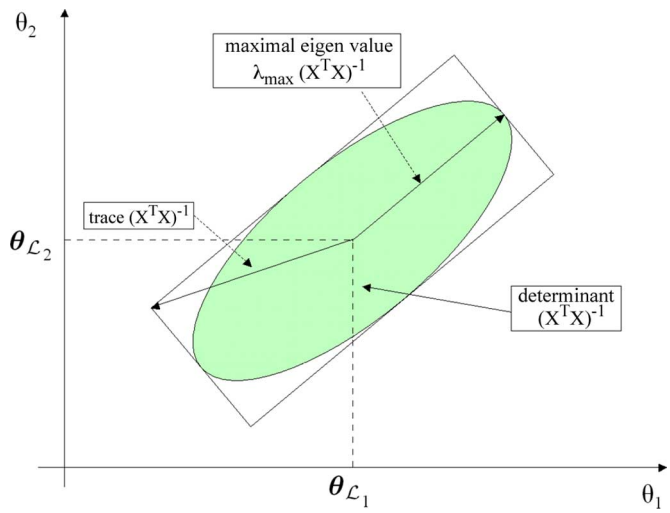

Fig. 1. Confidence area for two parameters.

trix, whose element x_{ij} is the value of variable j observed in experiment i. $\mathbf{X}$ is a (N, p) matrix, where N is the number of observations and p is the number of variables. We denote by $\mathbf{X}^+ = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ the pseudoinverse of $\mathbf{X}$.

Let $\boldsymbol{\theta}_{\mathcal{L}} = \mathbf{X}^+\mathbf{y}$ be the least squares estimator of the unknown function parameters for the data set $\mathcal{L}$. The model $f(\mathbf{x}, \boldsymbol{\theta})$ is postulated to be linear in its parameters. In the probabilistic framework, under the hypothesis of uncorrelated centered residuals with variance σ^2, the variance–covariance matrix of the parameters is

$$V(\boldsymbol{\theta}_{\mathcal{L}}) = V(\mathbf{X}^+ y) = \mathbf{X}^+ V(y) \mathbf{X}^{+'} = \sigma^2 (\mathbf{X}'\mathbf{X})^{-1}. \quad (2)$$

The Fisher matrix $\mathbf{X}'\mathbf{X}$ depends on the distribution of the experimental values of the variables. It is, therefore, natural to seek a distribution of points that reduces the variance of the parameters to the largest extent. Under the additional hypothesis of Gaussian residual error, the confidence area of the estimated parameters is a hyperellipsoid centered in $\theta_{\mathcal{L}}$ and defined, for a confidence level α, by [11]

$$(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathcal{L}})' \mathbf{X}'\mathbf{X} (\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathcal{L}}) \leq \sigma^2 \chi^2_{\alpha}(p) \quad (3)$$

where $\chi^2_{\alpha}(p)$ is the chi-square α quantile with p degrees of freedom.

Many optimality criteria may be considered. We will describe the main optimal experimental design techniques that make use of the spectral properties of the dispersion matrix $(\mathbf{X}'\mathbf{X})^{-1}$.

The confidence area, or, more generally, the volume of the confidence ellipsoid as shown on Fig. 1, can be acted upon by decreasing the following :

- the length of the main axis of the ellipsoid, i.e., the largest eigenvalue of $(\mathbf{X}'\mathbf{X})^{-1}$ (E-optimality criterion);
- the sum of the lengths of the axes of the ellipsoid, i.e., the trace of $(\mathbf{X}'\mathbf{X})^{-1}$ (A-optimality criterion);
- the volume of the ellipsoid, i.e., the determinant of $(\mathbf{X}'\mathbf{X})^{-1}$ (D-optimality criterion).

Various algorithms [1], [12] are available for finding exact solutions satisfying the previous optimality criteria, for postulated models that are linear in their parameters. In that context, the solution depends only on matrix $\mathbf{X}$; therefore, it does not depend on the model, insofar as it is postulated to be linear in its parameters. In other words, experimental planning can be performed *prior to modeling* in that context. That is no longer true for models that are nonlinear in their parameters, as will be shown in Section II-C, which describes a D-optimal experimental design methodology for such models.

C. Nonlinear Framework

In the nonlinear case, useful results are frequently obtained by performing a first-order Taylor expansion of the model, in parameter space, in the neighborhood of the parameter vector $\boldsymbol{\theta}_{\mathcal{L}}$ for which the least squares cost function is minimum

$$f(\mathbf{x}, \boldsymbol{\theta}) \simeq f(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{L}}) + \mathbf{Z}(\boldsymbol{\theta}_{\mathcal{L}})(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathcal{L}}) \quad (4)$$

where $\mathbf{Z}$ is the Jacobian matrix of the model

$$[\mathbf{Z}(\mathbf{x}_i, \boldsymbol{\theta})]_{ij} = \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \theta_j} \right)_{\mathbf{x}=\mathbf{x_i}} \quad (5)$$

$\mathbf{x_i}$ is the vector of variables for the ith observation, and θ_j is the jth parameter of model $f(.)$ with parameter vector $\boldsymbol{\theta}$. $\mathbf{Z}$ is an (N, p) matrix, where N is the number of observations and p is the number of parameters of the model. This provides a locally linear approximation of the model, whose variables are the partial derivatives of the model with respect to its parameters. Therefore, the Jacobian matrix $\mathbf{Z}$ of the model plays the same role as the experimental matrix $\mathbf{X}$ does for linear-in-their-parameters models. Actually, if the model is linear in its parameters, matrices $\mathbf{X}$ and $\mathbf{Z}$ are identical.

By contrast to matrix $\mathbf{X}$, matrix $\mathbf{Z}$ depends on the parameters of the model. That technique allowed, for instance, the estimation of confidence intervals [13], of the tangent-plane leverages and of the generalization error [14] of nonlinear models. In the same spirit, Issanchou and Gauchi [9] proposed, in the homoscedastic case, an optimal experimental planning technique based on the minimization of the approximate volume of the confidence ellipsoid, proportional to $(\mathbf{Z}'\mathbf{Z})$.

Since the Jacobian matrix depends on the parameters of the model, experimental planning cannot be performed prior to modeling. Therefore, a two-step procedure is necessary. Before the construction of the D-optimal design, an initial set of experiments must be available, e.g., by Latin hypercube sampling (LHS)[1]; from that initial data set, a first estimate of the parameters of the nonlinear model is obtained, allowing the computation of the Jacobian matrix.

The algorithms that are available for the construction of D-optimal experimental design can be applied simply, replacing matrix $\mathbf{X}$ by matrix $\mathbf{Z}$, in order to obtain D-optimal experiments that can be used in addition to the initial ones. Local D-optimality is often denoted as $D(\boldsymbol{\theta}_{\mathcal{L}})$-optimality, where $\boldsymbol{\theta}_{\mathcal{L}}$ represents the parameter vector for which the least squares cost function is minimum.

[1]The LHS method was developed to generate a distribution of experiments from a multidimensional distribution [15]. A square grid is a Latin square if there is only one sample in each row and each column. A Latin hypercube is the generalization of a Latin square in an arbitrary number of dimensions. The LHS sampling provides an efficient sample placement in the input space of variables.

D. Algorithmic Construction of D-Optimal Experimental Designs

There are many algorithms for the construction of optimal experimental designs; see, for instance, [5], [12], or [16]. We describe here Fedorov's algorithm [1], which is probably the most popular and easiest to code. The purpose is to select N experiments, in a set of N_c candidates, which maximize the determinant of the Fisher matrix $(\mathbf{X}'\mathbf{X})$ for linear-in-their-parameters models, or $(\mathbf{Z}'\mathbf{Z})$ for nonlinear-in-their-parameters models.

Step 1) Choose N experiments randomly in a set of N_c candidate experiments, which are typically the nodes of a "fine" grid.

Step 2) Perform all possible exchanges of an experiment i of the initial design with an experiment j of the candidate experiments; there are $N(N_c - N)$ different exchanges (repeated experiments are not allowed in the context of numerical experiments since repeated numerical experiments yield identical results); compute the $N(N_c - N)$ determinants of the corresponding Fisher matrices.

Step 3) Perform the exchange that increases the determinant of the Fisher matrix by the largest amount. Iterate to Step 2) if the termination criterion is not satisfied.

To compute the determinant at the current iteration, the following theorem can be used [11].

After the exchange of i with j at iteration t, the new information matrix is

$$(\mathbf{X}'\mathbf{X})_{[t+1]} = (\mathbf{X}'\mathbf{X})_{[t]} - \mathbf{x}_i\mathbf{x}_i' + \mathbf{x}_j\mathbf{x}_j'. \tag{6}$$

As a consequence, the determinant is

$$\det\left((\mathbf{X}'\mathbf{X})_{[t+1]}\right) = \det\left((\mathbf{X}'\mathbf{X})_{[t]}\right) \times [1 + \Delta(i,j)] \tag{7}$$

with

$$\Delta(i,j) = h_{jj} - \left[h_{ii}h_{jj} - hij^2\right] - h_{ii} \tag{8}$$

where h_{ij} is the element ij of the hat matrix $H = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$

$$h_{ij} = \mathbf{x}_i'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_j. \tag{9}$$

Various termination criteria may be considered. For instance, the algorithm may be stopped when the increase of the determinant is smaller than a chosen value. As usual with greedy algorithms, local optima exist in general, so that the solution thus obtained may be suboptimal.

III. Active Learning Background

In contrast to classical learning (passive learning), the active learner selects the most useful experiments to be added to the initial data set. The learner chooses the best instances from a given set of unlabeled examples (*pool-based sample selection* [17], [18]).

The active learning strategy can be summarized by the following three steps:

- train the learner using the current training set;
- choose a point $\mathbf{x}$ in the pool of candidate experiments;
- measure or compute the corresponding quantity of interest y and add the point $(\mathbf{x}, \mathbf{y})$ to the training set.

This procedure is an incremental strategy, which adds new training points iteratively.

The main question in active learning is how to choose the point $\mathbf{x}$ in the second step. Various strategies may be considered, such as the following:

- adding experiments where data is missing;
- adding experiments where confidence in model predictions is low [19];
- adding experiments in order to minimize the generalization error of the model; see, for instance, [20] for support vector machine, or [21], where the *expected integrated squared difference* (which is an estimation of the generalization error in a Bayesian framework) is minimized.

IV. Design of Experiments by LDR Bagging

In this section, we describe a new method called learner disagreement from experiment resampling (LDR). As explained in Section II-A, we denote by $\boldsymbol{\theta}_{\mathcal{L}}$ the vector of the parameters for which the cost function is minimum

$$\boldsymbol{\theta}_{\mathcal{L}} = \arg\min_{\boldsymbol{\theta}} \sum_{(y_k,\mathbf{x}_k)\in\mathcal{L}} l\left(y_k, f(\mathbf{x}_k, \boldsymbol{\theta})\right). \tag{10}$$

Clearly, different optimal parameter vectors will be derived from different training sets; that variability can be investigated by resampling methods such as bagging (bootstrap aggregation) [22]; we first describe that method, which is central to our experimental planning technique.

A. Bagging

Given a training set $\mathcal{L}$, the aggregated predictor is defined by

$$f_A(\mathbf{x}) = E_{\mathcal{L}}\left[f(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{L}})\right] \tag{11}$$

where $E_{\mathcal{L}}$ is the expectation value of the predictions of the model, for variable vector $\mathbf{x}$, for all possible training sets $\mathcal{L}$ of identical size; the expectation value is estimated by the average, hence the subscript A.

The prediction provided by the aggregated predictor is more accurate than the average of the predictions provided by the individual predictors of the same family on the same data set

$$\|f_A(\mathbf{x}) - y(\mathbf{x})\|^2 = \|E_{\mathcal{L}}\left[f(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{L}})\right] - y(\mathbf{x})\|^2 \tag{12}$$

$$\leq E_{\mathcal{L}}\|f(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{L}}) - y(\mathbf{x})\|^2. \tag{13}$$

An estimation of f_A can conveniently be obtained by the bootstrap [23], a statistical resampling method: examples are drawn randomly with replacement from the original data set $\mathcal{L}_n$ of size n, thereby generating an ensemble of B data sets of identical size n. Denoting by $\mathcal{L}_n^{*b}$ the bootstrap sample (or replicate) b, the estimated expectation by bootstrap (hence the subscript $\mathcal{B}$) with B replicates is

$$f_{n,\mathcal{B}}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} f\left(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{L}_n^{*b}}\right). \tag{14}$$

TABLE I
LDR ALGORITHM

LDR-Algorithm

Given:
T_n - set of training examples
P - set of candidate experiments (generally a grid of variable space)
y - unknown function
k - number of selected candidate experiments, appended to the training set after selection
n - size of each sample ($n \neq 0$)

Repeat k times:
1) Generate B bootstrap samples $\mathcal{L}_n^{*b}$
2) Generate the bootstrap aggregated predictor $f_{n,\mathcal{B}}(\mathbf{x})$
3) $\forall x_j \in P$ compute the estimated variance of the predictions $\sigma^2_{\mathcal{B}}[f(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{L}_n})]$
4) Select the point of maximal prediction variance in P, noted $\mathbf{x}_{\mathrm{new}}$
5) Remove $\mathbf{x}_{\mathrm{new}}$ from P and add $(\mathbf{x}_{\mathrm{new}}, y(\mathbf{x}_{\mathrm{new}}))$ to T_n; $(n \leftarrow n+1)$.

B. Active Learning by LDR Bagging

Similarly to estimating the expectation value of the predictions, their variance can be estimated by bootstrapping of the original data set

$$\sigma^2_{\mathcal{B}}\left[f\left(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{L}_n}\right)\right] = \frac{1}{B-1}\sum_{b=1}^{B}\left(f\left(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{L}_n^{*b}}\right) - f_{n,\mathcal{B}}(\mathbf{x})\right)^2. \tag{15}$$

The approach to active learning, or experimental planning, that we advocate here consists in adding new experiments in the regions of variable space where the bootstrap estimate of the variance of the predictions is the largest, i.e., where the predictors constructed from data sets obtained by bootstrap resampling disagree most. That can be viewed as a paradigm of a teacher–classroom interaction, where each student learns from a part of the data, the teacher asks questions, the classroom provides answers, and new questions are asked in the area where the greatest disagreement between all possible answers arises.

Therefore, our method can be summarized as follows.

- Find the point of maximal prediction variance

$$\mathbf{x}_{\mathrm{new}} = \arg\max_{x \in U} \sigma^2_B\left[f\left(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{L}_n}\right)\right]. \tag{16}$$

- Perform a numerical experiment, i.e., compute $f(\mathbf{x}_{\mathrm{new}})$, and include the experiment [$\mathbf{x}_{\mathrm{new}}$ and $f(\mathbf{x}_{\mathrm{new}})$] in the initial sample $\mathcal{L}_n$ in order to obtain the new sample $\mathcal{L}_{n+1}$

$$\mathcal{L}_{n+1} = \mathcal{L}_n \cup \left\{(\mathbf{x}_{\mathrm{new}}, y(\mathbf{x}_{\mathrm{new}}))\right\}. \tag{17}$$

This active strategy is terminated when the decrease of the prediction variance is not significant, or when the predefined maximum number of additional experiments is reached (see the LDR algorithm in Table I).

The generation of the bootstrap aggregated predictor (step 2) involves the training of the model by an appropriate procedure. For linear-in-their-parameters models, the procedure may be ordinary least squares; for non-linear-in-their-parameters models, such as neural networks, training is performed by minimizing the chosen cost function. In the latter case, the main source of variability should be the resampling process, rather than the existence of local minima of the cost function; to that end, for each bootstrap sample, several models are trained, and a single model is selected, as explained in Section VI-A.

C. Simple Didactic Example

We illustrate the previous procedure by the simple example of $\sin(x)/x$. The initial training set features ten experiments, not uniformly distributed in variable space. The bootstrap estimates of the prediction variance over all candidate points of a grid are computed, and the point with maximum prediction variance estimate is included in the initial training set, as shown in Fig. 2.

The estimated prediction variance decreases significantly at each step in the vicinity of the new points, and decreases globally in the domain. The active strategy is terminated when the decrease of the predictive variance is not significant (we would have stopped the heuristic at the fifth step). Fig. 3 compares the generalization error of models that learned on data selected by LDR and on data sets generated randomly. The generalization error is estimated by an integration Monte Carlo method which provides an estimate of

$$\int_U \left(y(\mathbf{x}) - f(\mathbf{x}, \boldsymbol{\theta})\right)^2 p(\mathbf{x}) d\mathbf{x}. \tag{18}$$

It shows that the generalization error in the LDR case decreases significantly with the number of new experiments.

Note that we compare the generalization performance of LDR-designed models with the generalization performance of models obtained by training from a single random data set. Section VI reports comparisons between LDR designed models, D-optimality designed models, and models trained from 500 different random data sets.

V. Active Learning by LDR Leave-One-Out

The prediction variance may also be estimated by leave-one-out, especially when the models are linear in their parameters: in that case, the variance of the parameters can be computed explicitly. The application of that technique to a benchmark [24] is described in Section VI.

The models were sought within the family of linear combinations of functions Ψ_k resulting from the tensorization of orthogonal Legendre polynomials. The family of orthogonal Legendre polynomials provides a well-conditioned information matrix

$$y(\mathbf{x}) = \sum_{k=0}^{P} \theta_k \Psi_k(\mathbf{x}) = \Psi'(\mathbf{x})\boldsymbol{\theta}. \tag{19}$$

The parameters θ_k are computed by ordinary least squares, as described in Section II-A or by singular value decomposition (SVD). We denote by $\boldsymbol{\theta}_{(i)}$ the parameter vector obtained by removing example i from the training set, by h_{ii}, the leverage of observation i [the ith diagonal element of the hat matrix $H = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$], and by ϵ_i, the residual of example i when it is present in the training set. The parameter vector $\boldsymbol{\theta}_{(i)}$ is obtained explicitly by [11]

$$\boldsymbol{\theta} - \boldsymbol{\theta}_{(i)} = (\mathbf{X}'\mathbf{X})^{-1}\frac{\mathbf{x}_i \epsilon_i}{1 - h_{ii}}. \tag{20}$$

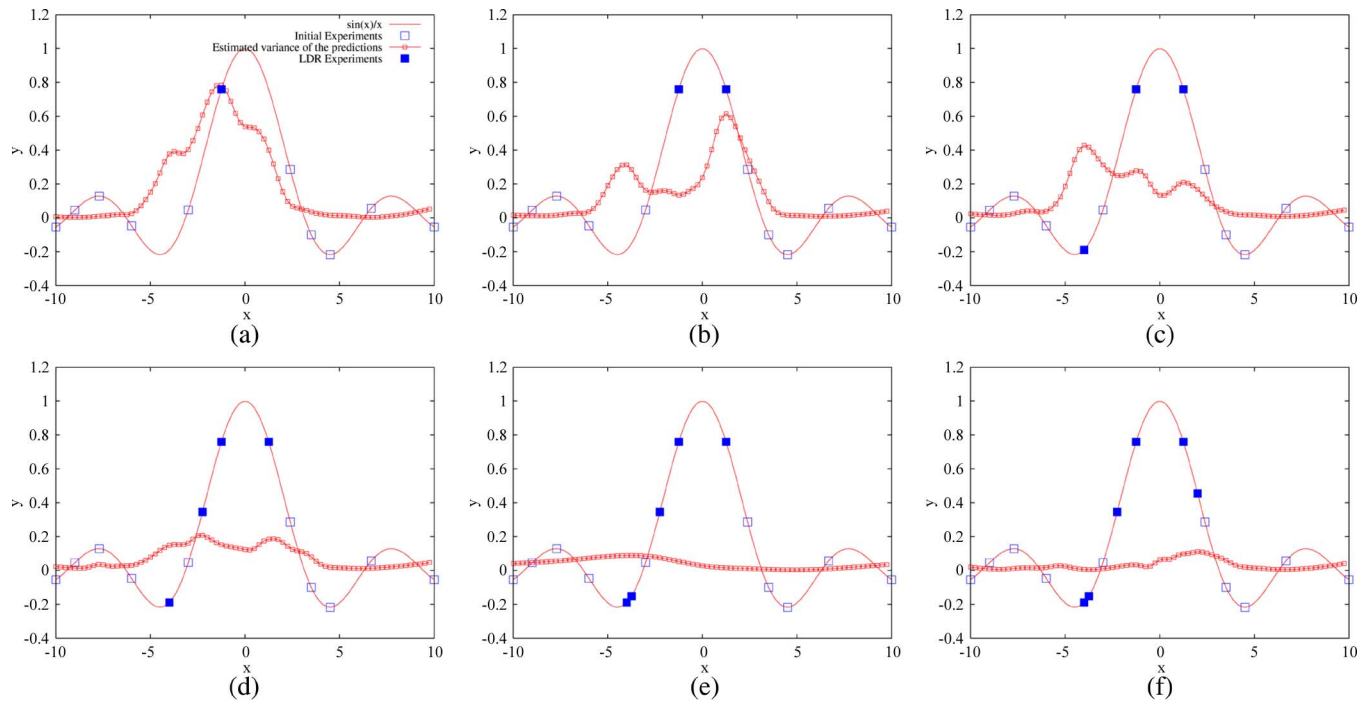

Fig. 2. Active learning with LDR bagging. (a) First iteration. (b) Second iteration. (c) Third iteration. (d) Fourth iteration. (e) Fifth iteration. (f) Sixth iteration.

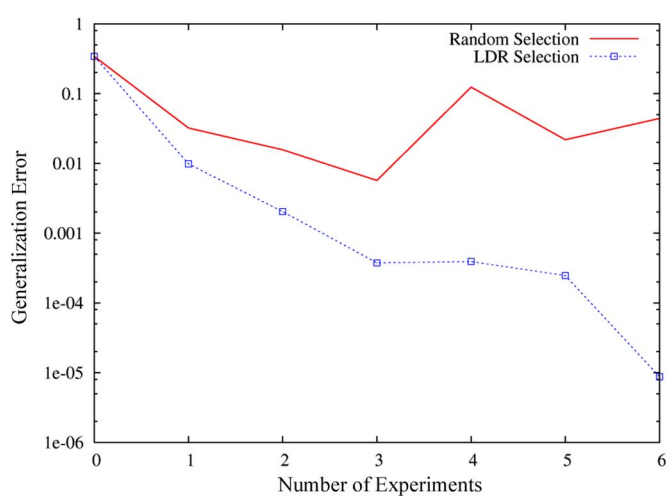

Fig. 3. Comparison of generalization error between models that learned on LDR-generated data sets and on random data sets.

We denote by α_i the quantity $\epsilon_i/(1-h_{ii})$ by Λ, the matrix of elements $\Lambda_{ij} = \alpha_i^2\delta_{ij} - (1/n)\alpha_i\alpha_j$, and by $\mathbf{X}^+ = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$, the pseudoinverse of $\mathbf{X}$. The estimated prediction variance $\sigma_y^2(\mathbf{x})$ is given by

$$\sigma_\theta^2 = \frac{1}{n}\mathbf{X}^+\Lambda\mathbf{X}^{+'} \Rightarrow \sigma_y^2(\mathbf{x}) = \frac{1}{n}\Psi'(\mathbf{x})\mathbf{X}^+\Lambda\mathbf{X}^{+'}\Psi(\mathbf{x}). \quad (21)$$

Since the prediction variance can be computed exactly (within numerical roundoff errors) from the pseudoinverse $\mathbf{X}^+$, the new point $\mathbf{x}_{\texttt{new}}$ can be obtained without resorting to resampling

$$\mathbf{x}_{\texttt{new}} = \arg\max_{\mathbf{x}\in U}\left\|\Psi'(\mathbf{x})\mathbf{X}^+\Lambda^{1/2}\right\|. \quad (22)$$

The new point is chosen in a set of candidate experiments.

VI. Results

In this section, the efficiencies of D-optimality, LDR active learning, and random sampling of variable space are compared on three different problems.

A. Homma–Saltelli Benchmark [24]

The method was validated on the Homma–Saltelli benchmark. The data-generating function is

$$y(\mathbf{x}) = \sin(x_1) + 7\sin^2(x_2) + 0.1x_3^4\sin(x_1) \quad \text{with} \quad x_i \in [-\pi;\pi] \quad \forall i = 1,\ldots,3. \quad (23)$$

The 100 experiments were obtained by LHS, thereby generating the initial data set.

In the following, the models are feedforward neural networks [multilayer perceptrons (MLPs)] with a single layer of hidden neurons. With the initial data set of 100 examples, we trained several MLPs with different numbers of hidden neurons. The generalization error of each MLP was estimated by the Monte Carlo integration method mentioned in Section IV-C. Models with 12 hidden neurons gave a good bias-variance tradeoff. The purpose of experimental planning was to supplement the initial training set of 100 examples with 60 additional examples. The generalization error was also estimated by the Monte Carlo integration method.

1) Results for LDR-Bagging Method:

- **Comparison between D-optimality, LDR active learning, and random sampling of variable space:** We compared D-optimality, LDR active learning, and random sampling of variable space in order to estimate the accuracy of the first two planning techniques with respect to the accuracy of a random strategy. Since the random strategy is not representative with only one data set, 500

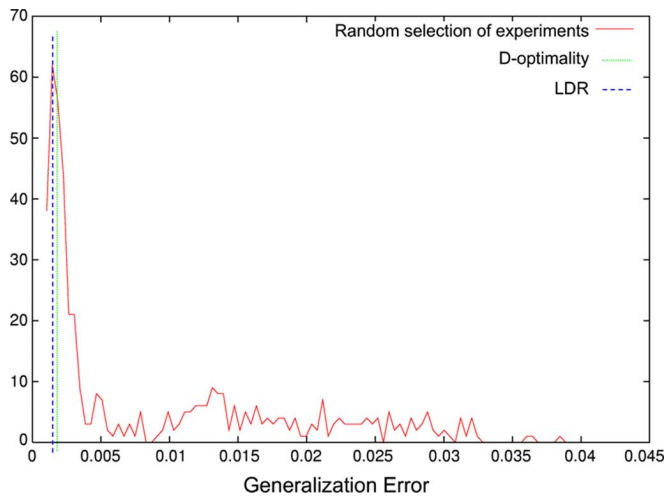

Fig. 4. Comparison between D-optimal design, LDR design, and random strategy.

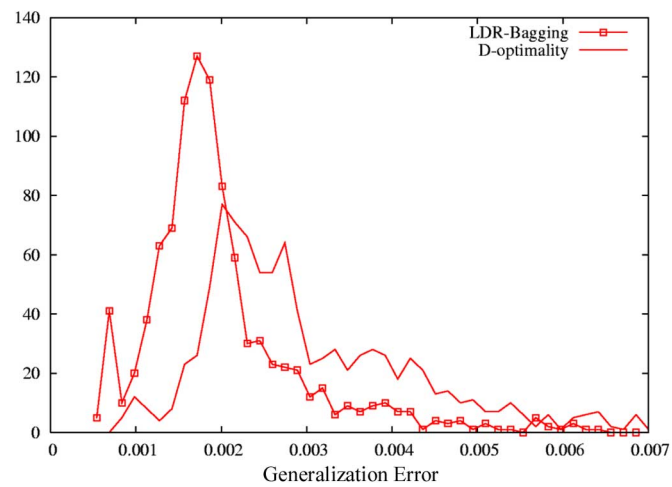

Fig. 5. Generalization error distribution of 1000 MLPs trained on D-optimal and LDR data sets.

random data sets were generated in order to provide a robust statistic of the random strategy accuracy.

Since neural network training depends on initial weights, we trained the MLP 50 times with different random weight initialization, allowing for a maximum number of cycles for each training. After these 50 trainings, we selected the MLP that had the smallest training error.[2] That time will be further reduced in the future by application of the method described in [25], based on the correlation between the MLP performance at convergence and its performance early in the training process, which allows discarding a model even before its training has been completed.

Fig. 4 shows the histogram of the estimated generalization errors of the 500 neural networks. The two lines are the estimated generalization errors of neural networks that learned on the D-optimality and the LDR-Bagging data sets.

As expected, both experimental design techniques led to better results than a random selection of experiments: models built on D-optimal training sets outperformed the random strategy in 86% of the cases, and the LDR method outperformed random selection in 91% of the cases.

- **Comparison between D-optimality and LDR active learning:** Because the cost function of neural networks has local minima, a statistical comparison between two experimental planning methods requires training the network with different initial values of the parameters. The 1000 different neural networks were trained on each data set. Each neural network was selected on its training mean square error among 50 different neural networks. Fig. 5 shows the distribution of the generalization error estimates.

  For the LDR method, the average generalization error is $\mu_b = 3 \times 10^{-3}$. For D-optimal design, the average is $\mu_d = 5.7 \times 10^{-3}$. The models that learned on LDR samples appear to be more efficient than the D-optimal ones. Indeed, 73% of the LDR models have a generalization error smaller than 2×10^{-3} against 26% in the D-optimal case.

[2]The mean square error on the training set is correlated with the generalization error when no measurement uncertainty is present.

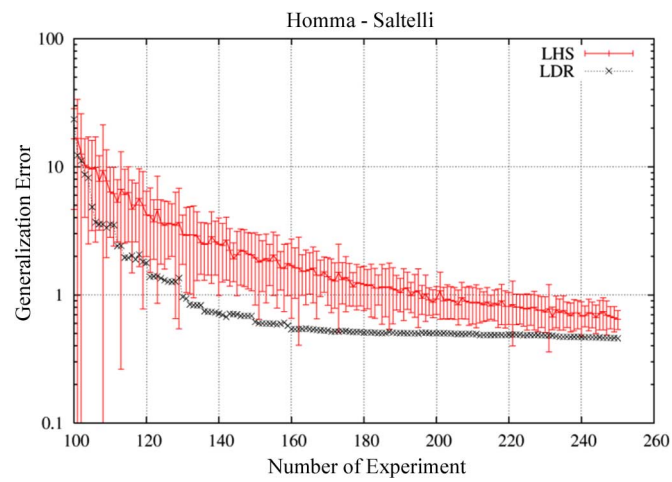

Fig. 6. Comparison between LHS and LDR selection. To approximate the Homma–Saltelli function, the models were sought as linear combinations of Legendre polynomials of degree six. The prediction variance was estimated by (virtual) leave-one-out. For LHS strategy, in every iteration $k = 1, \ldots, 150$, 50 LHS samples of size $100 + k$ were generated. The mean and the standard deviation are shown. For LDR strategy, a single point is added at each iteration. The selected point is the point for which the prediction variance is maximum.

2) Results for LDR-Leave-One-Out Method: We compared the generalization error, estimated by Monte Carlo, of models constructed on several samples of the same size (100–250 experiments), generated by LDR leave-one-out and by LHS. In order to obtain a robust comparison, we used 50 LHS samples for each sample size, and we computed the average and the standard deviation of the generalization error.

Fig. 6 shows the average evolution of the generalization error, and its standard deviation, of models that learned on both LHS and LDR samples. In real applications, the initial samples (100 experiments) would be based on low-discrepancy mathematical series, which are more robust, on average, than the LHS samples [26].

In that case, the LDR-leave-one-out active learning appears to be more efficient than LHS. For samples of identical size, the generalization error of models that learned on LDR samples is smaller than the average generalization error of LHS by at least the standard deviation of LHS generalization error.

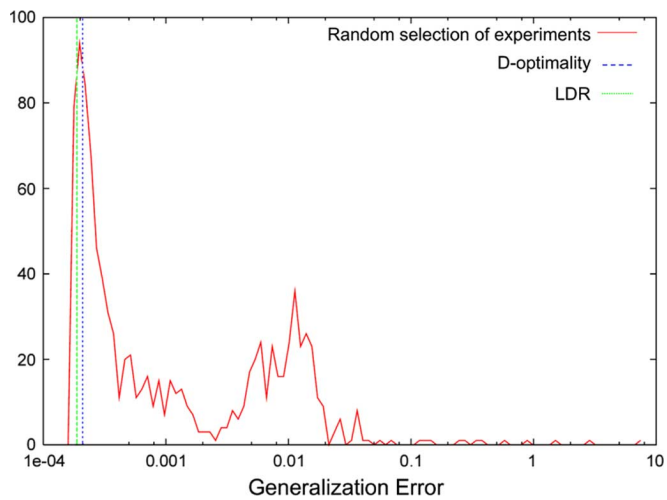

Fig. 7. Comparison between D-optimal design, LDR design, and random strategy.

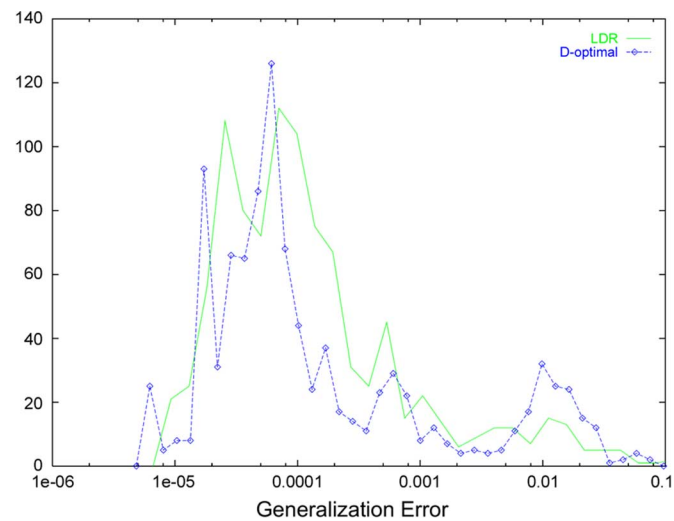

Fig. 8. Generalization error distribution of 1000 MLPs trained on D-optimal and LDR data sets.

B. Friedman Benchmark

In that case, the data-generating function is the Friedman function

$$y(\mathbf{x}) = \theta_1 \sin(\pi x_1 x_2) + \theta_2 (x_3 - \theta_3)^2 + \theta_4 x_4 + \theta_5 x_5$$

with

$$\theta_1 = \theta_4 = 0.4 \quad \theta_2 = 0.8 \quad \theta_3 = 0.5 \quad \theta_5 = 0.2$$

and

$$x_i \in [0;1] \qquad \forall i = 1, \ldots, 5. \tag{24}$$

An initial data set of 100 experiments was generated by LHS.

For this benchmark, we used the same test procedure used in Homma–Saltelli benchmark. In the following, the models are feedforward neural networks (MLPs) with a single layer of six hidden neurons.[3] The purpose of experimental planning was to supplement the initial training set of 100 examples with 30 additional examples. The generalization error was estimated by the Monte Carlo integration method (see Section IV-C).

1) Results for LDR-Bagging Method:

- **Comparison between D-optimality, LDR active learning, and random sampling of variable space:** We estimated the accuracy of the D-optimality and the LDR active learning planning techniques with respect to the accuracy of a random strategy with 500 random data sets. In each case, we used an accurate neural network selected on its training mean square error among 50 different neural networks.
  Fig. 7 shows the histogram of the estimated generalization errors of the 500 neural networks. The two lines are the estimated generalization errors of neural networks that learned on the D-optimality and the LDR-bagging data sets.
  As expected, both experimental design techniques led to better results than a random selection of experiments: models built on D-optimal training sets outperformed the random strategy in 94.8% of the cases, and the LDR method outperformed random selection in 96.7% of the cases.
- **Comparison between D-optimality and LDR active learning:** We performed a statistical comparison between D-optimality and LDR active learning. The 1000 different neural networks were trained on each data set. Each neural network was selected on its training mean square error among 50 different neural networks. Fig. 8 shows the distribution of the generalization error estimates. In that case, both experimental design techniques have the same accuracy.

[3]Models with six hidden neurons gave a good bias-variance tradeoff.

C. Engineering Application

For this application, we used a data set of more than 2000 real examples generated by a simulation model of a physical process. The quantity to be predicted is the multi-keV x-ray conversion efficiencies in the context of multi-keV x-ray production from prepulsed germanium foils.

An initial data set of 35 experiments was generated by LHS. For this benchmark, we used the same test procedure used in Homma–Saltelli and Friedman benchmarks.

In the following, the models are feedforward neural networks (MLPs) with three variables and a single layer of six hidden neurons. The purpose of experimental planning was to supplement the initial training set of 35 examples with 36 additional examples. The generalization error was estimated by the Monte Carlo integration method (see Section IV-C).

1) Results for LDR-Bagging Method:

- **Comparison between D-optimality, LDR active learning and random sampling of variable space:** In the previous cases, the efficiency of D-optimal planning and of LDR active learning were compared to the efficiency of a random strategy with 500 random data sets. In each case, we used an accurate neural network selected on its training mean square error among 50 different neural networks.
  Fig. 9 shows the histogram of the estimated generalization errors of the 500 neural networks. The two lines are the estimated generalization errors of neural networks that

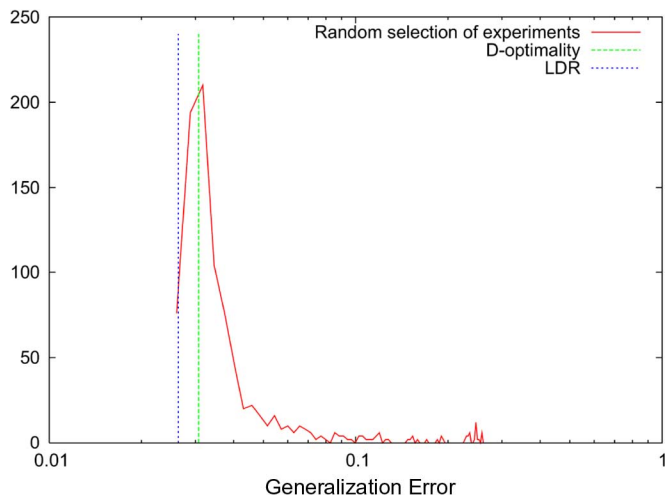

Fig. 9. Comparison between D-optimal design, LDR design and random strategy.

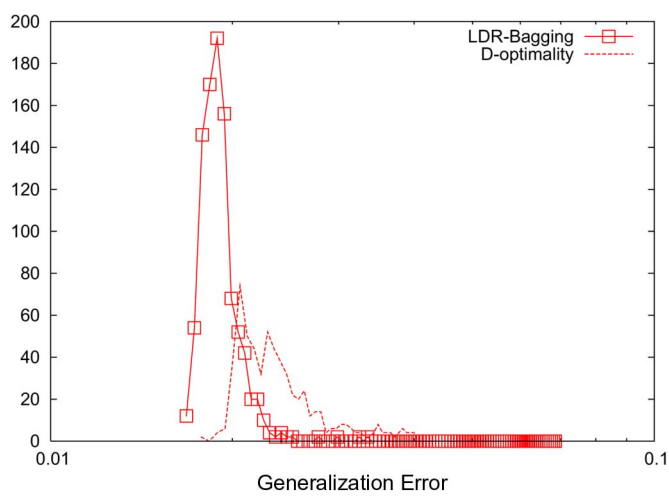

Fig. 10. Generalization error distribution of 1000 MLPs trained on D-optimal and LDR data sets.

learned on the D-optimality and the LDR-bagging data sets.

As expected, both experimental design techniques led to better results than a random selection of experiments: models built on D-optimal and LDR training sets outperformed the random strategy.

- **Comparison between D-optimality and LDR active learning:** We performed a statistical comparison between D-optimality and LDR active learning. The 1000 different neural networks were trained on each data set. Each neural network was selected on its training mean square error among 50 different neural networks. Fig. 10 shows the distribution of the generalization error estimates.

  The models that learned on LDR samples appear to be more efficient than the D-optimal ones.

  For finding the point of maximal prediction variance, the computational burden is the following: train 50 MLPs for selection and train 200 MLPs for computing the prediction variance with 200 replicates. The planning of an experiment takes a few minutes on a today's personal computer (PC), which is a negligible overhead with respect to the time necessary for performing the numerical experiment itself.

VII. Conclusion

A new active learning strategy (LDR), intended for use in the context of the planning of numerical experiments, has been described. The traditional optimal methods for experimental design give optimum data sets by minimizing the variability of the parameters due to experimental noise. In a context of numerical experiments, no experimental noise is present, so that the traditional approaches are not relevant. In order to generate a data set, the LDR method estimates the variance of the prediction of several models around the bagged predictor, and plans a new experiment at the location, in the space of variables, where the estimated prediction variance is maximal. The procedure is somewhat computer intensive, because it is based on resampling, but the computation time necessary for planning an experiment is negligibly small as compared to the computation time required by the experiment itself. A comparison between the prediction errors of models that learned on data sets designed by LDR and D-optimal design leads to the conclusion that the LDR method gives promising results in terms of quality of models that learned on such designs.

References

[1] V. V. Fedorov, *Theory of Optimal Experiments*. New York: Academic, 1972.

[2] J.-P. Gauchi, "Plans d'expériences optimaux pour modèles linéaires," in *Plans d'expériences—Applications à l'entreprise*. Paris, France: Editions Technip, ch. 7.

[3] J. Kiefer, "Optimum experimental designs," *J. Roy. Statist. Soc.*, vol. 21, pp. 272–319, 1959.

[4] J. Kiefer and J. Wolfowitz, "Optimum designs in regression problems," *Ann. Math. Statist.*, vol. 30, pp. 271–294, 1959.

[5] H. P. Wynn, "The sequential generation of D-optimum experimental designs," *Ann. Math. Statist.*, vol. 41, pp. 1655–1664, 1970.

[6] J.-P. Vila, "Local optimality of replications from a minimal D-optimal design in regression: A sufficient and a quasi-necessary condition," *J. Statist. Planning Inference*, vol. 29, pp. 261–277, 1991.

[7] D. MacKay, "Information-based objective functions for active data selection," *Neural Comput.*, vol. 4, pp. 590–604, 1992.

[8] D. Cohn, "Neural networks exploration using optimal experiment design," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1994, vol. 6.

[9] S. Issanchou and J.-P. Gauchi, "Plans d'expériences optimaux pour réseaux de neurones," presented at the ChimioMetrie 2004, Paris, France, 2004.

[10] M. Witczak, "Toward the training of feed-forward neural networks with the D-optimum input sequence," *IEEE Trans. Neural Netw.*, vol. 17, no. 2, pp. 357–373, Mar. 2006.

[11] C. R. Rao and H. Toutenburg, *Linear Models—Least Squares and Alternatives*, ser. Springer Series in Statistics. New York: Springer-Verlag, ch. 7.

[12] T. J. Mitchell, "An algorithm for the construction of D-optimal experimental designs," *Technometrics*, vol. 16, pp. 203–210, 1974.

[13] D. M. Bates and D. G. Watts, *Nonlinear regression analysis and its applications*. New York: Wiley, 1988.

[14] G. Monari and G. Dreyfus, "Local overfiting control via leverages," *Neural Comput.*, vol. 14, pp. 1481–1506, 2002.

[15] R. L. Iman, J. C. Helton, and J. E. Campbell, "An approach to sensitivity analysis of computer models, Part I. Introduction, input variable selection and preliminary variable assessment," *J. Quality Technol.*, vol. 13, pp. 174–183.

[16] A. C. Atkinson and A. N. Donev, "The construction of exact D-optimal designs with application in blocking response surface designs," *Biometrika*, vol. 76, pp. 515–526, 1989.

[17] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Mach. Learn.*, vol. 15, pp. 201–221, 1994.

[18] P. Melville and R. Mooney, "Diverse ensembles for active learning," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, Canada, 2004, pp. 584–591.

[19] S. Thrun and K. Möller, "Active exploration in dynamic environments," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1992, vol. 4.

[20] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *Proc. 17th Int. Conf. Mach. Learn.*, San Francisco, CA, 2000, pp. 839–846.

[21] K. K. Sung and P. Niyogi, "Active learning for function approximation," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1995, vol. 7.

[22] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[23] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. London, U.K.: Chapman & Hall, 1993.

[24] A. Saltelli and T. Homma, "Sensitivity analysis for model output, performances of black box techniques on three international benchmark exercises," *Comput. Statist. Data Anal.*, vol. 13, pp. 73–94, 1992.

[25] L. A. Feldkamp, D. V. Prokhorov, and C. F. Eagen, "Multiple-start directed search for improved NN solution," in *Proc. Int. Joint Conf. Neural Netw.*, Budapest, Hungary, Jul. 2004, pp. 991–996.

[26] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia, PA: SIAM, 1992.

Stéphane Gazut received the M.Sc. degree in computer science from Ecole Centrale d'Electronique, Paris, France, in 2003, the M.Sc. degree in information engineering from Université Paris XI, Paris, France, in 2003, and the Ph.D. degree in information engineering from the Systems and Structures Modelization Department, Commissariat à l'Energie Atomique, Gif-sur-Yvette, France, in 2007.

His current research interests include design of experiments theory and machine learning.

Jean-Marc Martinez received the Doctorat ès Sciences in physics from Université Paris XI, Paris, France, in 1983.

Currently, he is with the Systems and Structures Modelization Department, Commissariat à l'Energie Atomique, Gif-sur-Yvette, France. His activities are devoted to machine learning and uncertainties modeling in the field of numerical simulation.

Gérard Dreyfus (M'83–SM'89) received the Doctorat ès Sciences from Université Pierre et Marie Curie, Paris, France, in 1976.

Since 1982, he has been the Professor of Electronics at Ecole Suprieure de Physique et de Chimie Industrielles de la ville de Paris (ESPCI-Paristech), Paris, France and Head of the Electronics Research Department. His activities are devoted to machine learning, from neurobiological modeling to industrial applications.

Yacine Oussar graduated from Ecole Nationale Polytechnique d'Alger, in 1993, with an engineering degree in automatic control and received the Doctorat degree in nonlinear process modeling using neural and wavelet networks from Université Pierre et Marie Curie, Paris, France, in 1998.

He is an Assistant Professor at Ecole Suprieure de Physique et de Chimie Industrielles de la ville de Paris (ESPCI-Paristech), Paris, France. His research activities are nonlinear modeling using neural networks and support vector machines, model selection, and gray box modeling.