To the Graduate Council:

I am submitting herewith a dissertation written by Mark A. Buckner entitled "Learning from Data with Localized Regression and Differential Evolution." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Nuclear Engineering.

Robert E. Uhrig, Major Professor

We have read this dissertation
and recommend its acceptance:

J. Wesley Hines

Hamparsum Bozdogan

Laurence F. Miller

Andrei Gribok

Accepted for the Council:

Anne Mayhew

Vice Provost and Dean of
Graduate Studies

(Original signatures are on file with official student records.)

# LEARNING FROM DATA WITH LOCALIZED REGRESSION AND DIFFERENTIAL EVOLUTION

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Mark A. Buckner
May 2003

# Dedication

To the Ontological Trinity, the transcendental "Ultimate Prior", who is the precondition for the intelligibility of all reason, science and human endeavors; to the One in whom we live and breathe and have our being and without whom nothing would *be,* or *be knowable;* I humbly dedicate this work.

"For now we see in a mirror dimly, but then face to face; now I know in part, but then I will know fully just as I also have been fully known.[1] " It is "In Your light we see light[2]."

---

[1] 1 Corinthians 13:12

[2] Ps. 36:9

# Acknowledgements

I would like thank my family, friends, fellow coworkers and acquaintances whose love, encouragement, suggestions and prodding have made it possible for me to complete the Doctor of Philosophy in Nuclear Engineering. I especially want to acknowledge and thank the following.

Dr. Uhrig who has served as my major professor and mentor for the past several years has been instrumental in more ways than I can enumerate. I am especially grateful for his encouragement, prodding and intercession in overcoming the many obstacles I've encountered along the way.

Dr Gribok and Dr. Hines introduced me to the fields of ill-posed inverse problems, localized regression, regularization and evolutionary computing. I am grateful for their scientific supervision and the stimulating and challenging discussions related to application of evolutionary computing to localized regression. I am especially grateful to Dr Gribok for his friendship, patience, and technical insight and for the hours he spent discussing and molding many of the ideas presented in this dissertation.

I am grateful to Dr. Bozdogan for introducing me to his information complexity approach to multivariate regression and for his willingness to serve on my committee.

I am grateful to Dr. Miller for his encouragement and friendship throughout my graduate studies and for his willingness to serve on my committee.

I would like to thank Dr. Dodds for his encouragement both direct and indirect and for his leadership as department head, which has fostered an environment promoting creative and innovative research.

I am grateful to Dr. Aleksey M. Urmanov for his discussions, comments and suggestions during the early part of this research and specifically for his contributions on the objective functions (CL, ICOMPRPS, and GCV) used for optimal selection of localized ridge regression parameters.

I would like to acknowledge and thank UT-Battelle, the Oak Ridge National Laboratory, and the Engineering, Science and Technology Division for providing the part-time sabbatical to complete the research and write the dissertation.

I want to thank my fellow coworkers and the Defense Logistics Agency SMART Lab team Terri Rose, Richard Crutcher, Bobby Whitus and Simon Rose, and our Sponsor Jim Jenkins for their understanding, patience and encouragement.

I thank Dr. David W. Hall for his unceasing encouragement and occasional kick in the seat of the pants. His friendship and fellowship has been a treasure that will never be forgotten.

I also want to specifically acknowledge and thank Dr. Cyril Goutte for making the AMKREG code available and his responses to my many questions and Dr. James McNames for the code to generate the RampHill data set.

Last but by no means least, I want to express my deep appreciation and love to Lisa my precious wife, and our incredible children Sydney Noelle, Luke Isaac, and Madeline Adair. Without your encouragement, patience prayers and late night/early morning hugs I would have given up long ago. Daddy is finished and now we can play! This is for you.

# Abstract

*Learning from data* is fast becoming the rule rather than the exception for many science and engineering research problems, particularly those encountered in nuclear engineering. Problems associated with learning from data fall under the more general category of *inverse problems.* A *data-drive inverse problem* involves constructing a *predictive model* of a *target system* from a collection of input/output observations. One of the difficulties associated with constructing a model that approximates such unknown *causes* based solely on observations of their *effects* is that collinearities in the input data result in the problem being *ill-posed*. *Ill-posed problems* cause models obtained by conventional techniques, such as linear regression, neural networks and kernel techniques, to become unstable, producing unreliable results. Methods of regularization using ordinary ridge regression (ORR) and kernel regression (KR) have been proposed as viable solutions to ill-posed problems. Successful application of ORR and KR require the selection of optimal parameter values—ridge parameters for ORR and bandwidth parameters for KR. The common practice for both methods is to select a single parameter based on minimizing an objective function which is an estimate of empirical risk. The single parameter value is then applied to all predictor variables indiscriminately, in a sort of *one-size-fits-all* fashion. Versions of ORR and KR have been proposed that make use of individual *localized* ridge and a matrix of *localized* bandwidth parameters that are optimally selected based on the relevance of their associated predictor variables to reducing empirical risk. While the practical and theoretical value of both *localized regression techniques* is recognized they have obtained limited use because of the difficulties associated with selecting multiple optimal ridge parameters for localized ridge regression (LRR)—defined as the localized ridge regression problem—and multiple optimal bandwidth parameters for localized kernel regression (LKR)—defined as the localized kernel regression problem—particularly for multivariate predictor data with more than four variables.

This dissertation introduces a method of selecting optimal *ridge parameters* for LRR and a method of selecting a matrix of optimal *bandwidth parameters* for LKR based on the use of Differential Evolution (DE), a population based direct search global optimization technique. Three different objective functions, selected as prediction risk estimators, were developed and evaluated for LRR: *Mallows' CL*, an Information Complexity (ICOMP) based method of regression parameter selection (ICOMPRPS), and Generalized Cross-Validation (GCV). Leave-one-out cross-validation (LOO-CV) was used as the objective function for LKR.

Including the two methods of selecting optimal localized regression parameters the original contributions to the field of learning from data described in this dissertation are: i) DE automated selection of localized ridge regression parameters (DEALRR) using an objective function selected as an estimator of prediction

risk, ii) DE automated selection of a vector of bandwidth parameters for localized kernel regression ($DEALKR_D$) using LOO-CV as an objective function, iii) a method of optimizing the full bandwidth matrix for localized kernel regression using DE ($DEALKR_F$), iv) automatic selection of relevant input variables using $DEALKR_D$, and v) a method of detecting local minima (ModelM) in multidimensional data.

Case studies based on several practical examples using both artificial and real world data demonstrate i) the superior performance of DEALRR over ordinary least-squares and ordinary ridge regression for a) variable prediction and b) inferential sensing, ii) the superior performance of $DEALKR_D$ and $DEALKR_F$ over global kernel regression, and a hybrid conjugate gradient plus line search method for selecting a vector of LKR bandwidth parameters, for a) output target variable prediction, b) function approximation and c) inferential sensing, iii) the superior performance of $DEALKR_D$ for input variable selection, iv) the viability of optimally selecting the full bandwidth matrix LKR using $DEALKR_F$ and v) the ability to detect the presence of local minima in multidimensional data using ModelM.

# Contents

# List of Tables

# List of Figures

Chapter 1

# Introduction

This chapter introduces the problems associated with *learning from data* that are addressed by this dissertation by placing them within the context of the general learning problem common to all scientific research.

Section 1.1 defines the five problems which are: i) the localized ridge regression problem, ii) the localized kernel regression problem, iii) the full bandwidth matrix optimization problem, iv) the relevant predictor variable selection problem, and v) the local minima problem.

This findings of the research described in this document focuses on the application of differential evolution – a population based direct search global optimization technique – as a multi-parameter optimization engine for solving the localized ridge regression, localized kernel regression and full bandwidth matrix optimization problems.

Section 1.2 describes the context and motivation for this research as it relates to regularization of ill-posed problems and introduces localized ridge regression, localized kernel regression and the use of evolutionary algorithms for multi-parameter optimization.

Section 1.3 describes the author's significant contributions and section 1.4 closes the chapter with an overview and organization of the dissertation.

## 1.1 Problems addressed and Review of State of the Art

This section defines the specific problems addressed by this research as particular instances of the more general *learning problem* that is common to all *inverse problems* in science and engineering. The *learning problem* is defined as:

**Definition 1.1**

*Given a set D of n independent input-output observations,* $D = (\mathbf{x}_i, \mathbf{y}_i)$, *taken from the joint probability distribution* $p(\mathbf{X}, \mathbf{Y}) = p(\mathbf{Y} \mid \mathbf{X}) p(\mathbf{X})$, *where* $\mathbf{x}_i = (x_{1,i} \ldots x_{m,i}, \ldots x_{d,i})$ *is a d-dimensional vector of input variables and* $\mathbf{y}_i$ *is a vector target variable(s), develop a model* $\hat{m}(\mathbf{x}_i)$ *that is an estimator of the underlying model* $m(\mathbf{X})$ *(based on induction) that will accurately predict a new target variable* $\hat{\mathbf{y}}_{new}$ *when presented with a new input vector* $\mathbf{x}_{new}$ *given* $\mathbf{x}_{new}$ *is not contained in D,* $\mathbf{x}_{new} \notin D$.

### 1.1.1 Automatic Selection of Optimal Ridge Parameters for Localized Ridge Regression

The *localized ridge regression problem* is defined as:

**Definition 1.2**

*Given a localized ridge regression model*

$$\hat{y} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\lambda_i}$$

*where*

$$\hat{\boldsymbol{\beta}}_{\lambda_i} = (\mathbf{X}^T\mathbf{X} + \mathbf{V}\lambda_i^2\mathbf{V}^T)^{-1}\mathbf{X}^T y = \sum_{i=1}^{n} \frac{\mathbf{s}_i^2}{\mathbf{s}_i^2 + \lambda_i^2}\left(\mathbf{u}_i^T y\right)\cdot \mathbf{v}_i$$

*and* $\lambda_i$ *are the localized ridge parameters, find the optimal values of* $\lambda_i$ *that minimize the empirical risk of the model.*

For many real world predictive modeling tasks it is common for multidimensional predictor data $\mathbf{X}$ to contain collinear inputs. Inputs are collinear if they change together or if one can be represented as a linear combination of the other. Collinearity in the data causes instabilities in linear models because the matrix product $\mathbf{X}^T\mathbf{X}$ used to solve for the regression coefficients will be nearly singular, resulting in some of the model coefficients being very large. This has the ill-effect of small changes in the model inputs producing large changes in the model outputs, a classic definition of an ill-posed problem. Methods of regularization have been developed to address this problem, one of which is ridge regression. Ridge regression adds a scalar term and the identity matrix to the matrix product so that it becomes $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T$. Localized

ridge regression takes this concept a step farther using a vector of $\lambda$'s, each one appropriately selected for each component dimension.

The absence of a reliable method of automatically selecting optimal ridge parameters for localized ridge regression has severely limited the application of localized ridge regression even though various authors have suggested its theoretical value and anticipated improved performance over ordinary ridge regression [74,104,152,153,151,238].

This work describes a method developed to address the problem of automatically selecting optimal regularization parameters for localized ridge regression using Differential Evolution (DE), a population-based direct-search evolutionary algorithm. The method called **D**ifferential **E**volution **A**utomated selection of **L**ocalized **R**idge **R**egression parameters (DEALRR) selects optimal parameters by minimizing an objective function selected as an estimator of prediction risk (*empirical risk*). DEALRR is shown to outperform ordinary least squares, ordinary ridge regression, and principle component regression, for a prediction problem using real world data.

### 1.1.2 Automatic Selection of Optimal Bandwidth Parameters for Localized Kernel Regression

Thus *localized kernel regression problem* is defined as follows:

**Definition 1.3**

*Given a localized kernel regression model:*

$$\hat{y}(\mathbf{q};\mathbf{H}) = \hat{m}(\mathbf{q};\mathbf{H}) = \frac{\sum_{i=1}^{n} K(d_{\mathbf{H}}^2(\mathbf{x}_i,\mathbf{q}))y_i}{\sum_{i=1}^{n} K(d_{\mathbf{H}}^2(\mathbf{x}_i,\mathbf{q}))} \quad \hat{y} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\lambda}$$

*where the bandwidth matrix could be a scalar,* $\mathbf{H} \in S$*, or a diagonal matrix,* $\mathbf{H} \in D$*, find the optimal values for* $\mathbf{H}$ *that minimize the empirical risk of the model.*

Kernel regression is a non-parametric regression method that is widely used in science and engineering modeling applications, partly because it does not require knowledge of an underlying model for which parameters are sought, yet it provides a flexible way of generating prediction models by learning the relationship between input-output observations $(x_1, y_1)\ldots(x_n, y_n)$. When presented with a new input $x$, kernel regression estimates an output $\hat{y}$ based only on the outputs of those stored input-output observations

that have corresponding inputs that are *close* to $x$. Close is defined by a *distance measure* and a *kernel shape* whose size or width is controlled by a bandwidth parameter.

Localized kernel regression (LKR) is a multidimensional extension of kernel regression that uses a matrix of bandwidth parameters optimally selected for each input dimension instead of a single bandwidth parameter. The problem of automatically selecting a set of optimal bandwidth parameters is an area of active research and one for which there is no clear solution [85,127,136,183,239,235,236,238]. Therefore the most common implementations of kernel regression use a single scalar bandwidth parameter – a referred to as the "global" bandwidth – for all input dimensions. This is analogous to the use of a single ridge parameter in ordinary ridge regression.

The practice of using a global bandwidth parameter assigns all inputs equal weight without any other consideration. It would be much wiser to select bandwidth parameters for each input variable based on its relevance to producing accurate estimates of $\hat{y}$. While this intuitive concept is frequently discussed in the literature its acceptance and implementation has been restricted for the same reason as localized ridge regression, lack of a compelling method to automatically select optimal bandwidth parameters.

Various methods of automatically selecting bandwidth parameters have been advocated, but there is no clear winner [136,239]. Each appears to do well for a particular problem domain and poorly for others. Arguably one of the most promising methods, due to its speed and general robustness, is Adaptive Metric Kernel Regression (AMKREG) proposed by Goutte and Larsen [73]. AMKREG is a hybrid conjugate-gradient/line-search method (CGLS). But, as will be demonstrated in Chapter 5, AMKREG like all gradient-descent based techniques is susceptible to getting stuck in local minima and is sensitive to the initial "guess" or starting point used. Part of the general robustness of AMKREG stems from basing the initial "guess" on the distribution of the underlying data rather than relying on a random "guess". It will be been shown that while this approach to initialization may be good in many cases there are times when it will *guarantee* convergence to a sub-optimal, local minimum. This is particularly true when the data are sparse, which is a common scenario for many real-work problems.

In this work, a method is described as a solution to the problem of automatically selecting optimal diagonal bandwidth parameters for localized kernel regression using Differential Evolution (DE). The method called **D**ifferential **E**volution **A**utomated selection of bandwidth parameters for **L**ocalized **K**ernel **R**egression ($\text{DEALKR}_\text{D}$) – where the 'D' subscript indicates use of a diagonal bandwidth matrix, $\mathbf{H} \in D$ – selects optimal bandwidth parameters by minimizing the LOO-CVE, which is an estimate of generalization error or empirical risk. The performance of $\text{DEALKR}_\text{D}$ is compared to LKR using a global bandwidth parameter and to AMKREG (referred to as CGLS), for artificial and real world data sets. For cases where there were

no apparent sub-optimal local minima, the results of $DEALKR_D$ and CGLS were nearly identical. For cases where local minima were present in the error surface, $DEALKR_D$ outperformed CGLS. The results of $DEALKR_D$ and CGLS were always better than those of LKR using a global bandwidth parameter.

### 1.1.3 Optimization of the Full Bandwidth Matrix for Localized Kernel Regression

The *full bandwidth matrix optimization problem* is defined as follows:

**Definition 1.4**

*Given a localized kernel regression model:*

$$\hat{y}(\mathbf{q};\mathbf{H}) = \hat{m}(\mathbf{q};\mathbf{H}) = \frac{\sum_{i=1}^{n} K(d_{\mathbf{H}}^2(\mathbf{x}_i,\mathbf{q}))y_i}{\sum_{i=1}^{n} K(d_{\mathbf{H}}^2(\mathbf{x}_i,\mathbf{q}))} \quad \hat{y} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\lambda}$$

*with a full bandwidth matrix,* $\mathbf{H} \in F$ *, find the optimal values for the full matrix,* $\mathbf{H}$ *, that minimize the empirical risk of the model.*

This work introduces, for the first time, a method of solving the *full bandwidth matrix optimization problem* for localized kernel regression by using Differential Evolution (DE). The method called **D**ifferential **E**volution **a**utomated selection of bandwidth parameters for **L**ocalized **K**ernel **R**egression ($DEALKR_F$) – where the 'F' subscript indicates use of a full bandwidth matrix, $\mathbf{H} \in F$ – optimizes all parameters in the full bandwidth matrix by minimizing the LOO-CVE.

The performance of $DEALKR_F$ is compared to LKR using a global bandwidth parameter and to AMKREG (referred to as CGLS), for artificial and real world data sets. In all cases the LOO-CVE results of $DEALKR_F$ were better than all other methods. Even though it did not always yield the best training and test MSE, $DEALKR_F$ did what it was "asked" to do; it found the "best" globally optimal solution obtainable using LOO-CV as an objective function. $DEALKR_F$ enables the extraction of covariance information "hidden" in the off-diagonal data elements of the full bandwidth matrix that cannot be extracted by using only the diagonal elements, thus bringing a greater level flexibility and modeling capacity to LKR.

### 1.1.4 Selection of Relevant Predictor Variables

The *relevant predictor variable selection problem* is defined as:

**Definition 1.5**

*Given a set of n input-output observations,* $(\mathbf{x}_1, y_1)\ldots(\mathbf{x}_n, y_n)$ *, where* $\mathbf{x}_i = x_1,\ldots,x_d$ *is a d-dimensional vector of observed predictor variables of which some but not all* $x_i$ *have a causal affect on* $y_i$ *which is a scalar target variable; select only those predictor variables that are relevant to estimating a future target variable* $\hat{y}_t$ *given a new predictor vector* $\mathbf{x}_t$ *.*

Input variable selection is an active area of research [7,17,14,117,215,231,229] . It is crucial to the accuracy of the predicative model that only input variables that actually contribute to the predictive ability of the model be used. Inclusion of irrelevant parameters or exclusion of relevant parameters will increase the uncertainty of the predicative model [23,48,117]. It is shown that DEALKR affectively eliminates non-relevant input parameters by assigning them very large bandwidth parameters. Since only the variables with small bandwidths contribute to the reduction of the cross-validation error, those variables with large bandwidths can be deemed irrelevant and eliminated from the model reducing the model complexity, and uncertainty.

### 1.1.5 Detecting the existence of local minima

The *local minima problem* is defined as follows:

**Definition 1.6**

*Consider a model* $\hat{m}(\mathbf{x};\mathbf{p})$ *where* $\mathbf{p}$ *is a set of model parameters that control the estimation of* $\hat{y}_{new}$ *. Selecting the best set of parameters involves minimizing an objective function* $f(\mathbf{x};\mathbf{p})$ *that provides an indication of model accuracy. The local minima problem is to determine whether the error surface of the objective function being minimized has more than one minimum.*

The existence of local minima in an error surface calls into question solutions obtained by conventional approaches. Why should one result or model that is susceptible to getting trapped in local minima be favored over another? In these cases it might justify the potentially higher computational cost required to implement the techniques described in this work. A technique was developed to determine if local minima exist in a particular error surface and provide indication of when one should apply the more powerful, more computationally intensive techniques described in this work.

While it is possible for one to "see" or observe the existence of local minima in the error surface for data sets of one or two variables we do not have the this luxury once a data set exceeds two input dimensions. As will be shown in section 5.1.1.4 DEALRR and DEALKR are powerful modeling tools but their

increased power comes with some increase in computational cost, when compared with other more conventional techniques such as ordinary least squares, ordinary ridge regression, or even AMKREG. Consequently, it would not be prudent to blindly insist upon their use when simpler, more computationally efficient tools would be just as accurate.

The method developed to detect the existence of local minima capitalizes on the rapid convergence characteristics of the hybrid conjugate-gradient/line search (CGLS) technique. First we force the CGLS to start at multiple points that "cover" the data spaces of the localized ridge or bandwidth parameters and collect the results. A large standard deviation in any of the parameters would suggest the existence of local minima in the underlying error surface. This technique is applicable to problems of any dimension.

## 1.2 Context and Motivation for Localized Regression

According to Cherkassky and Mulier in order for a *learning process* to formulate a *unique* generalization model $\hat{m}(\mathbf{x}_i)$ from a set of finite data, it must possess the following characteristics [154 p. 40]:

1. *A wide, flexible set of approximating functions*: $f(\mathbf{x}, \omega), \omega \in \Omega$,

2. *A priori knowledge or assumptions* used to impose constraints on the potential classes of function in (1),

3. *An inductive principle or inference method* for combining a priori knowledge (2) and available training data to produce an approximation of the true but unknown dependency – stipulates what to do not how to do it, such as Empirical Risk Minimization (ERM), and

4. *A learning method,* such as Maximum likelihood estimators (MLE), linear regression, polynomial methods, or fixed topology ANNs.

A critical need for any *learning method* is that it must have an *optimization procedure* for parameter estimation. This is precisely the focus of this research which introduces the use of DE for the optimization of parameters associated with the *learning methods* of localized ridge regression and localized kernel regression.

### 1.2.1 Data-Driven Inverse Problems

Most *learning problems* in the field of science and engineering are *data-driven* and fall under the general category of *inverse problems*. By data-driven we mean learning problems that are required to develop

models in an inductive framework based on a collection of discrete input/output observations. An inverse problem involves development of a model that approximates underlying unknown *causes* based only on observations of their *effects*. Figure 1.1 depicts this *inverse modeling* scenario graphically.

In *supervised learning* the goal is to approximate an unknown relational mapping $f : \mathbf{X} \to \mathbf{Y}$ with a model $\hat{m}(\mathbf{X}, \mathbf{Y}) = \hat{f}(\mathbf{X}, \mathbf{Y}) = \hat{y}$ that produces an accurate estimate of $y$ given a set of input-output observations, $(x_1, y_1) \ldots (x_n, y_n)$.

The accuracy of the model is a measure of *empirical risk,* $R_{emp}$, and is described by a *distance measure* which relates the closeness of $\hat{y}$ to $y$. The best model $\hat{m}(\cdot)$ produces the smallest $R_{emp}$ when tested with an unseen set of data observations. A common distance measure used in regression is the $L_2$ loss function, $L_2(y, \hat{m}(x)) = (y - \hat{y})^2$. Thus $R_{emp}$ becomes the sum of the squared errors (SSE),

$$R_{emp} = SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad . \tag{1.1}$$

In *unsupervised learning* there is no distance measure and the goal is to discover patterns and relationships among the set of input variables. Probability distribution estimation and clustering are examples of unsupervised learning. Classification and regression are two examples of *supervised learning*. This work focuses on localized regression within the context of supervised learning and addresses the supervised learning problems associated with i) localized ridge regression and ii) localized kernel regression.

### 1.2.2 Ill-posed Problems

Inverse learning problems can be either *well-posed* or *ill-posed*. The French mathematician Hadamard was the first to describe well-posed and ill-posed inverse problems in 1902 [83,84]. Tikhonov and Arsenin extended Hadamard's conditions of well-posedness in [218]. To illustrate the concept of ill-posedness we will use a simple regression problem, and construct a regression model $\hat{m}(\cdot)$ that represents the relationship between a $d$-dimensional input set of predictor variables $\mathbf{X} = (X_1, \ldots X_m \ldots X_d)$ and a target output variable $Y$:

$$Y = m(\mathbf{X}) + \varepsilon \tag{1.2}$$

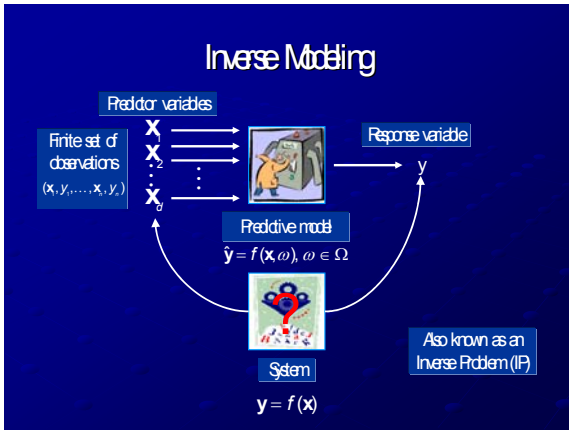



Figure 1.1 Graphical description of inverse modeling.

where, $\varepsilon$ is a Gaussian noise term. First we will describe an approach using a linear parametric model and then we will describe a semi-parametric modeling approach. Assuming a linear regression model we define $m(\cdot)$ in terms of a vector of regression coefficients or parameters $\boldsymbol{\beta} = (\beta_0, \ldots \beta_d)$ :

$$Y = \beta_0 + \beta_1 X_1 + \ldots \beta_d X_d + \varepsilon \Rightarrow \hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots \hat{\beta}_d x_d \tag{1.3}$$

Solving for $\hat{\boldsymbol{\beta}}$ is and inverse problem with the ordinary least squares solution (OLS), sometimes referred to as the maximum likelihood solution (MLS) for Gaussian $\varepsilon$ , is given by [132]

$$\hat{\boldsymbol{\beta}}_{ols} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T Y \,, \tag{1.4}$$

where, $\hat{\boldsymbol{\beta}}_{ols}$ is the vector or regression coefficient estimates. The OLS solution is an unbiased estimate of the true solution, if it exists. When the data matrix $\mathbf{X}$ is ill-conditioned, $\hat{\boldsymbol{\beta}}_{ols}$ has a large variance and becomes very unstable. We can see why if we re-write the OLS solution in terms of singular *value decomposition* (SVD) of the predictor variables $\mathbf{X} = \mathbf{U} \cdot diag(\mathbf{s}_i^{-1}) \cdot \mathbf{V}^T$

$$\hat{\boldsymbol{\beta}}_{ols} = \mathbf{V} \cdot diag(\mathbf{s}_i^{-1}) \cdot \mathbf{U}^T y = \sum_{i=1}^{d} \mathbf{s}_i^{-1} (\mathbf{u}_i^T y) \cdot \mathbf{v}_i \,, \tag{1.5}$$

where, $\mathbf{u}$ and $\mathbf{v}$ are the right and left eigenvectors of $\mathbf{X}$ of and $\mathbf{s}_i$ are the singular values of $\mathbf{X}$ . When $\mathbf{X}$ is ill-conditioned the last few singular values, which correspond to noise or irrelevant predictor components, are very close to zero. When inverted these small singular values produce very large regression coefficients. The large coefficients greatly amplify the noisy or irrelevant predictor variables which severely degrades

the predictive accuracy of the OLS solution. The affects of this are clearly seen when one considers the variance-covariance matrix of the OLS solution which is

$$Cov(\hat{\boldsymbol{\beta}}_{ols}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} = \sigma^2\mathbf{V}\cdot diag(\mathbf{s}_i^{-2})\cdot\mathbf{V}^T, \tag{1.6}$$

where $\sigma^2$ is the noise variance of $Y$. From equation (1.6), one can see that singular values close to zero result in a large variance in the solution rendering it statistically insignificant. The solution of $\boldsymbol{\beta}$ for $\mathbf{X}\boldsymbol{\beta} = Y$ is said to be well-posed if the following criteria are met [218]:

1. Existence: for each $y \in Y$ there exists a solution $\boldsymbol{\beta} \in \Omega$;

2. Uniqueness: the solution $\boldsymbol{\beta}$ is unique, e.g., for any pair of input vectors $\mathbf{x}, \mathbf{t} \in \mathbf{X}$ $\mathbf{x}\boldsymbol{\beta} = \mathbf{t}\boldsymbol{\beta}$, if, and only if, $\mathbf{x} = \mathbf{t}$;

3. Stability: with $\mathbf{X}\boldsymbol{\beta} = Y$ and $\mathbf{X}\tilde{\boldsymbol{\beta}} = \tilde{Y}$, $\lim_{Y\to\tilde{Y}} \tilde{\boldsymbol{\beta}} \to \tilde{\boldsymbol{\beta}}$ (with small changes in $Y$ the solution $\boldsymbol{\beta}$ is stable). This third criterion is the same as requiring the inverse operator $\boldsymbol{\beta}^{-1}$ to be continuous.

Conversely, if any of the above criteria are not meet, $\mathbf{X}\boldsymbol{\beta} = Y$ is said to be ill-posed.

Many predictive modeling engineering applications are ill-posed inverse learning problems for the following reasons. First, uniqueness is violated because one typically deals with sparse data sets and consequently there is not enough information in the training data to reconstruct a unique input—output mapping. Second, stableness is often violated due to *collinear* inputs or noise in the measurements. Third, there may not be enough information in the inputs for a solution to exist.

Figure 1.2 and Figure 1.3 present examples of 1D and 2D ill-posed problems. Both show how different solutions can be produced from the exact same training data, bringing one face-to-face with the problem of deciding which if any of the models are correct. Without any other prior knowledge of the true model which produced the data the selection process would be purely arbitrary.

According to Cherlassky and Mulier [154 p. 39], learning – the forming of generalizations from particular instances of training data (finite data sets) – is inherently ill-posed and solving it requires *a priori* knowledge in addition to the data.

Before describing regularization as a method for solving ill-posed problems we take a bief look at how much data a learning method needs in order to construct a model with a certain degree of accuracy.

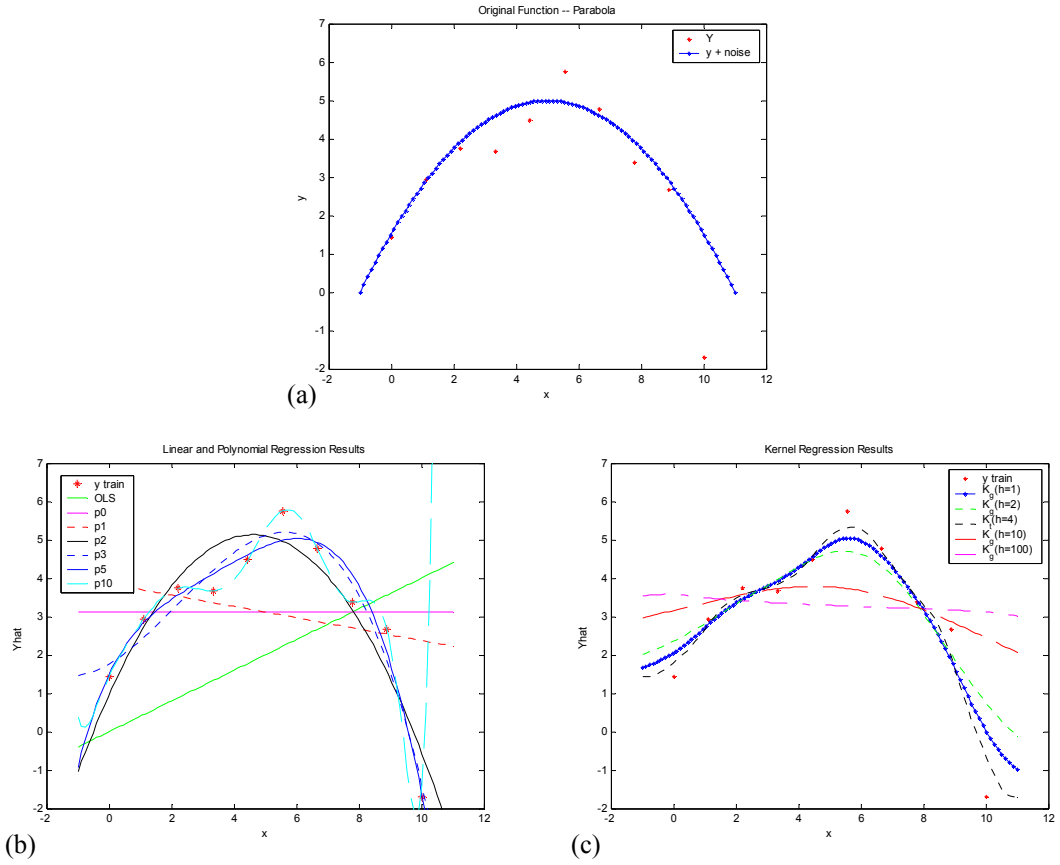



Figure 1.2 1D example of an Ill-posed problem: (a) Blue curve is the data generating function and red asterisks are the training points; (b) Results from linear regression and polynomial regression of varying degrees – where p0-p10 signify the orders of the polynomials; (c) Kernel regression results using a Gaussian kernel – where h=1 through h=100 indicate the kernel bandwidth.

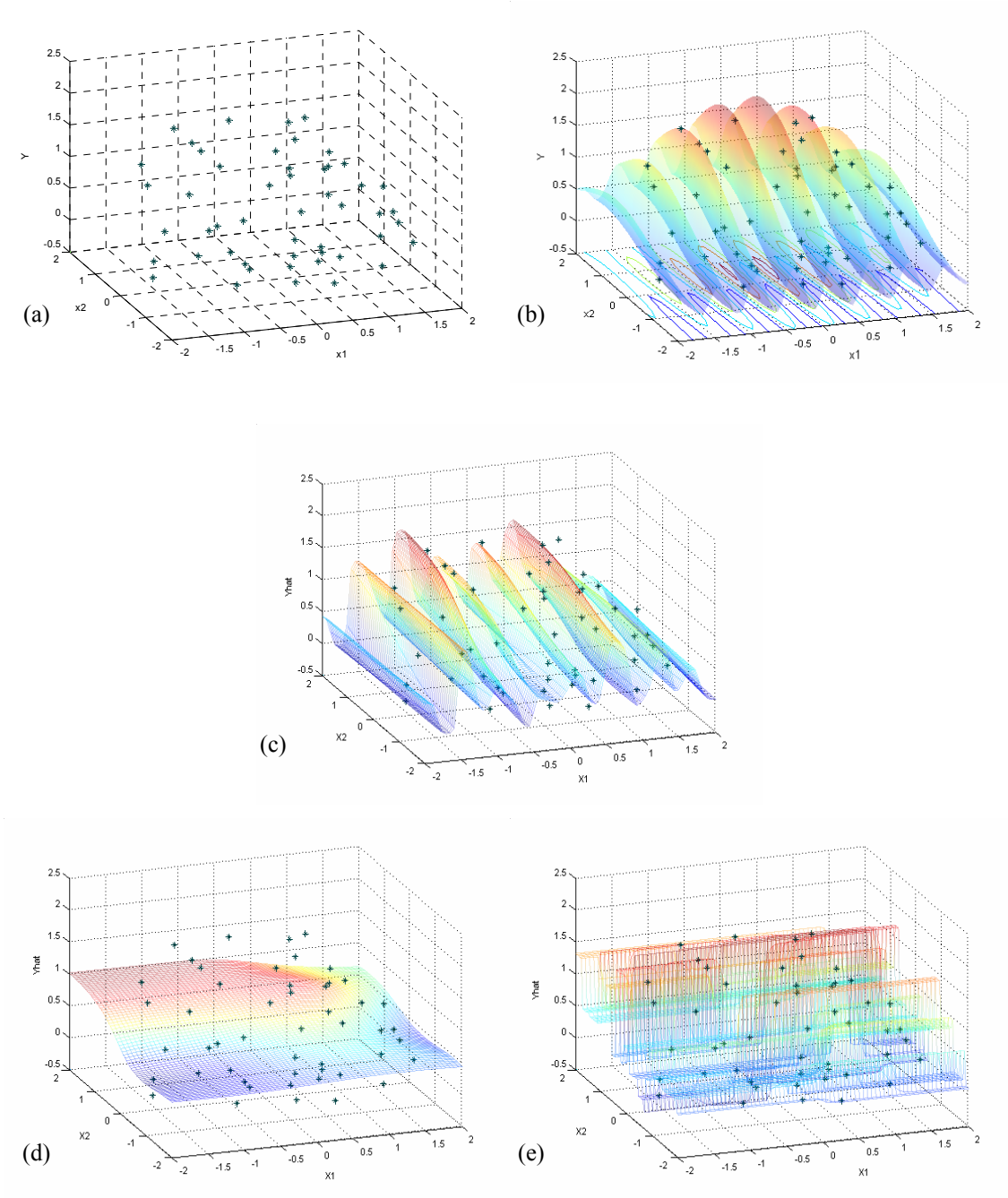



Figure 1.3 Illustration a a 2D ill-posed problem: (a) The training data; (b) data generating function is the surface plot and the asterisks are the training data; (c) – (e) are three different function approximations obtained from the same data set with training errors of 0.2291, 0.2291 and 5,776 and test MSEs of 0.2181, 0.2181 and 0.4172 respectively.

### 1.2.3 How Much Data is Enough?

The amount of data (discrete input-output data samples) a *learning method* needs in order to construct a model with a certain degree of accuracy depends upon i) the dimensionality, *d*, and ii) the degree of smoothness, *p,* of the class of functions being approximated [169]. For nonparametric polynomial regression the optimal rate of convergence $\varepsilon_n$ – how accurately a function can be approximated with *n* data samples – can be expressed as [169]:

$$\varepsilon_n = n^{-\frac{p}{2p+d}} \tag{1.7}$$

The number data points needed to approximate a function grows rapidly with *d* (the curse-of-dimensionality) and requires a greater amount of smoothness, *p* [169]. For $f(\mathbf{x}, \omega)$ with $p = 2$ and $d = 2$ to achieve an $\varepsilon_n = 0.05$, 8,000 samples are needed. If $d = 10$, $10^9$ samples are needed to achieve the same level of accuracy. Thus, when $d > 3$ or 4 one is forced to assume a *high degree* of smoothness. This is what provides the *a posteiori justification* of the smoothness assumption central to standard regularization techniques, which will be taken up next [169].

### 1.2.4 Regularization of Ill-posed problems

In addition to extending Hadamard's definition of well-posedness, Tikhonov proposed a method of turning ill-posed problems into well-posed problems called *regularization* [217]. The idea behind regularization is to *stabilize* the ill-posed problem by introducing some *a priori* information about the system, such as requiring smoothness in the input/output mapping of the model estimate $\hat{m}(\cdot)$. The objective is to create a model that *generalizes* well, that is it produces accurate targets estimates for predictor variables it has not "seen". Most regularization methods involve the addition of a *penalty term*, $\varepsilon_c$ to $R_{emp}$ equation (1.1),

$$\varepsilon_c = \left\| \mathbf{P}\hat{m} \right\|^2. \tag{1.8}$$

$\mathbf{P}$, which is dependent on the model $\hat{m}(\cdot)$, drives the creation of parsimonious models by restricting complexity consistent with the principle know as Occam's Razor [64]: "No more things should be presumed to exist than are absolutely necessary." $\mathbf{P}$ acts as a *stabilizer* of $\hat{m}(\cdot)$, requiring it to be smooth and continuous by penalizing non-smooth versions of the model $\hat{m}(\cdot)$.

### 1.2.5 Ordinary Ridge Regression

Many of the predictive modeling applications encountered in engineering are ill-posed due to collinearity in the predictor data variables. In fact, some systems such as fault-tolerant systems, drift detection systems, and automated process sensor calibrations monitoring, depend on collinear predictor variables to building successful and robust inferential models [79]. Collinear predictor variables cause traditional empirical modeling techniques such as linear regression, neural networks, kernel techniques and any other approach that does not employ regularization to construct very unstable models which produce unrepeatable results [100]. Examples of this can be found in most research fields.

Several penalized estimators have been proposed as regularization methods to deal with the instabilities in linear regression models caused by collinear inputs [42,71,78]. The most common method, termed *ordinary ridge regression* (ORR) adds a scalar term and the identity matrix to the matrix product so that it becomes $(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T$. Localized ridge regression uses a vector of $\lambda$'s, enabling the regularization of each input dimension as appropriate, instead of a one-size-fits-all approach, which will inevitably lead to a suboptimal solution by over-regularizing (over-smoothing) some input dimensions and under-smoothing others.

The absence of a reliable method of automatically selecting optimal ridge parameters for localized ridge regression is a problem that has severely limited its application even though various authors have suggested its theoretical value and improved performance over ordinary ridge regression [74,104,151,152,153,238].

Another way to regularize an ill-posed regression problem is with a *nonparametric* or *semi-parametric* modeling approach such as kernel regression. This approach is especially useful when little or nothing is know about the underlying function $m(\cdot)$

### 1.2.6 Kernel Regression

Kernel regression (KR) is an instance of a semi-parametric, *memory-based,* modeling technique sometimes referred to as *lazy learning* [15]. Kernel models generate target variable estimates using the predictor-target data pairs in memory $(x_1, y_1)\ldots(x_n, y_n)$, but only those with $x_i$'s that are *close* to the *query* point $q$. Four characteristics common to all memory-based learners are [4,144]:

1. A distance metric

2. How many nearby neighbors to considered

3. A weighting function (optional)

4. A method of predicting based on the local data.

A kernel functional, $m(\cdot)$, can be described by the following equation:

$$m(\cdot) = \hat{y} = \sum_{i=1}^{n} K_h(\mathbf{x}_i, \mathbf{q}) y_i \tag{1.9}$$

where $K_h(\mathbf{x}_i, \mathbf{q})$ is the kernel function with bandwidth *h,* possessing the following properties [24]:

1. $K_h(\mathbf{x}_i, \mathbf{q})$ is nonnegative

2. $K_h(\|\mathbf{x}_i, \mathbf{q}\|)$ is radially symmetric

3. $K_h(\mathbf{x}_i, \mathbf{q})$ largest when $\mathbf{q} = \mathbf{x}$

4. $K_h(\mathbf{x}_i, \mathbf{q})$ decreases monotonically with $|\mathbf{x} - \mathbf{q}|$.

A kernel that is radially symmetric uses a scalar bandwidth *h,* which is sometimes referred to as a global bandwidth parameter. The research described in this work presents a method of optimally selecting a vector of bandwidth parameters and a full matrix of bandwidth parameters for localized kernel regression using an evolutionary algorithm.

### 1.2.7 Evolutionary Algorithms for Multi-parameter Optimization

Evolutionary Algorithms (EA) have been successfully applied to solve complex engineering optimization problems. Arguably the best know representatives are Genetic Algorithms (GA) [67,107] and Evolutionary Strategies (ES) [188,178]. Others include Evolutionary Programming (EP) [59], Genetic Programming (GP) [122-125], Simulated Annealing (SA) [121,181], Particle Swarm Optimization/Theory (PSO) [50,97,118,119,164,165], and Self-Organizing Migrating Algorithm (SOMA) [190,247]. There are many good introductions [56] and surveys available [8, 60,61]. Specific references to the use of EAs for optimization include:

- Applications of EA to L1 norm errors-in-variable problems [163];

- Multi-parameter optimization using PSO [168];

- Multi-objective optimization with EA [36,37,249];
- Constrained optimization using EA [142], PSO [167];
- Objective function stretching suing PSO [166]; and
- Hybrid EAs for optimization [35].

A recent book by Corne et al. *New Ideas in Optimization* [38] provides an overview of some of the latest innovations in optimization.

The remaining Chapters describe how Differential Evolution (DE), a population-based direct-search EA, was successfully applied to solve the LRR problem: selecting optimal regression parameters for LRR and the LKR problems: selecting optimal bandwidth parameters for LKR. The next section describes the significant contributions that have resulted from this work.

## 1.3 Contributions

The author's significant contributions to the field of learning from data:

1. **Automatic Selection of Localized Ridge Regression Parameters:** This work describes a method of automatically selecting and optimizing a vector of localize ridge regression parameters by using differential evolution to minimize a cost function selected to be an estimate of prediction risk. This method is referred to as **DE A**utomated selection of **LRR** parameters (DEALRR). The prediction risk estimators implemented were a) Mallows' CL, which has be proven to be an unbiased estimate of prediction error [137]; b) an Information Complexity (ICOMP) based method of regression parameter selection (ICOMPRPS), and Generalized Cross-Validation (GCV).

2. **Automatic Selection of Bandwidth Parameters for Localized Kernel Regression:** A method of automatically selecting a diagonal bandwidth matrix of parameters for localized kernel regression is described that uses differential evolution to minimize the leave-one-out cross-validation error. This method e referred to as **DE A**utomated selection of bandwidth parameters for **LKR** ($DEALKR_D$)

3. **Method of Optimizing the Full Bandwidth Matrix for Localized Kernel Regression:** Many researchers have proposed the potential value in using off-diagonal elements of the bandwidth matrix but a viable method of optimizing such a large number of parameters has not been

proposed. The calculation of gradients for the cross-validation error with respect to the diagonal elements of the bandwidth matrix has enabled their optimization by use of gradient-descent techniques. But that has not been the case for the off-diagonal elements. A method will be described that uses DE to optimize the full bandwidth matrix by minimizing the LOO-CVE. This method is referred to as $DEALKR_F$.

4. **Automatic Selection of Relevant Input Variables with Localized Kernel Regression:** $DEALKR_D$ is show to be capable of automatically selecting relevant input variables. $DEALKR_D$ automatically assigns bandwidths to variables based on their relevance in reducing the cross-validation error: relevant inputs are assigned small bandwidth parameters while irrelevant input variables are assigned large bandwidth parameters. Consequently, variables with large bandwidths can be eliminated from the LKR model which would affectively reduce the bias of the model and therefore reduce the model uncertainty.

5. **Method of Detecting Local Minima in Multidimensional Data:** A **M**ethod **o**f **d**etecting **l**ocal **M**inima (ModelM) is presented that can be applied to data sets of any dimension. The method uses the rapid convergence characteristics of a hybrid conjugate-gradient/line search (CGLS) technique to *"survey"* the cross-validation error surface by controlling the starting points of a population of CGLS *"scouts"*. For data sets greater than two dimensions histograms of the bandwidth parameters for each dimension are used to visualize the results. If there is a single minimum and the error surface is sloped enough all CGLS results should converge to the exact same point on the error surface regardless of their starting point. This would produce histograms that have single peaks and very small standard deviations for all dimensions. Large standard deviations in any of the bandwidth histograms are a strong indicator that local minima exist and that DEALKR should be used.

The next section describes the organization and layout of the dissertation.

## 1.4 Organization

This chapter described the problems addressed by this work 1) automatic selection of optimal ridge parameters for localized ridge regression, 2) automatic selection of bandwidth parameters for localized multivariate kernel regression, 3) automatic selection of a full matrix of bandwidth parameters for localized multivariate kernel regression, 4) selection of relevant input variables and 5) how to determine if local minima exist in higher dimensional data sets. This chapter also provided a brief overview of the context of

and motivation for the application of differential evolution as a multi-parameter global optimizer for these problems.

Chapter 2 introduces localized ridge regression (LRR) in the context of linear regression, ordinary least squares (OLS), regularization and ordinary ridge regression (ORR) and briefly describes prior work on optimization of LRR parameters.

Chapter 3 describes localized kernel regression (LKR) with a discussion of distance measures, kernel shape and bandwidth selection. Prior work on optimization of localized kernel regression is reviewed specifically the work of Goutte and Larsen on AMKREG, highlighting its strength (speed) and weakness (local minima problem).

Chapter 4 takes up the topic of applying DE to the learning problems described earlier. The inner workings of DE are exposed, explaining how DE differs from other evolutionary algorithms, describing, population creation, mutation, crossover, recombination and selection. Also use of the objective functions DEALRR, $DEALKR_D$ and $DEALKR_F$ are described.

Chapter 5 illustrates the local minima problem and presents a method for detecting local minima demonstrating its effectiveness on artificial data.

Chapter 6 applies DEALRR, $DEALKR_D$, $DEALKR_F$ and ModelM to artificial and real world data sets demonstrating their superior performance over OLS, ORR, and LKR with both global bandwidths and diagonal matrix bandwidths selected using CGLS.

Chapter 7 concludes with a brief summary and provides some recommendations for future areas of research.

Chapter

2

# Localized Ridge Regression

This chapter provides background and introduction to Localized Ridge Regression (LRR) illustrating the progression from linear regression to ordinary ridge regression (ORR) to LRR..

## 2.1 Linear Regression

Recall from our discussion of ill-posed problems the simple regression problem described by equation (1.2) which we will rewrite as follows:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon \quad , \quad \varepsilon \sim N(0, \sigma^2 I_n)$$

where $\boldsymbol{\beta} = (\beta_0, \ldots \beta_d) \in R^d$ is dx1 vector of regression coefficients, $\mathbf{y}$ is an *n x 1* vector of response variables and variables $\mathbf{X} = (X_1, \ldots X_m \ldots X_d)$ **,** is an *n x d* matrix of predictor variables, and $\boldsymbol{\varepsilon}$ is an *n x 1* vector of the prediction errors. Also, recall the O*LS*:

$$\hat{\boldsymbol{\beta}}_{ols} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

where $\hat{\boldsymbol{\beta}}_{OLS}$ is a vector of OLS regression coefficient estimates. While the *OLS* solution, if it exists, is an unbiased estimate of the true solution, when $\mathbf{X}$ is ill-conditioned as is the case for collinear inputs, $\hat{\boldsymbol{\beta}}_{OLS}$ becomes extremely unstable, due to very small singular values (see equation (1.5)). In some cases these small singular values can be attributed to noisy, non-informative inputs variables but this is not always the case. In any event, near zero singular values introduce a large variance in the solution, making it statistically insignificant.

## 2.2 Ordinary Ridge Regression

ORR was introduced as a method of dealing with collinear or ill-conditioned data [104] to avoid the problem of instability. The ORR solution is:

$$\hat{\boldsymbol{\beta}}_{\lambda} = \left(\mathbf{X}^T\mathbf{X} + \lambda^2\mathbf{I}_d\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{2.1}$$

where $\lambda \geq 0$ is the *ridge parameter* and $\mathbf{I}_d$ is the $d \times d$ identity matrix. Now let us rewrite equation (2.1) in terms of the *SVD* of $\mathbf{X}$:

$$\hat{\boldsymbol{\beta}}_{\lambda} = \mathbf{V} \cdot diag\left(\frac{\mathbf{s}_i^2}{\mathbf{s}_i^2 + \lambda^2}\right) \cdot \mathbf{U}^T\mathbf{y} = \sum_{i=1}^{n} \frac{\mathbf{s}_i^2}{\mathbf{s}_i^2 + \lambda^2}\left(\mathbf{u}_i^T\mathbf{y}\right) \cdot \mathbf{v}_i = \sum_{i=1}^{d} f_i \boldsymbol{\rho}_i \mathbf{v}_i \tag{2.2}$$

with

$$f_i = \frac{s_i^2}{s_i^2 + \lambda^2},$$

defined as the filter factors and $\boldsymbol{\rho}_i = \mathbf{u}_i^T\mathbf{y}$ as the correlation coefficients. Notice that for $\lambda=0$, the ORR solution becomes the *OLS* solution; for $\lambda>0$, the solution is different. The filter factors determine if the information in the $i^{th}$ component is incorporated into the solution or damped. If $\lambda$ is large with respect to a singular value $\mathbf{s}_i$ the filter factor dampens the corresponding component while if $\lambda$ is small with respect to a singular value $\mathbf{s}_i$ its corresponding component is passed. Therefore, if chosen large enough $\lambda$ eliminates the destroying effect of the near zero singular values and makes the solution stable and statistically significant. The variance-covariance matrix is written as:

$$Cov\left(\hat{\boldsymbol{\beta}}_{\lambda}\right) = \sigma^2\left(\mathbf{X}^T\mathbf{X} + \lambda^2\mathbf{I}_d\right)^{-1}\mathbf{X}^T\mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \lambda^2\mathbf{I}_d\right)^{-1} = \sigma^2\mathbf{V} \cdot diag\left(\frac{\mathbf{s}_i^2}{\left(\mathbf{s}_i^2 + \lambda^2\right)^2}\right) \cdot \mathbf{V}^T. \tag{2.3}$$

From equation (2.3) we can see that as $\lambda \to \infty$ the variance of the corresponding solution goes to zero, $\sigma^2 \to 0$. While a decrease in the variance is good, in that it leads to a reduction of uncertainty in the model, increasing $\lambda$ also increases the bias term of the ORR solution. The general relationship between the uncertainty of a solution and its squared bias term and variance are depicted by Figure 2.1. This relationship is sometimes called the bias-variance dilemma.

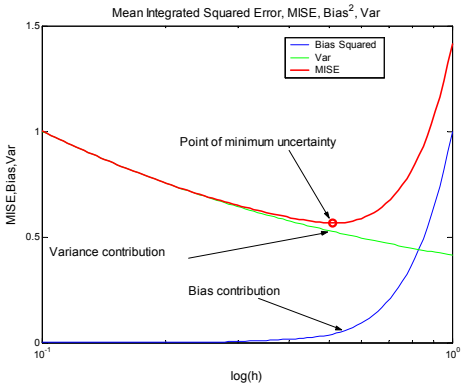



Figure 2.1 Graphical illustration of the relationship between model uncertainty, and the bias squared and variance, also know as the bias-variance dilemma.

Fortunately for us, Horel proved that there always exists some optimal ridge parameter, which we will call $\lambda_{opt}$, that will balance the bias and variance such that the MSE of the ORR solution is less than that of the OLS solution [104] :

$$MSE(\lambda_{opt}) = E\left[\left(\boldsymbol{\beta}_{true} - \hat{\boldsymbol{\beta}}_{\lambda_{opt}}\right)^T\left(\boldsymbol{\beta}_{true} - \hat{\boldsymbol{\beta}}_{\lambda_{opt}}\right)\right] = E\left\|\boldsymbol{\beta}_{true} - \hat{\boldsymbol{\beta}}_{\lambda_{opt}}\right\|^2 < MSE(\lambda_{OLS}) \tag{2.4}$$

This means that even if the data matrix is not badly ill-conditioned, one can improve prediction accuracy by using ORR over OLS. While regularizing the solution will bias the estimate, it will also reduce the variance making the model more stable. Consequently, the probability that the ORR estimates are closer to the true parameter values is greater than that of the OLS estimate.

One encounters three problems when attempting to develop ORR models. The first problem arises from the fact that the MSE is not computable unless the true solution is known. The second is how to choose an optimal ridge parameter and the third is the inherent assumption that the variance of a component is a valid indicator of its predictive relevance.

This first problem can be mitigated by using the *mean predictive error* (MPE) as an estimate of MSE, which will be discussed in section 2.2.1.

The second problem – the optimal selection of ridge parameters – is exemplified by the fact that the performance of ORR is very sensitive to the selection of the ridge parameters. Several methods of choosing optimal ridge parameters have found their way into engineering practice [12,19,157,47,68,82,104,105,120,151,214,224,225]. The most common methods are the *Discrepancy Principle (DP)* [146]; *Mallows' CL* [137], *Generalized Cross Validation (GCV)* [70], and the *L-curve*

*method* [86,87,89,132]. Unfortunately, each method has its own weakness. *CL* and *DP* are highly sensitivity to underestimations of the noise level and consequently their application has been limited to cases in which the noise level can be estimated with high fidelity [86]. On the other hand, noise-estimate-free *GCV* occasionally fails, presumably due to the presence of correlated noise [237]. The *L-curve* method is widely used but is know to be non-convergent [134,233] in some cases. Particularly, it fails to converge to a true solution as $n \to \infty$ [233] or as the error norm approaches zero [54]. All these methods directly or indirectly estimate the MPE and attempt to select the ridge parameter that minimizes the MPE estimate. The difficulty associated with selecting an optimal ridge parameter is very well documented in the literature. For more detailed information on the various parameter choice methods see [47,58].

A third problem arises from the implied assumption that the variance of a component is a good indicator of its predictive relevance. As it is typically implemented, ORR uses the ridge parameter as a threshold to determine which components are relevant and which are irrelevant. It passes only those components whose associated singular values are larger than ridge parameter and completely ignores components with singular values that are less than the ridge parameter. While the singular value of a component is a measure of the variance in the component, a high variance does not necessarily guarantee that a component will have correspondingly high predictive relevance. In fact, it is possible for components to have high variances and large singular values (larger than the ridge parameter), and yet possess little or no predictive information [19,101,102] (as will be demonstrated in section 6.6.1). In these cases ORR would include these irrelevant components in the model, which would increase the variance and degraded the predictive performance of the model. The opposite case would occur when components with low variances and small singular values (smaller than the ridge parameter) are mistakenly ignored.

Clearly a more optimal approach would be to individually select a ridge parameter for each component so that the associated component could be passed or damped based on its predictive relevance rather than its amount of variation. This approach which used a vector of ridge parameters, one for each component, is referred to as *localized ridge regression* (LRR) in this work and will be discussed in section 2.3.

### 2.2.1 Mean Predictive Error as an Estimate of MSE

While the MSE on future samples is not known it can be estimated using the *mean predictive error* to select $\lambda_{opt}$,

$$MPE(\lambda_{opt}) = E\left[\left(\mathbf{X}\boldsymbol{\beta}_{true} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\lambda_{opt}}\right)^T\left(\mathbf{X}\boldsymbol{\beta}_{true} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\lambda_{opt}}\right)\right] = E\left\|\mathbf{X}\boldsymbol{\beta}_{true} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\lambda_{opt}}\right\|^2, \tag{2.5}$$

which can be successfully approximated using available data.

Various methods can and have been used to approximate MPE. One common example, is Mallows' *CL* [137]:

$$CL(\lambda_{opt}) = \frac{1}{n}\left\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\lambda_i}\right\|^2 + \frac{2\hat{\sigma}^2}{n}\text{trace}(H(\lambda)) \quad CL(\lambda) = \frac{\left(y - X\hat{b}_\lambda\right)^T\left(y - X\hat{b}_\lambda\right)}{n} + \frac{2\sigma^2}{n}trace(H(\lambda)) \tag{2.6}$$

where the hat matrix, $H(\lambda)$, is defined as:

$$H(\lambda) = \mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \lambda^2\mathbf{I}_d\right)^{-1}\mathbf{X}^T \tag{2.7}$$

## 2.3 Localized Ridge Regression

The LRR concept was originally introduced under the name of generalized ridge regression (GRR) [104]. ORR uses the same ridge parameter for each input variable, which means in equations (2.1) and (2.2), $\lambda = \lambda_1 = \lambda_2 \ldots = \lambda_d$. But if we slightly modify equation (2.6) we come up with a *localized* version of ridge regression that enables us to apply individual ridge parameters selectively to each component and thus partially damp or pass a component based on its relative contribution to reducing the MPE:

$$\hat{\boldsymbol{\beta}}_{\lambda_i} = \left(\mathbf{X}^T\mathbf{X} + \mathbf{V}\,\text{diag}\left(\lambda_i^2\right)\mathbf{V}^T\right)^{-1}\mathbf{X}^T\mathbf{y} = \sum_{i=1}^{d} f_{i,\lambda_i}\boldsymbol{\rho}_i\mathbf{v}_i \tag{2.8}$$

with the *filter factors* computed as $f_{i,\lambda_i} = \mathbf{s}_i^2/\left(\mathbf{s}_i^2 + \lambda_i^2\right)$, and $\lambda_i$ being the localized ridge parameters. A sufficiently large $\lambda_i$ with respect to its corresponding singular value prevents the corresponding *component* from contributing to the solution while a $\lambda_i$ near zero will allow its corresponding component to contribute significantly to the solution. Values of $\lambda_i$ that are slightly above or below a singular value will proportionately *filter* a component, partially passing it or block it, as defined by the filter factor. The superiority of this approach over OLS, and ORR has been demonstrated [19,101,102] and results will be presented in section 6.6.1.2.

## 2.4 Prior Work on Optimization of Localized Ridge Regression Parameters

Little has been published on the topic of selecting optimal localized ridge regression parameters.

### 2.4.1 Iterative Techniques

Orr described an iterative method for optimizing each parameter. The parameter of the component with the highest correlation is first optimized individually and then the next most highly correlated and so on continuing iteratively until a solution converges [155]. This method is time consuming and subject to becoming trapped in local minima.

### 2.4.2 Gradient-based Techniques

McNames described a technique of gradient-based optimization for vectored ridge regression in his PhD dissertation [141]. The approach optimized the gradient of the cross-validation error with respect to smoothing parameters, model inputs, and ridge coefficients. As indicated in his summary the approach "can get stuck in shallow local minima after only modestly improving the model performance."

Chapter

3

# Localized Kernel Regression

Local regression or *lazy learning* (LL) is a non-parametric, memory-based technique that stores past data exemplars in memory and processes them when a new query is made. So rather than modeling the entire input space with a parametric model such as a linear regression function or neural network or, the local regression function $f(X)$ is estimated over the entire domain $\Re^p$ by fitting a different but simple model at every query point $q$. Only the observations closest to the query point $q$ are used to fit the model, resulting in an estimated function $\hat{f}(X)$ that is smooth in $\Re^p$. These local models are constructed "on the fly" by using a weighting function or kernel $K(d(x_i, q))$, which assigns a weight to $x_i$ based on its distance from $q$. When a query is made, the algorithm locates the training exemplars in its neighborhood and performs a weighted regression with the nearby observations. The observations are weighted with respect to their proximity to the query point. The distance function, $d$, determines the size of the neighborhood around each query point $q$ and is the only parameter that needs to be determined from the training data, while the model includes the entire data set. But, in order to construct a robust local model, one must optimally select an appropriate distance function, $d$, implement locally weighted regression, and consider smoothing techniques such as regularization.

Of the various methods for constructing local models, the most commonly used are: 1) k-nearest neighbor , 2) weighted averaging, 3) kernel regression and 4) locally weighted regression. The primary focus of this research is on kernel regression.

## 3.1 Kernel Regression

Kernel regression (KR) models the input-output relationship of the observed set of data pairs $D = (x_i, y_i)$ as a conditional expectation. The conditional expectation of observing $y$ given $x$ can be expressed as:

$$E(y \mid x) = m(x) \tag{3.1}$$

where $m(\cdot)$ is the regression model. This can be re-written in the more familiar regression form:

$$y = m(x) + \varepsilon \tag{3.2}$$

where $\varepsilon$ is a noise term. Given a set of $n$ independent observations, $(x_1, y_1),\ldots,(x_i, y_i),\ldots(x_n, y_n)$, the Nadaraya-Watson estimator of $\hat{y}$ given a single dimensional *query* point $q$ can be expressed as [15,147,183,239]:

$$\hat{y}(q;h) = \hat{m}(q;h) = \frac{\sum_{i=1}^{n} K_h(d_q) y_i}{\sum_{i=1}^{n} K_h(d_q)} \tag{3.3}$$

where, $d_q$ is some distance measure between the query point $q$ and all data in memory $x_i$, and $K_h$ is a kernel weighting function with a bandwidth of $h$. A common distance measure is the Euclidean distance,

$$d_q = d(x_i, q) = \left( \sum_{i=1}^{n} (x_i - q)^2 \right)^{1/2}. \tag{3.4}$$

A commonly used kernel shape is the Gaussian kernel. Others can be used (see section 3.2.1 for a description of other commonly used kernels). The Gaussian kernel with bandwidth $h$ is defined by:

$$K_h(d) = (2\pi h)^{1/2} * e^{(-d^2/2h^2)} \tag{3.5}$$

We shall now extend equation (3.3) to address the multivariate case.

## 3.2 Localized (Multivariate) Kernel Regression

As Hastie et al. describes in [94 p. 174], the above method can be generalized to the multivariate case of two or more dimensions $\Re^p$. For the multivariate case we are now dealing with a *p*-dimensional kernel. Transformation to the multivariate case leads to much greater design flexibility. An additional *p-1* degrees of freedom is gained along the principle dimensions of the model by providing access to the diagonal elements of the bandwidth matrix, and $p^2$*-1* degrees of freedom are accessible if we choose to make use of the off-diagonal elements of the bandwidth matrix. This added modeling power is very appealing until one

realizes that each of these bandwidth parameters has to be carefully selected to produce a model that will provide robust, reliable predictions. This implementation of multivariate kernel regression, which will be referred to as localized kernel regression (LKR), can be described by the equation:

$$\hat{y}(\mathbf{q};\mathbf{H}) = \hat{m}(\mathbf{q};\mathbf{H}) = \frac{\sum_{i=1}^{n} K(d_{\mathbf{H}}^2(\mathbf{x}_i,\mathbf{q}))y_i}{\sum_{i=1}^{n} K(d_{\mathbf{H}}^2(\mathbf{x}_i,\mathbf{q}))} \tag{3.6}$$

where $\mathbf{x}$ is the $p$-dimensional input vector and $d_{\mathbf{H}}^2(\mathbf{x}_i,\mathbf{q}) = (\mathbf{x}_i - \mathbf{q})^T \mathbf{H}^{-1}(\mathbf{x}_i - \mathbf{q})$ is the squared Euclidian distance parameterized by $\mathbf{H}$ which is a $p \times p$ positive definite bandwidth matrix.

The bandwidth matrix $\mathbf{H}$ can be specified at three different levels of complexity each offering an increase in flexibility. The simplest uses a scalar bandwidth parameter $h$ resulting in spherical kernel:

$$\mathbf{H} \in S \equiv \mathbf{H} = h\mathbf{I} = \begin{bmatrix} h_1 & 0 & \dots & 0 \\ 0 & h_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & h_p \end{bmatrix}, \quad h_1 = h_2 = ,\dots, = h_p = h > 0 \tag{3.7}$$

This form has the advantage of dealing with only a single bandwidth parameter, but it comes with the severe price (more severe in some cases than others) of applying the same degree of smoothing to all input dimensions indiscriminately.

The next level of complexity involves the diagonal elements of the bandwidth matrix:

$$\mathbf{H} \in D, \quad \mathbf{H} = diag(h_1,\dots,h_p) = \begin{bmatrix} h_1 & 0 & \dots & 0 \\ 0 & h_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & h_p \end{bmatrix}, \quad h_1,\dots,h_p \geq 0 \tag{3.8}$$

The greatest level of complexity, which offers the greatest degree of modeling flexibility, is the full bandwidth matrix:

$$\mathbf{H} \in F, \quad \mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,p} \\ h_{2,1} & h_{2,2} & & h_{2,p} \\ \vdots & & \ddots & \vdots \\ h_{p,1} & h_{p,2} & \dots & h_{p,p} \end{bmatrix}, \quad h_{1,1},\dots h_{p,p} \geq 0 \tag{3.9}$$

Using a 2D Gaussian kernel as an example may better explain the affects of using $\mathbf{H} \in S$, $\mathbf{H} \in D$ or $\mathbf{H} \in F$. Surface and contour plots of a 2D Gaussian kernel which has been parameterized with the three different levels of the bandwidth matrix $\mathbf{H}$ are presented in Figure 3.1. From this cursory introduction one can conclude that levels two and three offer the greatest flexibility and potentially the greatest level of accuracy, but both are affected by the *curse-of-dimensionality*, which is the exponential growth of the hyper-volume or potential parameter space as a function of dimensionality [9].

There are two primary decisions involved in implementing a kernel regression model: i) the shape of the kernel and ii) the size of the kernel. First we consider shape and then we will address size.

### 3.2.1 Kernel Shape

Many different kernel shapes have been described in the literature, yet most researchers agree the difference in performance of one shape over another is nearly negligible compared to the optimal selection of the bandwidth [139]. Expressions for six of the more commonly used kernel shapes are presented in Table 3.1.

Figure 3.2 presents one dimensional versions of the Gaussian, Epanechnikov, Biweight, Triweight, Triangular, and Uniform kernels with six different bandwidths. The same six bandwidths are applied to 2D versions of the same kernels and presented in Figure 3.3 through Figure 3.8.

LKR models must be properly regularized in order to generate results that are stable and consistently reproducible. Proper regularization involves three interrelated selections: i) the kernel shape, ii) the kernel bandwidth, $h$, and iii) the number of relevant input variables. Most researchers agree that kernel shape is inconsequential when compared to optimal bandwidth selection. Bandwidth selection and selection of relevant input variables are tightly coupled with the kernel bandwidth effectually determining the relative contribution an input variable has on the output. An input variable with a small bandwidth more greatly influences the output prediction than one with a large bandwidth.

### 3.2.2 Kernel Bandwidth

This leads us to the consideration of how one might automatically select optimal bandwidth parameters for $\mathbf{H}$ that minimize some objective function $f(\mathbf{x})$. The objective function provides some measure of how well a particular set of objective variables meet some set of design criteria. The objective variables in this case are the parameters of the bandwidth matrix which could be of the type $\mathbf{H} \in S$, $\mathbf{H} \in D$ or $\mathbf{H} \in F$ as defined in equations (3.7)-(3.9).

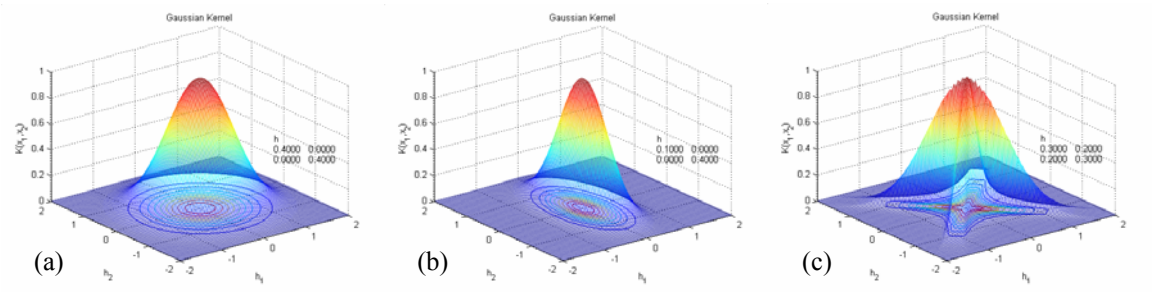



Figure 3.1 Surface and contour plots of 2D Gaussian kernels parameterized by (a) $\mathbf{H} \in S$ ,(b) $\mathbf{H} \in D$ and (c) $\mathbf{H} \in F$ .

Table 3.1 Kernel functions.

| **Kernel** | $K(d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q})$ |
|---|---|
| Gaussian: | $e^{-0.5 d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q})}$ |
| Epanechnikov: | $(1 - d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q})) \vee 1 \; if \; d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q}) < 1$ |
| Biweight: | $(1 - d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q}))^2 \vee 1 \; if \; d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q}) < 1$ |
| Triweight: | $(1 - d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q}))^3 \vee 1 \; if \; d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q}) < 1$ |
| Triangular: | $\left(1 - \sqrt{d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q})}\right) \vee 1 \; if \; d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q}) < 1$ |
| Uniform: | $d_{\mathbf{H}}^2(\mathbf{x}_i, \mathbf{q}) \; if \leq 1 \; else \; 0$ |

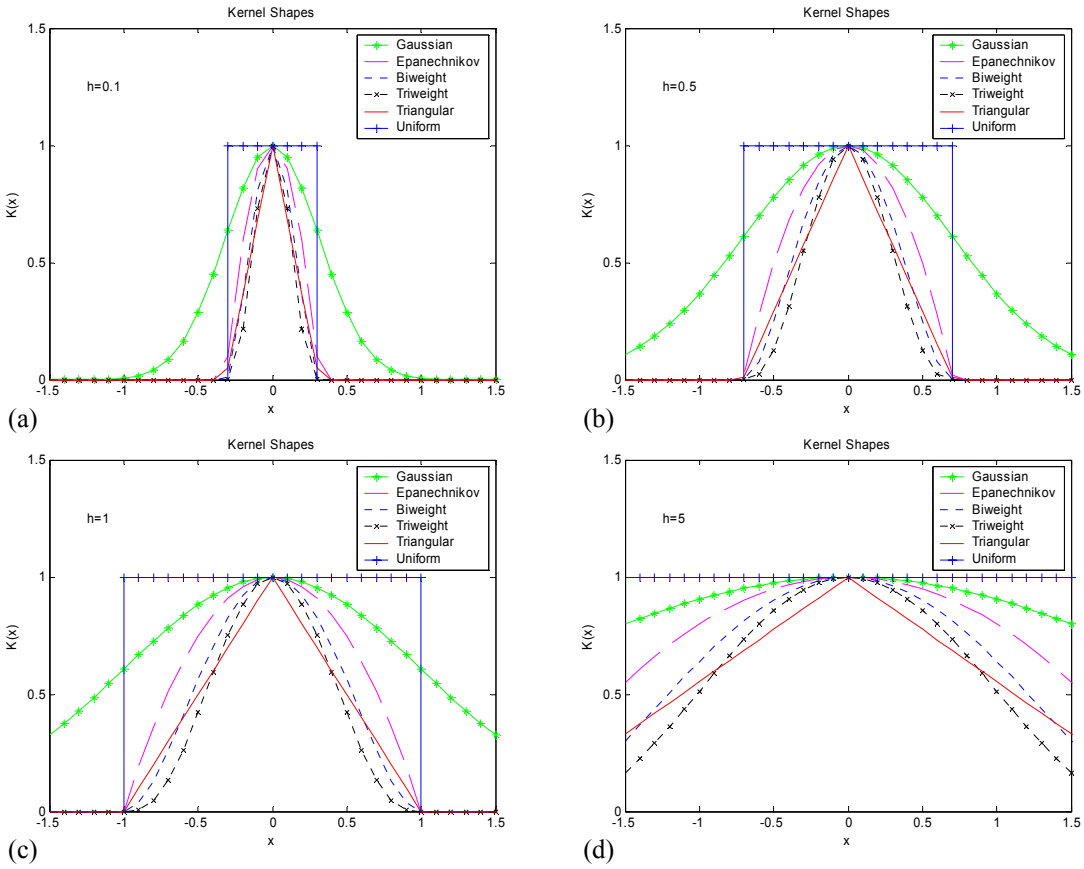



Figure 3.2 Comparison of kernel shapes for 1D Gaussian, Epanechnikov, Biweight, Triweight, Triangular, and Uniform kernels with different bandwidths: (a) h=0.1, (b) h=0.5, (c) h=1, and (d) h=5.

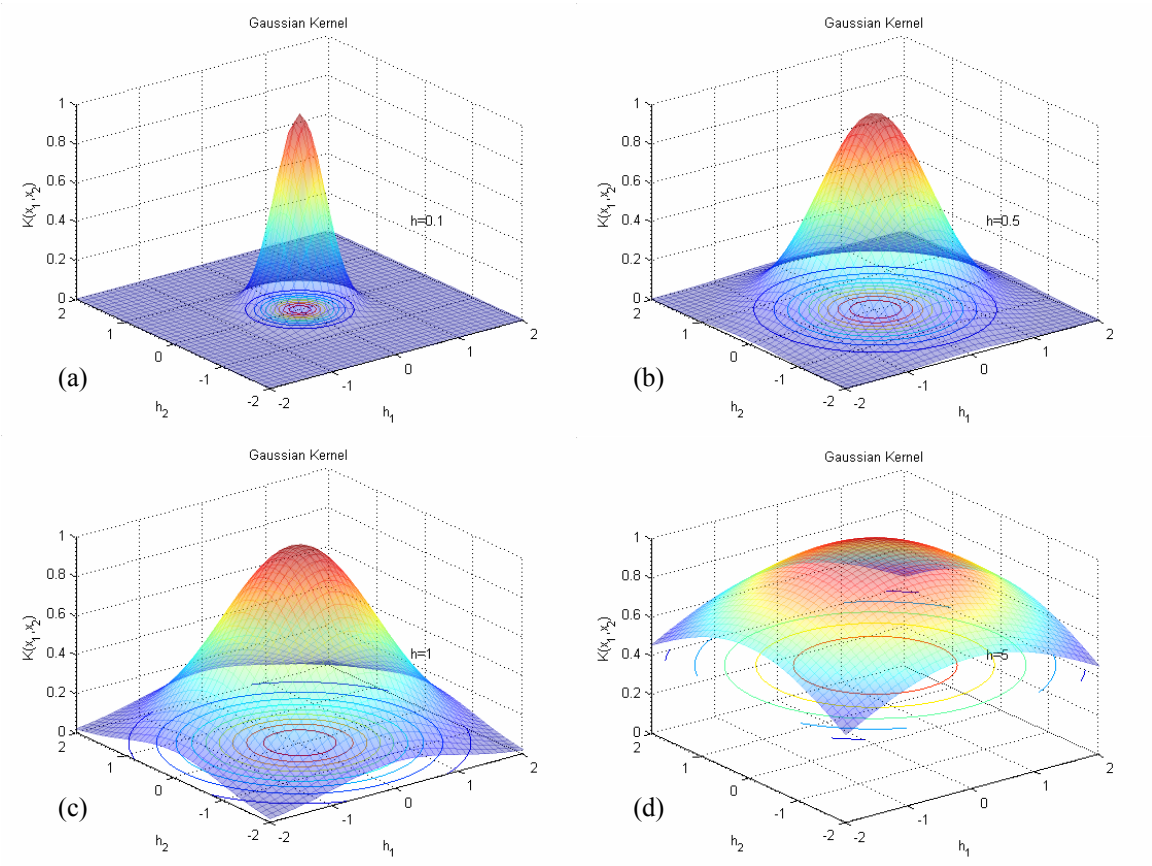



Figure 3.3 Comparison of different bandwidths for 2D Gaussian kernel: (a) $h$=0.1, (b) $h$=0.5, (c) $h$=1, and (d) $h$=5.

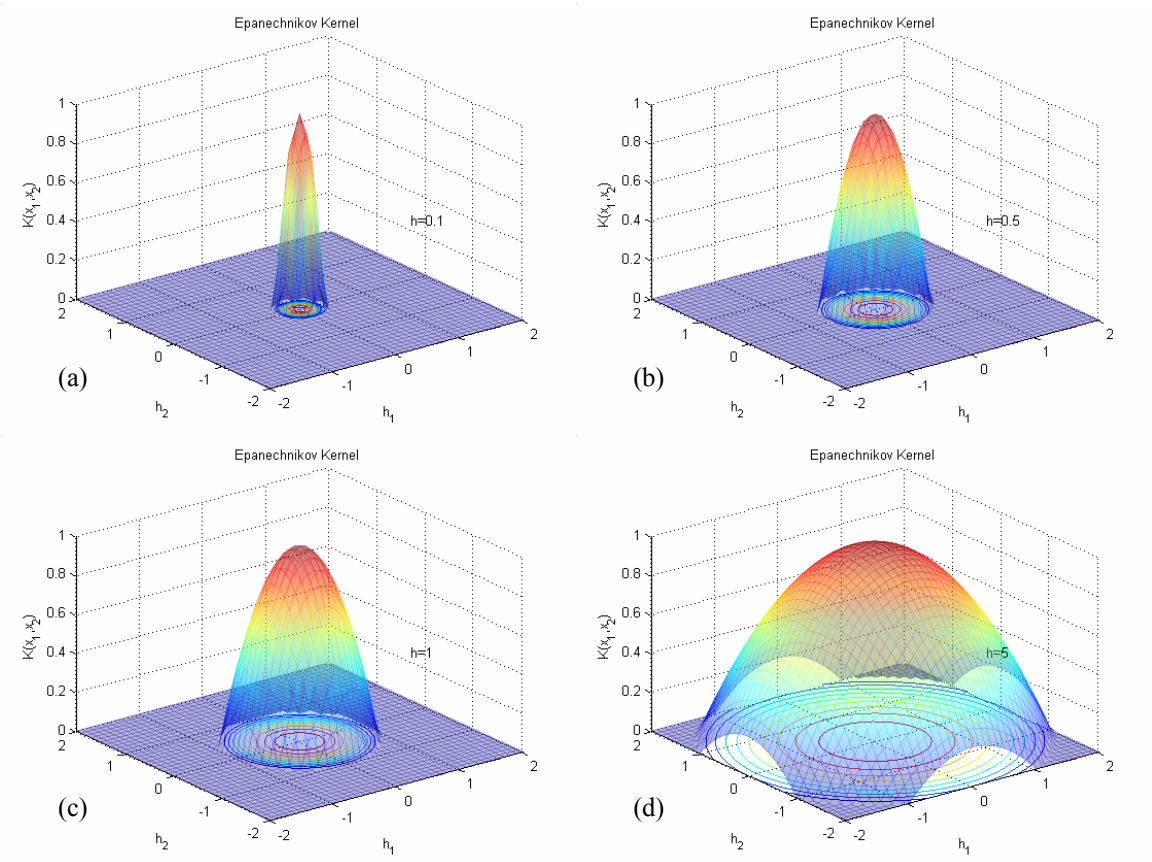



Figure 3.4 Comparison of different bandwidths for 2D Epanechnikov kernel: (a) $h$=0.1, (b) $h$=0.5, (c) $h$=1, and (d) $h$=5.

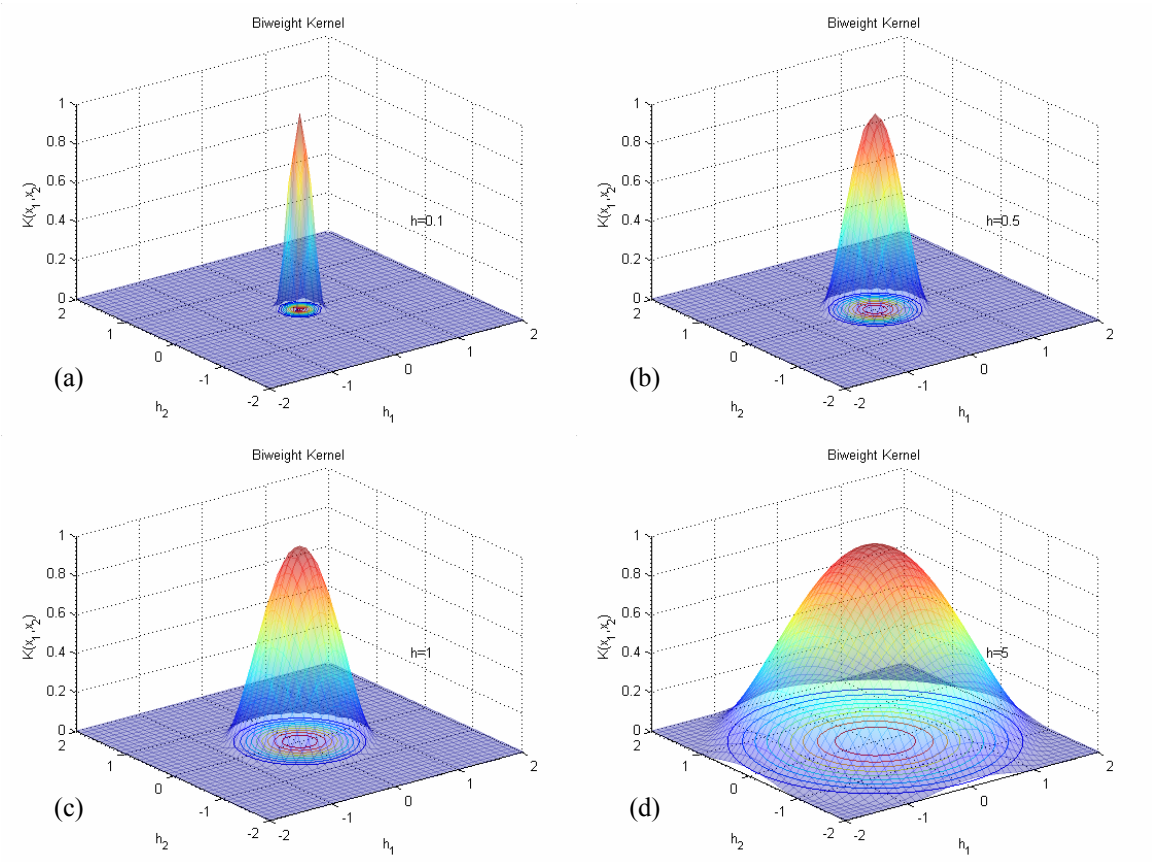



Figure 3.5 Comparison of different bandwidths for 2D Biweight kernel: (a) $h$=0.1, (b) $h$=0.5, (c) $h$=1, and (d) $h$=5.

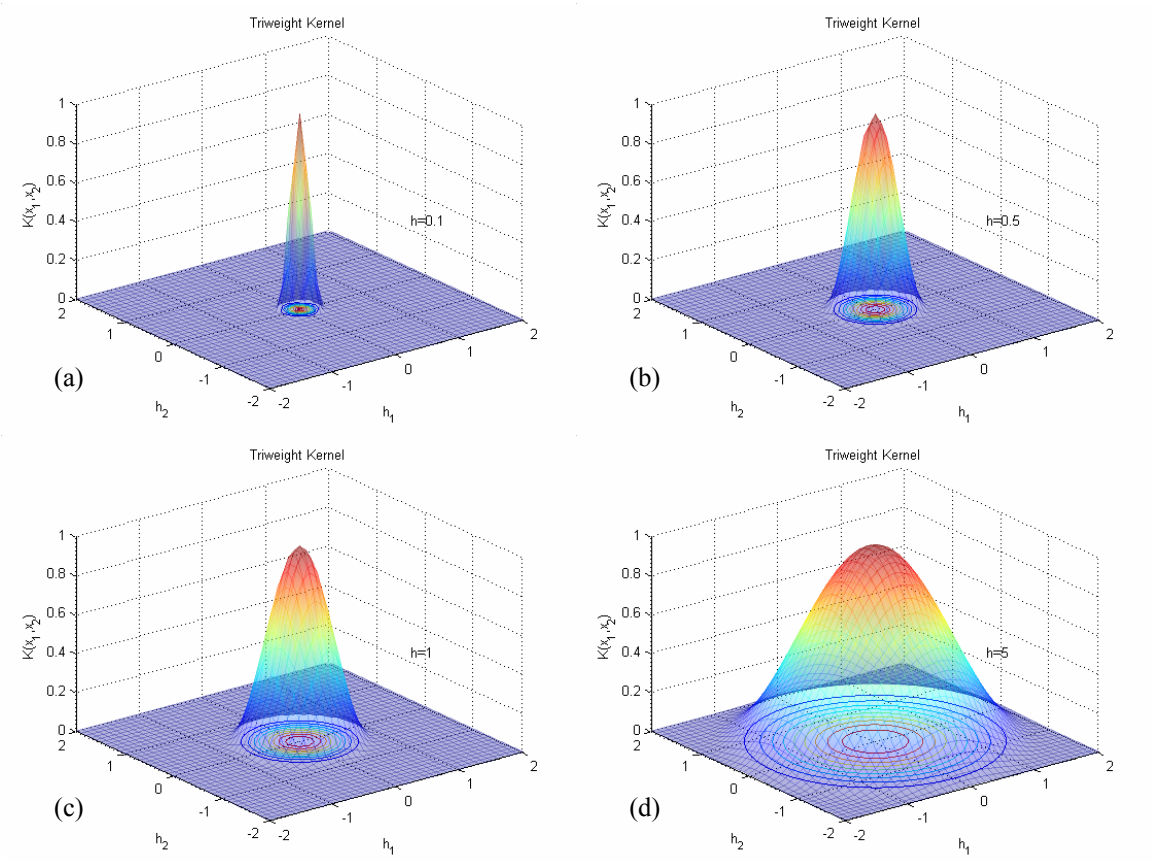



Figure 3.6 Comparison of different bandwidths for 2D Triweight kernel: (a) $h$=0.1, (b) $h$=0.5, (c) $h$=1, and (d) $h$=5.

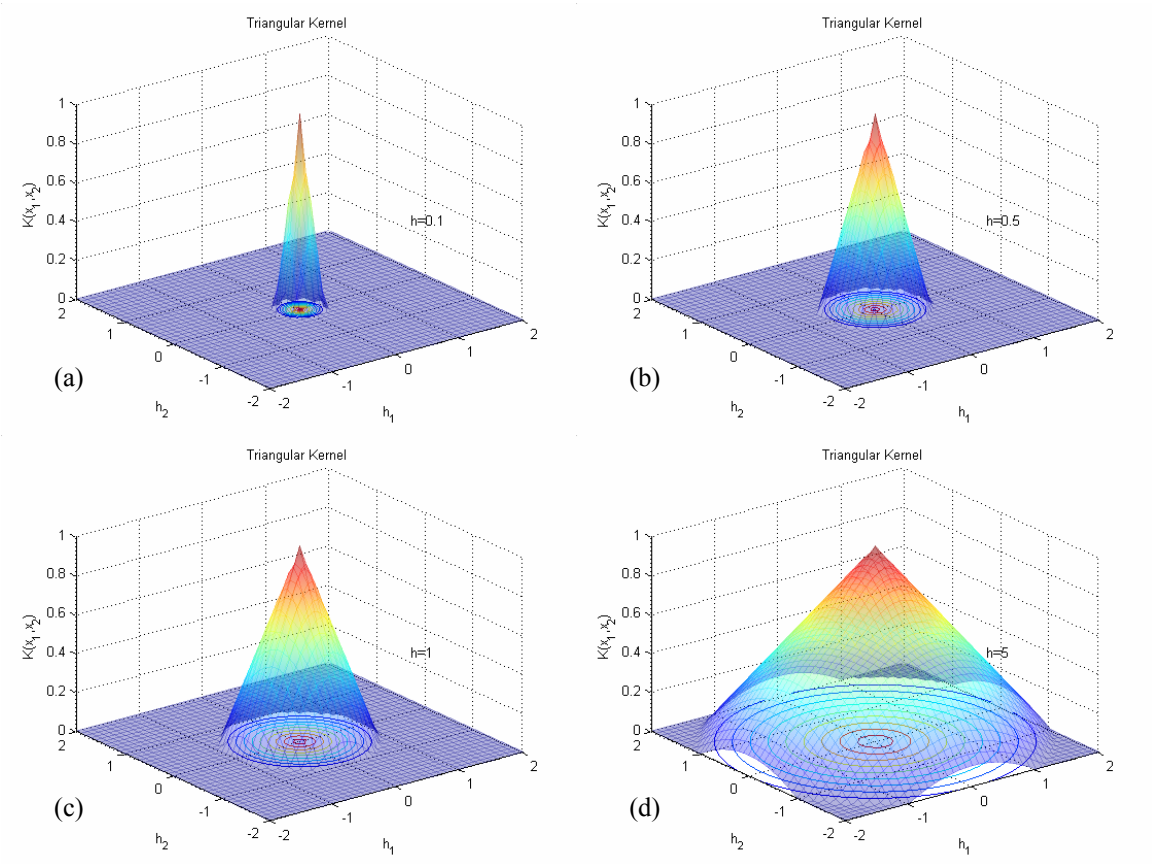



Figure 3.7 Comparison of different bandwidths for 2D Triangular kernel: (a) $h$=0.1, (b) $h$=0.5, (c) $h$=1, and (d) $h$=5.

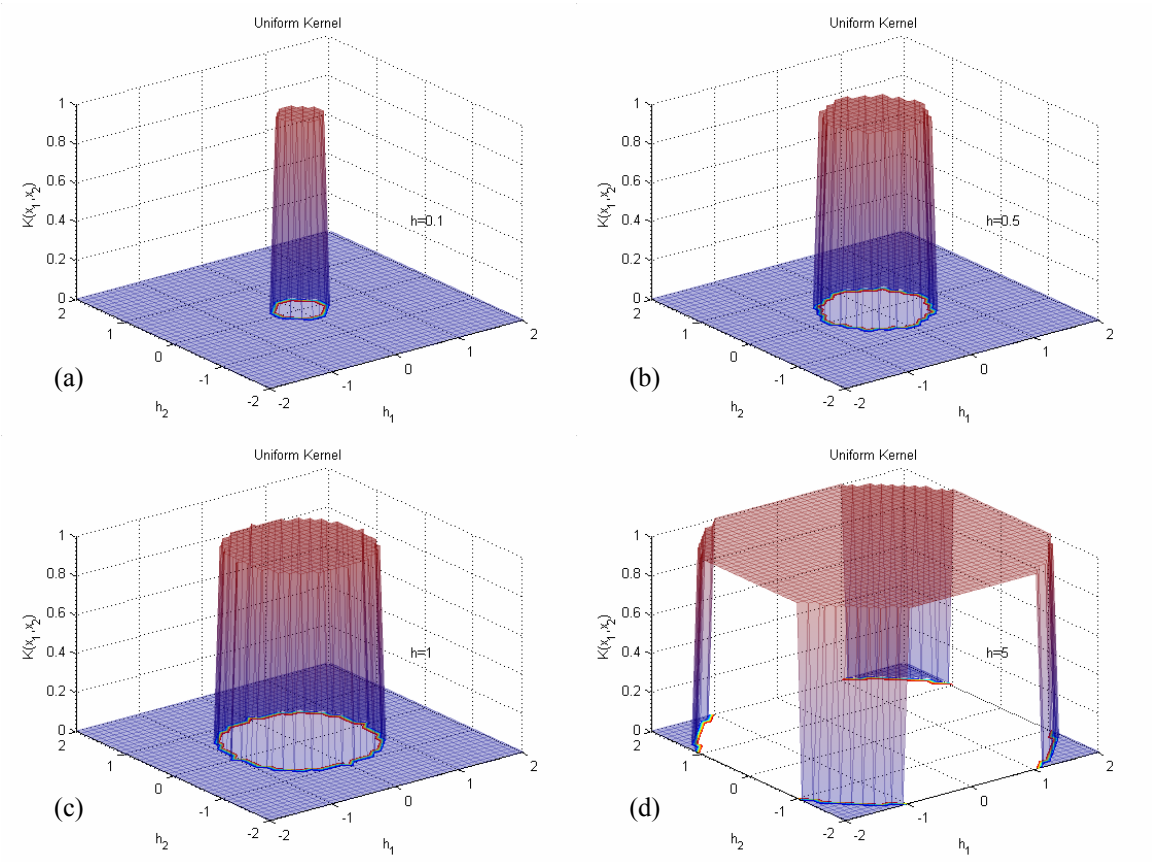



Figure 3.8 Comparison of different bandwidths for 2D Uniform kernel: (a) $h$=0.1, (b) $h$=0.5, (c) $h$=1, and (d) $h$=5.

The most common implementation of KR makes use of a single "global" bandwidth parameter which is selected by minimizing the cross-validation error, which is an estimate of the generalization error. This indiscriminate *one-size-fits-all* approach, results in sub-optimal performance as will be demonstrated by the following example.

This example makes use of the RampHill data set described in [131,141] which has 2-inputs and 1-output. Performance on this function can provide a good indication of the robustness of non-linear modeling methods because it contains several features that are often found in real world processes [164]. It contains two regions with different *constant* outputs, one *linear* region designated *ramp* and one localized *nonlinear* region designated *hill*. The training data contains 100 input-output pairs and was designated as RampHill1_100. The test data contains 5,041 data pairs. A surface plot of the test data along with the 100 training points is presented in Figure 3.9 (a). Figure 3.9 (b) presents a contour plot of RampHill1_100 with the regions "hill", "ramp" and "constant" labeled accordingly. The RampHill1_100 dataset was modified to demonstrate the impact of irrelevant inputs when a global bandwidth is used. The RampHill1u10_100 dataset was generated by adding eight irrelevant input vectors making the original input training data matrix [100 x 10]. The irrelevant inputs were randomly generated from a uniform random distribution U(0,1).

The leave-one-out cross-validation error (LOO-CVE) surface for RampHill1_100 is presented in Figure 3.10 (a). LOO-CVE is described in section 4.4.1. For now, notice that there is a global minimum LOO-CVE of 0.0227 at $h_1,h_2$ = (0.01, 0.0087). The Gaussian kernel produced by these bandwidths is shown in Figure 3.10 (b). Selection of the global bandwidth is made by finding a single value of $h$ that minimizes the LOO-CVE. Figure 3.11 (a) shows the global LOO-CVE curve for which the optimal $h$ (0.9438) is at the minimum LOO-CVE = 0.3887. The resulting function approximation of RampHill1u10_100 using the global bandwidth parameter is plotted in Figure 3.11 (b) with a training MSE = 0.0401 and test MSE = 0.4504. Notice how the noise of the irrelevant inputs has affected the output prediction, which should be smooth. A good model of the RampHill1u10_100 data set should have bandwidths close to $h_1,h_2$ = (0.01, 0.0087) for inputs 1-2 but very broad values for inputs 3-10.

Goutte and Larsen introduced a method of selecting individual bandwidth parameters for LKR called Adaptive Metric Kernel Regression (AMKREG) in [72,73]. AMKREG uses a combination of conjugate-gradient and approximate line search to select parameters of the diagonal smoothing matrix $\mathbf{H} = diag(h_1^2, h_2^2, \ldots h_p^2)$ by minimizing the LOO-CVE. A smoothing parameter $h_p$ is the reciprocal of the bandwidth parameter $h$. AMKREG is fast and accurate as long as the LOO-CVE surface is smooth and it does not contain local minima. AMKREG and all other classical gradient descent based optimization techniques are highly susceptible to getting stuck in a local minima. Consequently, their accuracy and

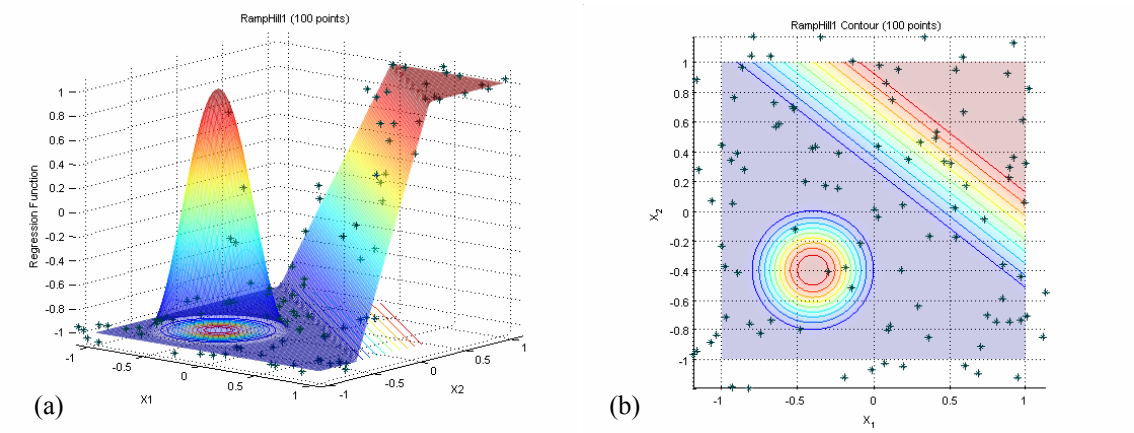



Figure 3.9 Ramphill1_100 data set: (a) surface plot represents the 5,041 test points and asterisks the 100 training points, (b) contour plot with labeled on designated regions.

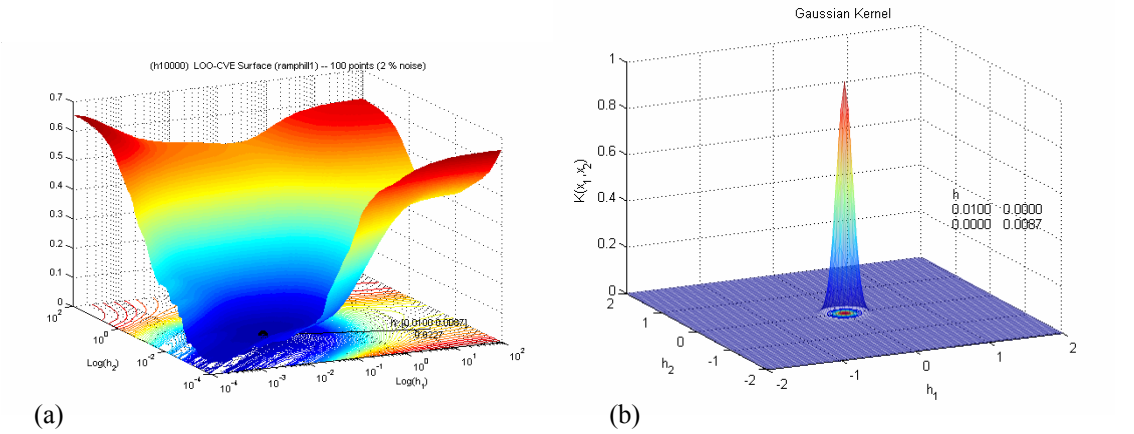
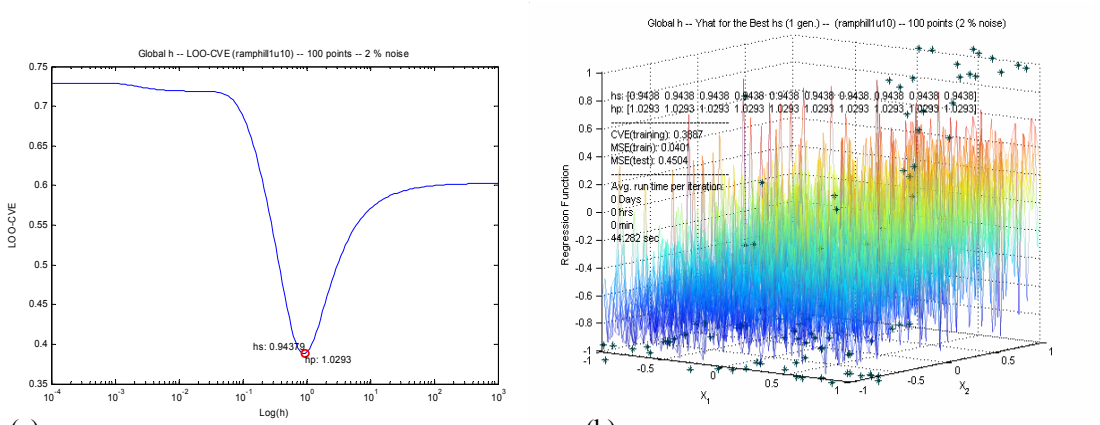



Figure 3.10 RampHill1u10_100 data set: (a) LOO-CVE surface. (b) Shape of the Gaussian kernel using the optimal bandwidth $h_1,h_2 = (0.1,0.0087)$ corresponding to the LOO-CV global minimum.




(a) (b)

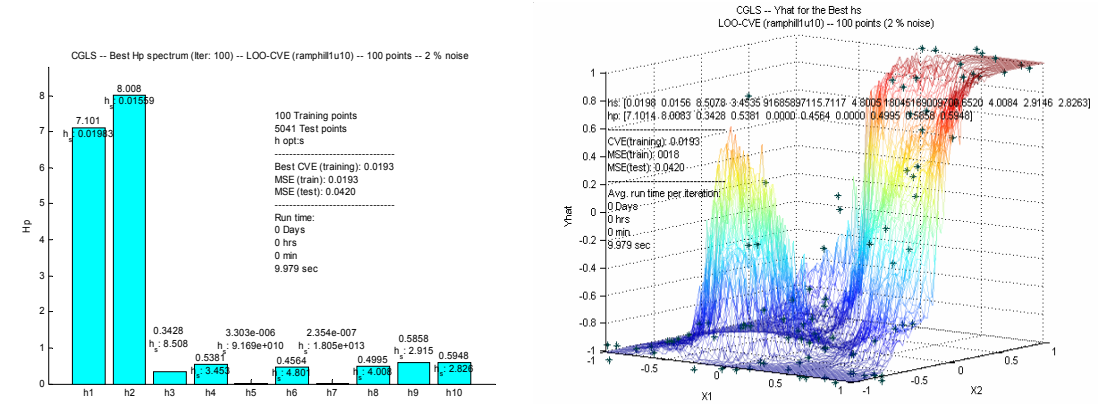
Figure 3.11 Global bandwidth results for RampHill1u10_100: (a) LOO-CVE plot. (b) Function approximation obtained using the global bandwidth parameter (0.9438).

reliability depend heavily on the "appropriateness" of the initial starting point and a cost function that is relatively free of local minima.

In the example that follows, a variant of AMKREG, which will be referred to as CGLS for conjugate gradient/line-search was used to select bandwidth parameters for RampHill1u10_100. The resulting function approximation is presented in Figure 3.12 (b). Figure 3.12 (a) is the smoothing parameter spectrum of the best bandwidth parameters found by CGLS. A smoothing parameter is the reciprocal of the bandwidth $h$ so a large smoothing parameter corresponds to a small bandwidth and signifies an important or relevant input variable. A small smoothing parameter corresponds to a large bandwidth and signifies an irrelevant input variable. Notice that the first two smoothing parameters are the largest indicating the relevance of their associated input variables while parameters 3-10 are much smaller indicating their associated variables are less significant. The right panel contains the function approximation (mesh plot) of RampHill1u10_100 using the bandwidth parameters obtained by CGLS.

The CGLS approach is able to select bandwidth parameters that accurately suppress irrelevant parameters primarily because the underlying LOO-CVE surface did not contain local minima, but the results presented in section 6.5 demonstrate the devastating impact local minima has on the ability of CGLS to accurately select relevant input variables. The judicious application of a multi-dimensional optimization approach that is not susceptible to getting stuck in local minima could produce significant improvements in performance over traditional global bandwidth methods and CGLS based LKR techniques.

Clearly a robust yet simple approach for multi-parameter optimization is needed to address both the LRR problem and LKR problems.




(a) (b)

Figure 3.12 CGLS results for RampHillu10_100: (a) Smoothing parameter spectrum of the best bandwidth parameters obtained by CGLS. (b) Function approximation of using the best smoothing parameters.

## 3.3 Prior Work on Selection of Bandwidth Parameters for Localized Kernel Regression

The automated selection of bandwidth parameters can be grouped into three loose categories: i) *quick and simple*, ii) *classical* and iii) *plug-in* [136,239]. A *quick and simple* approach aims at finding an easily computable bandwidth that provides a "reasonable" model for a wide range of situations but without any mathematical assurance of optimality. *Classical* approaches are extensions of methods used in parametric modeling. They typically are more mathematically rigorous, requiring more computational resources, with the aim of asymptotically approaching some minimum measure of empirical risk often expressed as mean integrated squared error (MISE), $\hat{f}(\cdot;h)$. Examples of c*lassical* methods [136] are *least squares cross-validation* (LSCV), *biased cross-validation* (BCV), Mallow's $C_p$, and Akaike's Information Criterion (AIC).

For the most part the methods alluded to above address the selection of bandwidth parameters for the univariate case, or if they address the multivariate kernel, it is for the use of single global bandwidth parameters as in the $\mathbf{H} \in S$ case. And, to the authors knowledge, a method of selecting parameters for the $\mathbf{H} \in F$ case has not been directly addressed in the literature. The examples for the $\mathbf{H} \in D$ have evolved from or been extensions to kernel density estimation. Arguably one of the best techniques of selecting bandwidth parameters for the $\mathbf{H} \in D$ case is AMKREG [72,73].

### 3.3.1 Adaptive Metric Kernel Regression

AMKREG was introduced by Goutte and Larsen in [72,73]. Using a combination of conjugate-gradient and approximate line search (CGLS) it selects smoothing parameters $\mathbf{H} = diag(h_1^2, h_2^2, \ldots h_p^2)$ that minimize the LOO-CVE. The equations describing the conjugate-gradient (CG) method are described by Goutte in [72] and the line-search (LS) method is described by Rasmussen in [177]. These CG and LS were implemented in the AMKREG code developed by Goutte [75]. The AMKREG MATLAB code was obtained from a URL provided by Goutte [75] and modified to enable direct comparison with the methods described in this research.

AMKREG is fast due to it use of conjugate-gradient and simple line search. It optimizes bandwidth parameters by minimizing the LOO-CVE. Based on the authors experience AMKREG is fast and generally robust. Its robustness appears to come from basing its initial "guess" on the distribution of the underlying data – it uses $n \cdot std(x)$ where $0.2 \le n \le 1$ (1 was used for all results reported in this work) – rather than relying on a random "guess". In section 5.1.1.3 it will be shown that while this approach to initialization may be good in many cases, there are times when it will guarantee convergence to a sub-optimal *local* minimum rather than the optimal *global* minimum. This is especially true for higher dimensional data sets and when the data are sparse, which is a common scenario for many real-work problems.

Chapter 4

# Optimal Parameter Selection by Differential Evolution

This chapter provides an introduction to differential evolution and describes how it is used to address the LRR and LKR problems.

## 4.1 Introduction

Differential Evolution (DE) [170,171,192,193,194,197,198] is a population-based, direct-search algorithm for global optimization. While originally designed to operate on continuous floating point variables, DE has been extended to optimize a mixture of integer, discrete, and continuous variables and is able to employee multiple linear and non-linear constraints [129]. A living bibliography of DE and its applications is maintained by Lampinen and is available on-line [130].

DE has been successfully applied to a variety of real-world optimization problems. Good summaries are provided in [6,130,171]. Fischer et al. [57] applied DE to the selection of neural network parameters and in a benchmark comparison with the more traditional backpropogation using gradient descent, found DE to be a superior method.

DE is exceptionally simple (less than 30 lines of C-code) and has proven to be robust for a variety of real-world optimization problems [171]. While DE is similar to other population based search algorithms, like ES and GAs, it differs in both its self-referential mutation scheme and its selection process. First an overview of the basic mechanics and operation of DE will be provided and then a description of how DE was used to address the LRR problem and LKR problems.

## 4.2 DE Mechanics

Both the LRR and LKR problems can described as trying to answer the question: 'Which parameter values produce a model that best represents the input-output relationship in observed data $(x_1, y_1)\ldots(x_n, y_n)$?'. In order to answer this question we must have an *objective function* $f(\mathbf{x})$, defined over the space of all *objective variables that* can provide a measure of how well a set of objective variables meets the design criteria. The *objective variables* represent those parameters of a system that can be modified, which in the immediate context are the LRR parameters and the LKR bandwidths. The objective function is sometimes referred to as a *cost* function or *fitness* function. Our goal is to find the optimal values of $\mathbf{x}$ that provide a minimum value of $f(\mathbf{x})$. If the objective function is a maximization function it can be transformed into a minimization function by multiplying $f(\mathbf{x})$ by –1.

DE operates on a population $\mathbf{P}$ of candidate vectors $\mathbf{x}$. The size of the population, *NP*, is held constant for all generations. Each member of the population is denoted as $\mathbf{x}_{i,G}$ where *i* indexes the population and *G* is the particular generation. A population of *NP, D*-dimensional members is written as:

$$\begin{aligned} &\mathbf{P}_G = (\mathbf{x}_{1,G},\ldots,\mathbf{x}_{i,G},\ldots,\mathbf{x}_{NP,G}), \\ &\textit{where} \\ &\mathbf{x}_{i,G} = (x_{1,1,G},\ldots,x_{j,1,G},\ldots x_{D,1,G}),\ i = 1,2,\ldots,NP,\ \ G = 1,2,\ldots,G_{\max} \end{aligned} \tag{4.1}$$

The basic structure of a DE algorithm is as follows:

**Algorithm 4.1: Structure of basic a Basic DE Algorithm**

Set value to reach (VTR), maximum generations ($G_{max}$), size of population (NP)

G=0

Initialize population $\mathbf{P}_G = (\mathbf{x}_{1,G},\ldots,\mathbf{x}_{i,G},\ldots,\mathbf{x}_{NP,G})$ =initPop(NP,D,initOpt)

Evaluate cost( $\mathbf{P}_G$ ) = $\{f(\mathbf{x}_{1,G}),\ldots,f(\mathbf{x}_{i,G}),\ldots,f(\mathbf{x}_{NP,G})\}$

while cost( $\mathbf{P}_G$ ) < VTR and G<Gmax

$\mathbf{P}'_G$ = mutate( $\mathbf{P}_G$ ) $\mathbf{P}_G$

$\mathbf{P}''_G$ = recombine( $\mathbf{P}'_G$ )

cost( $\mathbf{P}''_G$ ) = evaluate( $\mathbf{P}''_G$ )= $\{f(\mathbf{x}''_{1,G}),\ldots,f(\mathbf{x}''_{i,G}),\ldots,f(\mathbf{x}''_{NP,G})\}$

$\mathbf{P}_{G+1}$ =select( $\mathbf{P}_G$ , $\mathbf{P}''_G$ )

G=G+1

end while

### 4.2.1 Initialization

For most real-world engineering problems the parameters of the objective function will be constrained by lower and upper boundary conditions $x_{j,i}^{(L)}$ and $x_{j,i}^{(U)}$ where $j=1,\ldots,D$ of member $i$. There are numerous strategies for generating the initial population. Two common strategies are (ii) sample from a uniform random distribution (rand), (ii) sample from a log distribution (logspace). It is also possible to use domain knowledge about the problem to initialize the population.

The scheme used to initialize the LKR populations was based on a random permutation and the (logspace) initialization. This approach affectively increased the number of candidates with small values and was found to be affective at creating populations that were not susceptible to premature convergence.

Next a new population is created using a self-referential mutation and recombination scheme which is described in the next section.

### 4.2.2 Mutation and Recombination

Mutation for most EAs is driven by a predefined probability distribution function. DE utilizes a self-referential mutation scheme which uses the differences of randomly sampled objective vectors from the current population. The distribution of the objective vector differences is determined by the distribution of the population itself which reflects the contour of the underlying error surface of the objective function being optimized.

DE uses mutation and recombination to produce a single trial vector for each for each objective vector in the population of the current generation. A trial vector $\mathbf{u}_{i,G+1} = \mathbf{v}_{i,G+1}$ is created according the following algorithm:

**Algorithm 4.3: Creation of a trial vector**

1. Input values for CR $\in$ [0,1), and F $\in$ (0,1+]
2. Randomly select indices of 3 objective vectors from the current population,
   **r1,r2,r3 $\in$ (1,2,…,NP)**
3. Create NP, (1 x D) vector of uniform random values to be compared against CR

```
for i=1:NP
    j_rand (i) = rand* ones(1,D)
    for j=1:D
        if (rand_j(i) < CR or if j = j_rand
```

$$u_{j,i,G+1} = v_{j,i,G+1} = x_{j,r3,G} + \mathrm{F}(x_{j,r1,G} - x_{j,r2,G})$$

```
            else
                u_{j,i,G+1} = x_{j,i,G}
            end
        end
    end
```

In this algorithm the random vectors $\mathbf{x}_{r1,G}, \mathbf{x}_{r2,G}, \mathbf{x}_{r3,G}$ differ from one another the target or *"parent"* vector $\mathbf{x}_{i,G}$. When $rand_j[0,1) < CR$ or if $j = j_{rand}$, the $j^{th}$ parameter of the $i^{th}$ *"child"* vector is a linear combination of the $j^{th}$ parameter for three randomly chosen vectors, otherwise the $j^{th}$ parameter of the *"parent"* is passed along to the *"child"* vector. The "or $j = j_{rand}$" condition is used to ensure that at least one of the parameters of the *"child"* vector differs from its *"parent"*. A trial vector is created for every *"parent"* vector in the current population. CR and F are user specified control variables. CR is the crossover factor controlling whether a particular parameter will be taken from the *"parent"* vector $x_{j,i,G}$ or from the mutated vector $v_{j,i,G+1}$. The values of NP, CR and F affect the convergence properties and robustness of DE and often depend on the characteristics of the objective function. Guidelines for selecting the parameters are provided in [128,171,192,193,195,198,199,]. Successful selection of the parameters can usually be obtained after a few trial iterations using differing values.

#### 4.2.2.1 *Enforcing Boundary Constraints at Runtime*

In many cases the boundary constraints can be ignored after the initial population is produced allowing DE to search the parameter space beyond the initialization bounds. Several methods have been suggested to enforce the boundary constraints when necessary. The *hard boundaries* simple truncate or round the offending parameter, while *soft boundaries* use some type of an asymptotic rule. The soft boundary rule implemented in this work is described in the algorithm below.

**Algorithm 4.4: Enforce Soft Boundaries**

Check the trial vector against the of upper $\mathbf{x}^{(hi)} = x_1^{(hi)}, \ldots x_j^{(hi)}, \ldots, x_D^{(hi)}$ and lower

$\mathbf{x}^{(low)} = x_1^{(low)}, \ldots x_j^{(low)}, \ldots, x_D^{(low)}$ boundary constraint vectors and make adjustments if necessary

```
for i = 1:NP
    for j=1:D
        if  u_{j,i,G+1}^{(low)} < x_j^{(low)}
            u_{j,i,G+1} = (x_{j,i,G} + x_j^{(low)})/2
        elseif   u_{j,i,G+1} > x_j^{(hi)}
```

```
            u_{j,i,G+1} = (x_{j,i,G} + x_j^{(hi)})/2
        else
            u_{j,i,G+1} = u_{j,i,G+1}
        end
    end
end
```

Boundaries were applied in two different ways in this work. First, boundaries were used to initialize a population if there was a reason to think that a solution was more likely to be found in a particular range of values or if it is highly unlikely to be found within a range of values. Second, hard boundaries were used to ensure that both the LRR parameters and LKR bandwidth parameters were possible. This had the affect of initially focusing the search but then allowing the population to "wander" wherever the error surface might take it. The use of boundaries has the added benefit of preventing bad objective vectors from being evaluated. This is especially important when long execution times are required for objective functions such as when LOO-CVE is used on large data sets.

### 4.2.3 Selection

The selection scheme utilized by DE is also different from other ES and GAs. Each successive population, $P_{G+1}$, is selected from either the current *"parent"* population, PG, or the *"child"* population, $P'_{G+1} = U_{i,G+1}$, as follows:

```
for (i = 1; i ≤ NP; i = i + 1)
{
      if f(U_{i,G+1}) ≤ f(X_{i,G})
            X_{i,G+1} = U_{i,G+1} = u_{j,i,G+1}
      else
            X_{i,G+1} = X_{i,G} = x_{j,i,G}
}
```

As shown by the above pseudocode each individual "child" in the trail population is compared with a single "parent" in the current population and the individual with the lower objective function "survives" and passes on into the next generation. This means that all the individuals in each successive generation are at least as good as their "parent" in the current generation. In contrast to other EAs, which compare a candidate individual to all other individuals in the population, DE only compares the candidate individual to a single member of the current population.

### 4.2.4 Recombination Strategies

There are at least 10 variants of DE, which differ primarily in the recombination or crossover strategies employed [5,171,172]. A shorthand notation scheme was developed to unambiguously describe the strategies, DE/*p*/*n*/*c*, where *p* designates the "parent" vector to be perturbed, *n* is the number of random difference vectors used to perturb *p,* and *c* designates the type of crossover used, either exponential (exp) or binomial (bin). In binomial crossover a random value $rand_j[0,1)$ is generated for each component in the vector . If $rand_j[0,1) < CR$ the *child's* vector parameter $u_{j,i,G+1}$ is inherited from $v_{j,i,G+1}$ otherwise it comes from $x_{j,i,G+1}$ . In exponential crossover parameters are inherited from $x_{j,i,G+1}$ until the first $rand_j[0,1) < CR$ is encounter, all remaining parameters are inherited from $u_{j,i,G+1}$ .

The strategies most commonly encountered are:

1. DE/best/1
2. DE/rand/1
3. DE/rand-to-best/1
4. DE/best/2
5. DE/rand/2
6. DE/current-to-rand/1
7. DE/current-to-rand/1/bin

One of the objectives of this research was to identify a robust DE strategy not necessarily the fastest one. All the results reported on here were obtained using only two strategies: i) DE/rand-to-best/1/bin designated as strategy 3, and ii) DE/current-to-rand/1/bin designated as strategy 7. Strategy 7 was used exclusively for selection of LRR parameters and parameters. Strategy 3 was the primary strategy used for selecting bandwidth parameters LKR, with occasional use of strategy 7.

Figure 4.1 provides a diagram illustrating strategy 3 (DE/rand-to-best/1) and strategy 7 (DE/current-to-rand/1/bin) is illustrated in Figure 4.2.

The two types of objective functions used for his research are described in the next section. The first are estimators of prediction risk used were used for LRR. They include Mallows' CL, Information Complexity (ICOMP) and Generalized Cross-Validation (GCV). The second type leave-one-out cross-validation (LOO-CV) provides an estimate of generalization error. LOO-CV was used for LKR.

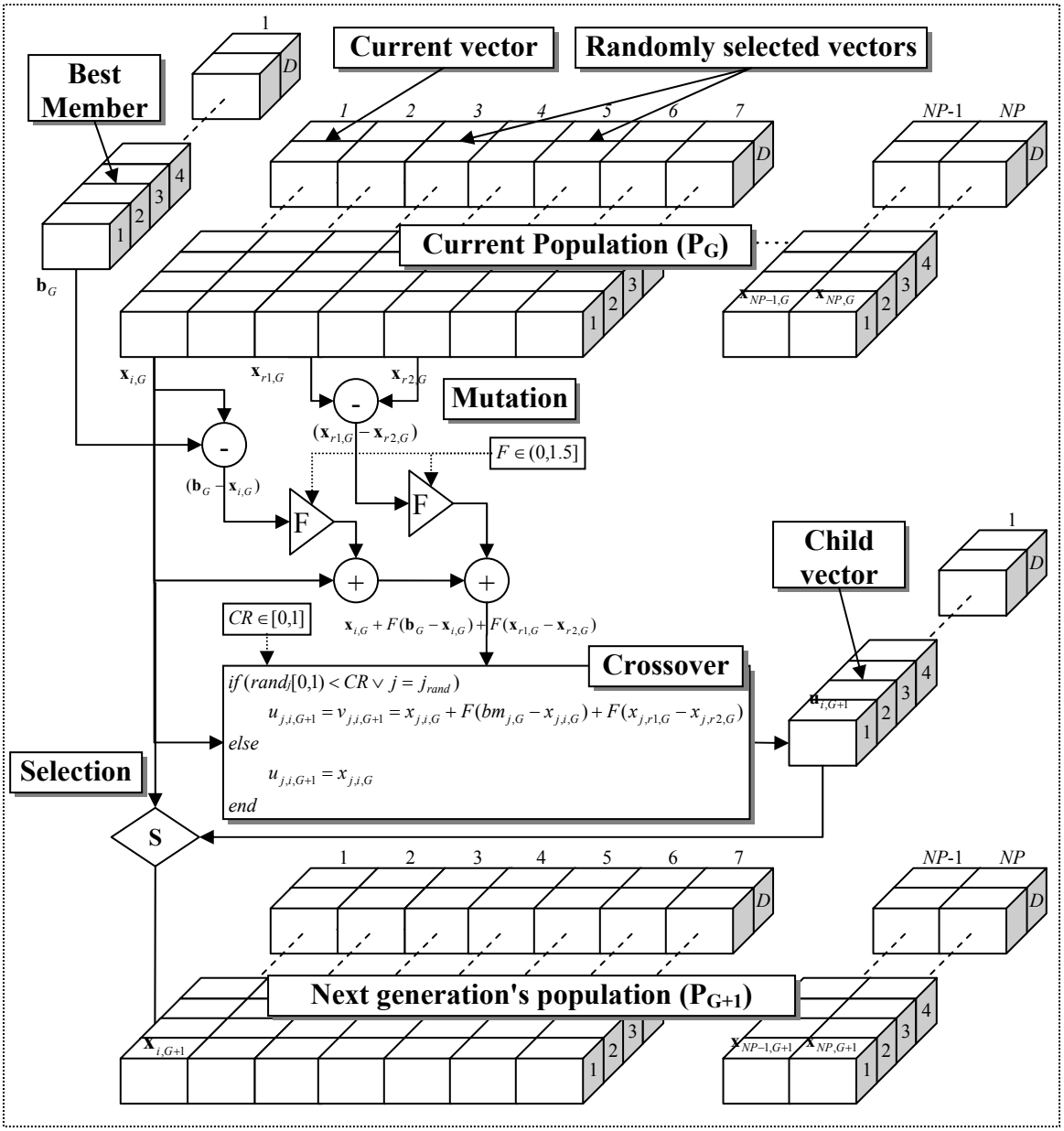



Figure 4.1 Representation of DE/rand-to-best/1/bin strategy.

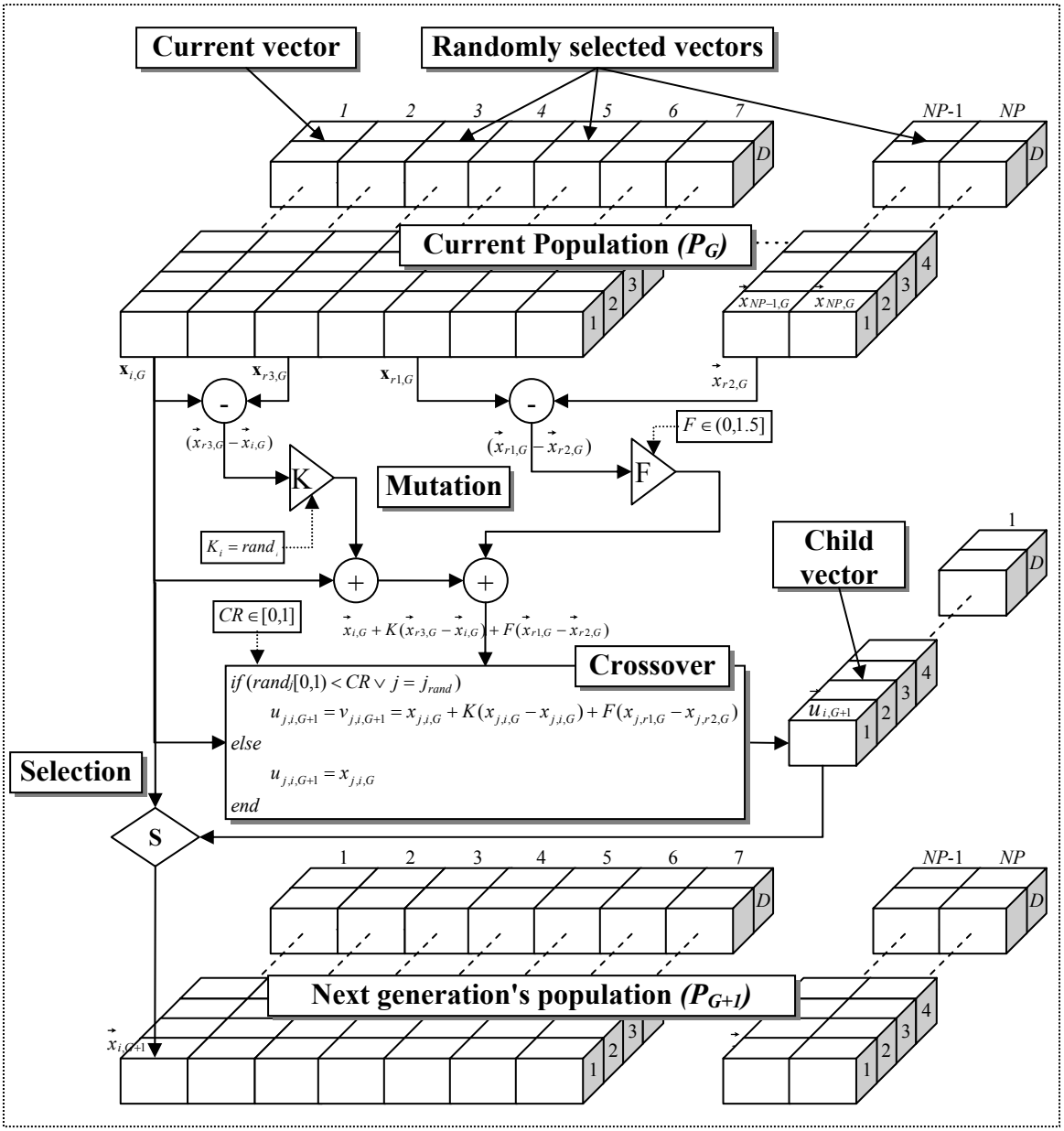



Figure 4.2 Representation of DE/current-to-rand/1/bin strategy.

## 4.3 DE Enabled Automatic Selection of Optimal Localized Ridge Regression Parameters (DEALRR)

Now that we have a general idea of how DE operates, let us take a look at how it can be adapted to automatically select optimal LRR parameters including the methods of population initialization, the strategy and associated control parameters and the objective functions that were minimized. The approach to be described will be referred to as DEALRR for **d**ifferential **e**volution enabled **a**utomatic selection of optimal **l**ocalized **r**idge **r**egression parameters.

### 4.3.1 Mean Predictive Error Estimators as Objective functions for DEALRR

Recall from section 2.2.1 that the MSE, which is seldom if ever known, can be approximated by a MPE estimation derived from observed data. Three MPE estimators were adapted for use as objective functions for selecting LRR parameters: i) Mallows' *CL*, ii) Information Complexity (ICOMP), and iii) Generalized Cross-Validation (GCV).

#### 4.3.1.1 *Mallows' CL (CL)*

We can approximate MPE for LRR by utilizing a slightly modified version of Mallows' [137] *CL* as used for ORR

$$CL(\lambda_i) = \frac{1}{n}\left\| y - X\hat{b}_{\lambda_i} \right\|^2 + \frac{2\hat{\sigma}^2}{n}\text{trace}(H), \tag{4.2}$$

where $trace(H)$ is the sum of the diagonal elements of the hat matrix, which is defined as $H = X\left(X^T X + V\,\text{diag}\left(\lambda_i^2\right) V^T\right)^{-1} X^T$. *CL* is an unbiased estimate of the MPE, but it is only reliable for white Gaussian noise and correctly specified models.

Unlike ORR case in which one λ is optimized and then used for all input variables, optimization of *CL* for LRR requires multidimensional optimization of a vector of $\lambda_i$'s. The number of possible combinations of even a moderate number of real-valued ridge parameters becomes unmanageable even when considering a fairly coarse grid of possible ridge parameters values. The MATLAB code used to implement this objective function is provided in Appendix A.2 [226].

4.3.1.2 *Information Complexity Regularization Parameter Selection (ICOMPRPS)*

With a limited number of observations, *CL's* estimation inaccuracy penalization term in (4.2) becomes inadequate requiring further refinement. The information complexity regularization parameter selection (ICOMPRPS) method has been demonstrated to provide additional penalization of the estimation inaccuracy and to behave favorably for a limited number of observations [224]:

$$ICOMPRPS(\lambda_i) = \tfrac{1}{n}\left\| y - X\hat{b}_{\lambda_i} \right\|^2 + \tfrac{2\hat{\sigma}^2}{n}\left(\text{trace}(H) + C_1(J)\right), \tag{4.3}$$

where $\hat{J} = \left(X^T X + \text{diag}(\lambda_i)\right)^{-1}$. $C_1$ is the maximal covariance complexity index, which is used to measure the degree of interdependency between parameter estimates [228]. $C_1$ is computed as $C_1(J) = m/2 \log \bar{v}_a / \bar{v}_g$, where $\bar{v}_a$ and $\bar{v}_g$, are the arithmetical and geometrical means of the eigenvalues of $J$. The MATLAB code used to implement this objective function is provided in Appendix A.1 [226].

4.3.1.3 *Generalized Cross-Validation (GCC)*

Probably the most widely used noise-level-free regularization parameter selection method is generalized cross validation (GCV) [237]. According to this method, the vector of regularization parameters, $\lambda_i$'s, are chosen that minimize the GCV function:

$$GCV(\lambda_i) = \tfrac{1}{n}\left\| y - X\hat{b}_{\lambda_i} \right\|^2 \Big/ \left(\tfrac{1}{n}\text{trace}(I_n - H)\right)^2 \tag{4.4}$$

GCV requires no prior knowledge of the noise level and it is insensitive to white Gaussian noise in the response. GCV occasionally fails, presumably due to the presence of correlated noise [237]. The MATLAB code used to implement this objective function is provided in Appendix A.3 [226].

### 4.3.2 LRR Parameter Representation for DE Operations

From what has been presented, one can see that selecting the optimal combination of LRR parameters, which minimizes one of the objective functions for a particular set of training data, requires optimization of all parameters simultaneously. The mapping of *regularization parameters* or *objective variables* on to a population of DE members was as follows. Each regularization *parameter* or *objective variable* $\lambda_i$ in a vector of LRR parameters $\boldsymbol{\lambda} = \lambda_i, i = 1, \ldots, d$ was represented as a *chromosome,* $x_{j,i}$, of a particular individual, $\mathbf{x}_i$, in a population, $\mathbf{P}$, of a specific generation, $\mathbf{P}_G = (\mathbf{x}_{1,G}, \ldots, \mathbf{x}_{i,G}, \ldots, \mathbf{x}_{NP,G})$.

### 4.3.3 Initialization

Almost any initialization strategy will work. If there is prior knowledge about the range of values that should be considered this can be used to set bounds and guide creation of the initial population which may accelerate convergence. If no prior knowledge is available, a good approach is to try and bound the range of expected values for each objective variable being optimized but even this is not critical.

The following algorithm was used to initialize a *D*-dimensional, population of NP members.

**Algorithm 4.2: Initialization Strategies**

1. Input values to be used for initializing the population: D: length of the population vectors to be created; seedFlag: indicates if a vector of truncated values is to be used to generate the new population; seedVec: vector of values to be used as a seed for seedXform; initOpt: initialization option selects the initialization strategy to be used; NP: size of the population; initMin: [1 x D] vector of values used to set the lower boundary; initMax: [1 x D] vector of values used to set the upper boundary; and $\mathbf{X}$ matrix of input variables
2. Determine if a population is to be created using the vector of parameters from the truncated objective function the call the appropriate initialization function.

```
if seedFlag = 'n'
   iPop = initPop(initOpt, NP, D, initMin, initMax, seedVec, X)
else
   iPop = seedXform(NP, seedVec, X)
end
```

**Function [iPop] = initPop(initOpt, D, initMin, initMax, $\mathbf{X}$)**

1. Create a population sampled form a uniform distribution if the initOpt = 'rand'

```
seedX = std(X);
pop = zeros(NP,nx);
for i=1:NP
   pop(i,:) = initMin + rand(1,nx).*(initMax - initMin);
end
```

2. Create a population sampled form a log distribution if the initOpt = 'logspace'

```
pop = zeros(NP,D);
for i = 1:nx
   ind=randperm(NP)';
   seedX  = logspace(-4,2,NP)';
   pop(:,i) = seedX(ind,:);
end
```

3. Create a population using a linearly spaced grid if the initOpt = 'lin'

```
pop = zeros(NP,D);
if nx==2
    nGrid=floor(sqrt(NP));
    s1=linspace(initMin(1),initMax(1),nGrid)';
    s2=linspace(initMin(2),initMax(2),nGrid)';
    [h1,h2]=meshgrid(s1,s2);
    h1c=reshape(h1,length(h1).^2,1);
    h2c=reshape(h2,length(h2).^2,1);
    pop0=[h1c,h2c];
    needMorePop = NP-(floor(sqrt(NP)))^2;
    if needMorePop > 0
        for m = 1:needMorePop
            popMakeUp(m,:) = initMin + rand(1,nx).*(initMax - initMin);
        end
        pop=[pop0;popMakeUp];
    end
else
    for i=1:D
        ind=randperm(NP)';
        seedX  = linspace(initMin(i),initMax(i),NP)';
        pop(:,i) = seedX(ind,:);
    end
end
```

4. Return the initialized population

```
Pop=pop
```

### Function [iPop] = txformSeed(NP, seedVec, X)

1. Determine the number of dimensions in the input data matrix

```
[mx,nx]=size(X);
```

2. Compute singular values for $\mathbf{X}$

```
bestmem=seedVec;
D=nx;
[U,s,V] = csvd(X);
max(s);
seedMax = 100*s';    % 100*max(s);
seed = not(round(bestmem)) .* seedMax;
seedvec = not(round(bestmem)) .* seedMax;
```

3. Create an initial population using an exponential probability density function (expdf) of values from 0 up to the singular value for those parameters that were passed by the truncated objective function

```
expdf_pass = exprnd(1/6,NP,1);        % initial exponential pdf to be
                                      %  multiplied by the singular
                                      %  values;
ipop_pass = expdf_pass*s';
for i = 1:D
    ipop_pass(:,i) = round(bestmem(i)) * ipop_pass(:,i);
end % End For i = 1:D
```

4. For those parameters that were dumped by the truncated objective function create an initial population with an expdf from 100 times the singular value down to the singular value

```
expdf_dump = abs(100*(1-exprnd(1/6,NP,1)));
ipop_dump = expdf_dump*s';
for i = 1:D
    ipop_dump(:,i) = not(round(bestmem(i))) * ipop_dump(:,i);
end % End For i = 1:D
ipop = ipop_pass + ipop_dump;if findstr(lower(initOpt),'std')
```

5. return

A technique that was found to be especially useful for quickly identifying the most relevant components for LRR was to use a truncating version of the objective function. The truncating objective function would completely pass or completely dump a component depending on whether the parameter value was a '1' or '0'. Optimization for this approach was significantly faster than using the continuous version of the objective function for all objective variables. The vector of truncated LRR parameters returned was then used as a seed vector for creating a new initial population to which the continuous version of the objective function would be applied. . The components passed ($x_{j,i}$=1) by the truncated objective function would have their initial values biased towards zero using an exponential probability distribution function (expdf). The components that were dumped ($x_{j,i}$=0) would have their initial values biased towards 1,000 times the singular value of the component using an exponential probability distribution function.

Figure 4.3 (a) presents a graphical example of using the truncated version of ICOMPRPS to guide creation of an initial population for the gas data set. The upper plot of Figure 4.3 (a) shows the current population with each objective variable 1-6 represented as dots. The black line and x's are the best member of the current generation (1's are passed, 0's are dumped), and the read line and asterisks are the singular values of each objective variable. Figure 4.3 (b) shows the initial population created using the best members from the truncating objective function. Notice the concentration of objective variables 1,3,4,5 and 6, which were passed by the truncating objective function, between 0 and their respective singular values and that of variable 2, which was dumped, is between the singular value and 1,000 times the maximum value.

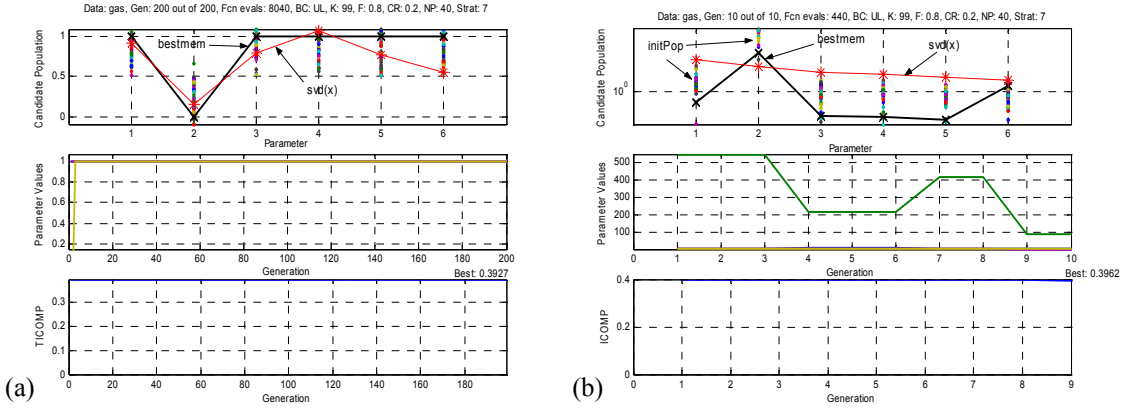



Figure 4.3 (a) A graphical example of using the truncated version of ICOMPRPS to seed creation of an initial population to be used with the continuous version of ICOMPRPS shown in (b).

The 'rand' initialization strategy was used primarily for DEALRR while the 'logspace' strategy was used most often for DEALKR. Once initialized each member of the population is evaluated using one of the objective functions described in the previous section.

### 4.3.4 Mutation, Recombination and Selection

DE/current-to-rand/1/bin (strategy 7) with the setting $NP = 40$, $CR = 0.2$, $F = 0.8$ and $K = rand_i$ (once per vector) was found to provide robust, reproducible results for a variety of data sets tested. Occasionally, NP was reduced to increase execution speed.

#### 4.3.4.1 *Enforcing Boundary Constraints at Runtime*

Soft asymptotic boundaries were implemented for both truncating and non-truncating strategies. For truncating case the lower boundary was set at zero and the upper at one. And for the non-truncating case the lower boundary was set at zero but the upper boundary was set at 1,000 times the singular value of the respective objective variable.

### 4.3.5 Termination criteria

An automated termination strategy was not implemented as part of this work. Termination of a particular run was generally determined by observing the evolutionary progress of the population and looking for such things as i) a lack of diversity or variation in the population, ii) little if any change in the parameter values for the best member for extended generations, iii) little if any change stability in the best value of the objective function for the best member. It is worth noting that strategy 7 (DE/current-to-rand/1/bin) with

setting of $NP = 40$, $CR = 0.2$, $F = 0.8$ and $K = rand_i$ (once per vector) produced repeatable results in nearly all test cases considered for DEALRR within 2,000 generations. That is to say this combination of strategy and settings worked well on the limited range of data used for this part of the research. Further testing and evaluation of this configuration on a wide variety of data is needed before one could conclude if this configuration was robust enough to be used in a "set and forget" fashion.

## 4.4 DE Automated Selection of Optimal Localized Kernel Regression Bandwidth Parameters (DEALKR)

This section describes how DE was used to address the LKR problem of automatically selecting optimal bandwidth parameters and includes the objective function used, the bandwidth matrix representation for DE operations, population initialization, and the strategy and parameter configurations. This approach is referred to as DEALKR for **D**ifferential **E**volution enabled **A**utomatic selection of optimal **L**ocalized **K**ernel **R**egression bandwidth parameters.

### 4.4.1 LOO-CV as an Objective Function for DEALKR

LOO-CV was selected as the objective function to be minimized for DEALKR. While it is the most computationally intensive form of cross-validation it makes use of the largest amount of data and based on the authors experience is more reliable than other cross-validation methods especially for sparse data sets. MSE is used to report and compare training and testing results.

#### 4.4.1.1 *Leave-One-Out Cross-Validation*

Cross-validation is a method of averaging the effect of choosing a particular validation set over several other sets of the same size. Ideally one would like to consider all possible sets, but this is impracticable. So we deal with only a disjoint subset of validation sets. The data is split into $Q$ sets $S_j$ each of which are of size $V$. This would make the total number of sets $N = QV$. For simplicity we will assume that set $S_1$ contains data examples 1 to $V$, and that set $S_2$ contains data $V+1$ to $2V$ and so on. $Q$ different models $\hat{f}_j$ are trained leaving out every set $S_j$ in succession. The cross-validation estimate is then the average of these models on the left-out subset [72]:

$$\hat{G}_{CV} = \frac{1}{N}\sum_{j=1}^{V}\sum_{k\in S_j}(f_i(x^{(k)}) - y^{(k)})^2 \tag{4.5}$$

Leave-one-out cross-validation is a special case of cross-validation when V=1. To explore this first consider a model $\hat{f}k$ that is obtained when $k$ is left out. Applying the limit to equation leads to:

$$\hat{G}_{LOO} = \frac{1}{N}\sum_{k=1}^{N}(\hat{f}k(x^{(k)}) - y^{(k)})^2 \tag{4.6}$$

This is objective function that was used for all DEALKR, CGLS and global bandwidth selections.

So the optimal bandwidth matrix for a particular data set is the one that minimizes the LOO-CVE. $\text{DEALKR}_\text{F}$ produced the lowest LOO-CVE in every case. The other metrics for comparing performance of the various methods were the training and test MSE.

4.4.1.2 *Mean-Squared Error*

Comparisons of results were made using the MSE:

$$MSE(M) = \frac{1}{n}\left(y - \hat{y}_M\right)^2, \tag{4.7}$$

where, M is the method, e.g., OLS, ORR, DEALRR, $\text{DEALKR}_\text{D}$, $\text{DEALKR}_\text{F}$, CGLS or global, $y$ is the true output, $\hat{y}$ is the output prediction of the method, and $n$ is the number of points in $y$. MSE values were calculated for both training data and the test data. The training MSE is an indicator how well the technique fit the training data and the test MSE indicates how well the technique generalizes by its performance on data it has never seen. Determination of the best technique is based on the test MSE.

### 4.4.2 Bandwidth Matrix Representation for DE Operations

Mapping the *full matrix* of *bandwidth parameters* as *objective variables* enables all levels $\mathbf{H} \in S$, $\mathbf{H} \in D$ and $\mathbf{H} \in F$ to be optimized and evaluated. An arbitrary *m x m* bandwidth matrix was represented as follows.

$$\mathbf{H}_{i,G} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & & h_{2,m} \\ \vdots & & \ddots & \vdots \\ h_{m,1} & h_{m,2} & \cdots & h_{m,m} \end{bmatrix}_{i,G} = \left[h_{1,1}, h_{2,1}, \ldots, h_{m,1}, h_{1,2}, h_{2,2}, \ldots h_{1,m}, h_{2,m}, \ldots, h_{m,m}\right]_{i,G} = \tag{4.8}$$

$$x_{1,i,G}, \ldots, x_{j,i,G} = \mathbf{x}_{i,G}, \quad i = 1, \ldots, NP, \quad \textit{and} \quad j = 1, \ldots, D^2$$

where $G$ is the current generation and $NP$ is the size of the population. When $\mathbf{H} \in D$, $\mathbf{H} = diag(h_1, \ldots, h_m)$ and when $\mathbf{H} \in S$, $\mathbf{H} = h\mathbf{I}$. With this representational mapping it should be is easy to see that other variables could be targeted for optimization in parallel with the bandwidth parameters, such as bandwidth shape or if localized polynomial regreesion were used, the degree of a polynomial. This would be a natural extension since LKR is actually a zero-order polynomial.

### 4.4.3 Initialization

Starting populations were created using the 'logspace' initialization option described in Algorithm 4.2.

### 4.4.4 Mutation, Recombination and Selection

Both DE/rand-to-best/1/bin (strategy 3) and DE/current-to-rand/1/bin (strategy 7) were used for DEALKR. The settings used most frequently were for strategy 3: $NP$ = 20, $CR$ = 0.2, $F$ = 0.8 and for strategy 7: $NP$ = 20, $CR$ = 0.2, and $F$ = 0.8. For the real world data sets with eighty two input variables $NP$ = 40.

#### 4.4.4.1 *Enforcing Boundary Constraints at Runtime*

A soft lower boundary was set to zero while an upper boundary was not set.

### 4.4.5 Termination criteria

As with DEALRR an automated termination strategy was not implemented for DEALKR. Termination of a particular run was determined by observing the output history of the evolutionary progress. Maybe an example will help to illustrate how this was done.

Snapshots at three different points in the evolutionary history of a DEALKR run for RampHill1_100 are presented in Figure 4.4. The left panels show the history in three different subplots. The top subplot shows the individual parameters of the current population as dots and the parameters of the best member so far as best x's connected with a line. The middle subplot traces the histories of the best values for each parameter. The bottom subplot is the history of the objective function value for the best member, which in this case is the LOO-CVE. The right panels present a 3D visualization of the objective function values for each individual of the current generation as dots and the best member of each successive generation as blue crosses. Notice the change in population diversity from generation 10 to 40 and then form 40 to 250. Also notice there is virtually no change in the best member parameter values or the LOO-CVE after about generation 70. The same type of characteristic convergence is depicted by the clustering affect observed in the 3D LOO-CVE plots form generation 10 to 40 and 40 to 250. These were the kinds of heuristic

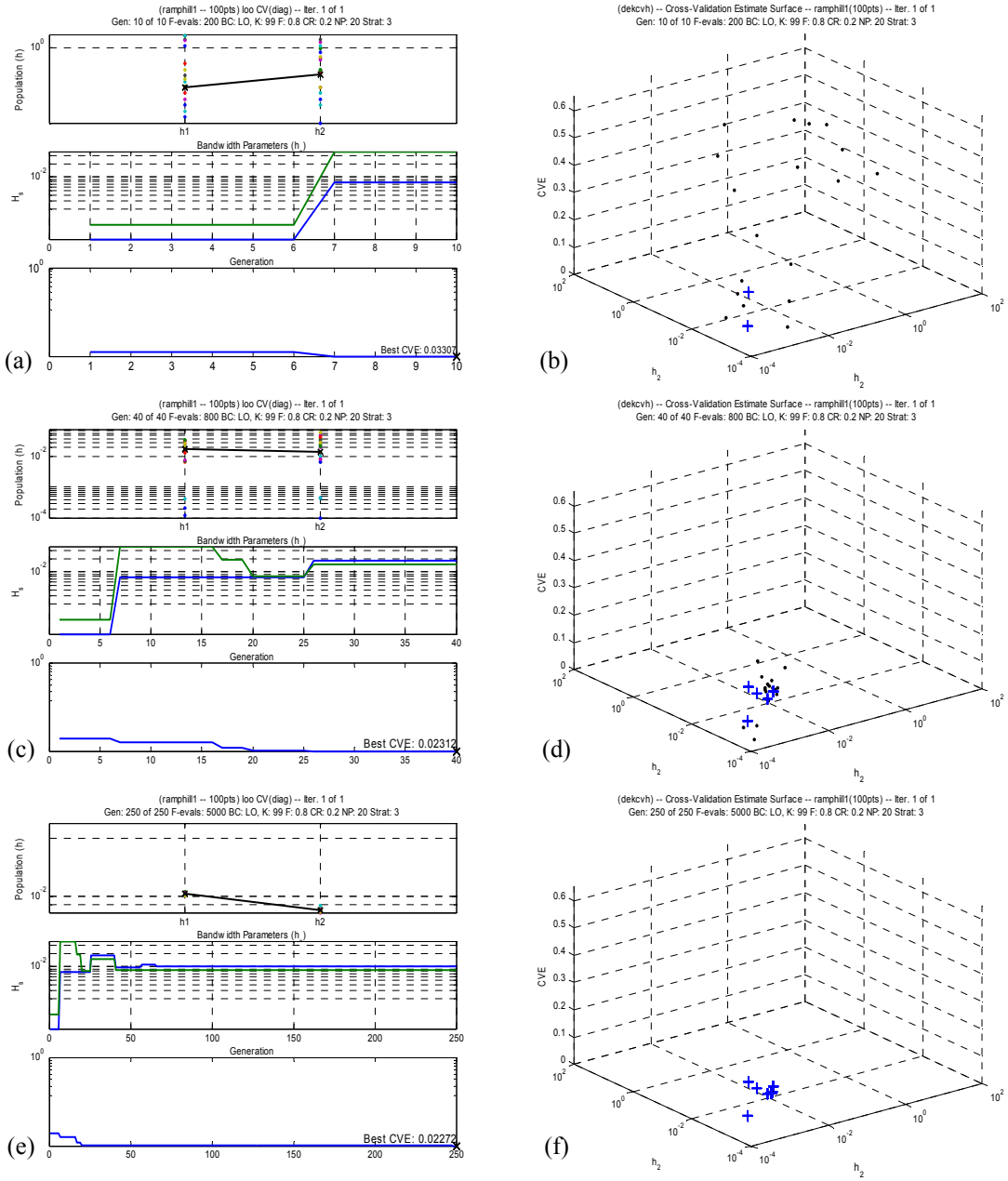



Figure 4.4 Snapshots of a DEALKR evolutionary history for RampHill1_100. (a) History through generation 10. (b) 3D view of LOO-CVE history at generation 10. Dots are members of the current population. Blue +'s represent best members of each generation. (c) History through Generation 40. (d) 3D view of LOO-CVE history at generation 40. (e) History through generation 250. (f) 3D view of LOO-CVE history at generation 250.

indicators used to decide when to terminate a particular run. Another is to execute multiple runs and observe whether statistically similar results are obtained. If the LOO-CVE surface contains multiple minima the convergence rate will be slowed down but as long as diversity was maintained in the population DE always converged to a confirmed global minimum.

Chapter

5

# Method of Detecting Local Minima in Multidimensional Data (ModelM)

Before presenting results of the test cases let us explore the issue of local minima and describe how to determine whether or not they exist. Multiple two-input single-output data sets will be used to explain the approach, which theoretically it could be applied to data sets of any dimension. The approach will be referred to as ModelM for a **M**ethod **o**f **de**tecting **l**ocal minima in **M**ultidimensional data.

## 5.1 The Problem with Local Minima and Sparse Data

The existence of local minima in the error surface of a function can render the results obtained by global bandwidth techniques and those based on gradient techniques untrustworthy. Once the input dimension exceeds two there is no practical way to visualize the error surface and so it is virtually impossible to know whether the bandwidth parameters obtained by a gradient technique are for the global minimum or one of many possible sub-optimal local minima. This represents a classical ill-posed problem: there is no unique solution for the given data set.

This section demonstrates the affect of local minima on LKR results obtained by CGLS for a two-input and one-output data model. The size of the training data sets were 50, 100, 200, and 500 with 2% noise added to the output. The ability of DE to find the global minimum is demonstrated for all training sets. Also described in this section is a method of detecting the existence of local minima in multivariate data sets.

The next section describes the data sets used to explore the problem of local minima. All data sets used in this section are based on a variation of a function described by Goutte in [73].

### 5.1.1 S4G1_50 Data Set

The frequency component in the $x_1$ direction was designed to be three times higher than that of the original. The modified function is described by:

$$y = (\sin(12x_1) + e^{(0.5x_2)}) \times e^{(-x_1^2 - x_2^2)/4}, \; -2 \le x_1 \le 2, \quad and \quad 0 \le x_2 \le 1. \tag{5.1}$$

The original function used $\sin(3x_1)$ instead of $\sin(12x_1)$. The data sets were designated S4G1_x, where 'S4' is used to identify it as a sigmoidal function with a frequency 4 times higher than the original, 'G1' because it was derived from Goutte's work, and '_x' indicates the number of training points in the data set. Figure 5.1 (a) presents S4G1_50, with 50 training points shown as asterisks and 5,041 test points as a surface plot. An accurate approximation of this function will require a kernel with a small bandwidth in the $x_1$ direction, to adequately cover the higher frequency components, and the large bandwidth in the $x_2$ direction. This was the primary reason this function was created. The data sets used for this part of the work differ only in the number of input-out data pairs used for training: 50 for S4G1_50, 100 for S4G1_100, 200 for S4G1_200 and 500 for S4G1_500.

#### 5.1.1.1 *LOO-CVE Based Optimal Solution*

The optimal solution or set of bandwidth parameters for the 50 sample training data was selected by minimizing LOO-CVE which will be referred to as the global minimum. The LOO-CVE for S4G1_50 was computed for 10,000 ($h_1$,$h_2$) bandwidth pairs generated on a log-log grid between $10^{-4}$ and $10^3$. The resulting LOO-CVE surface is presented in Figure 5.1 (b). Notice the presence of several basin-like attractor regions in addition to the global minimum which had a LOO-CVE = 0.1657 at $h_1$,$h_2$ = (0.005, 1.0). A 3D view of the Gaussian kernel produced by the optimal bandwidth parameters is shown in Figure 5.1 (c). Notice the bandwidth in the $x_2$ direction is 200 times larger the bandwidth in the $x_1$ direction. This is consistent with the statement made in the previous paragraph that it would require a kernel with a small $x_1$ bandwidth and a larger $x_2$ bandwidth to accurately approximate the S4G1 function. The resulting function approximation with a test MSE = 0.1321 is shown in Figure 5.1 (d).

#### 5.1.1.2 *Global Bandwidth Results*

An optimal global bandwidth parameter, e.g., $\mathbf{H} \in S$, was determined by computing the LOO-CVE for 200 different values of $h$ from $10^{-4}$ to $10^3$ and selecting the one that produced the minimum LOO-CVE. Figure 5.2 (a) shows the LOO-CVE plot with a minimum of 0.2291 at h = 0.4553. The resulting function

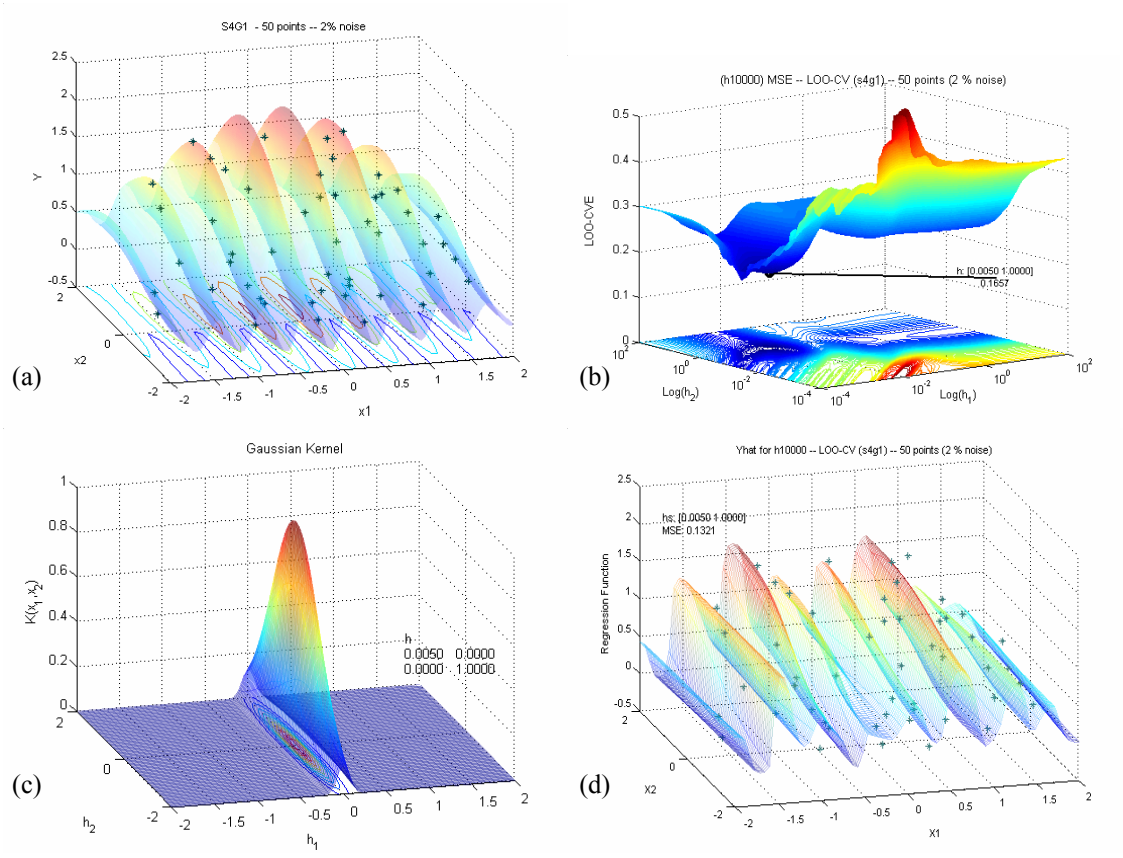



Figure 5.1 S4G1_50 Data set: (a) Surface plot of test data; asterisks are training data (b) LOO-CVE error surface. (c) Gaussian kernel produced by optimal LOO-CVE bandwidths. (d) Function approximation using Gaussian kernel with optimal bandwidths.

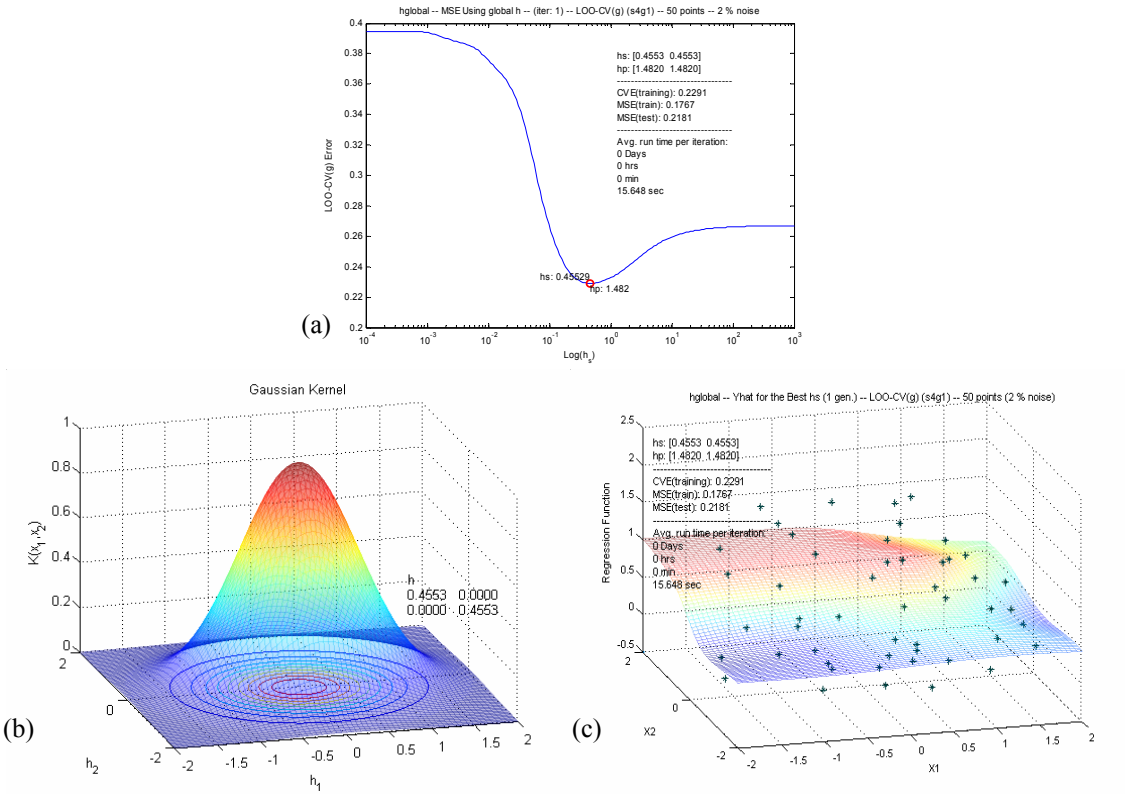



Figure 5.2 Global bandwidth solution for S4G1_50: (a) LOO-CVE plot. (b) Gaussian kernel shape for global bandwidth (0.4553). (c) Function approximation using the global bandwidth.

approximation produced a test MSE = 0.2181 and a training MSE = 0.1767. The Gaussian kernel produced by this bandwidth is shown in Figure 5.2 (b). Figure 5.2(c) shows the function approximation of S4G1_50 using the global $h$ as a mesh plot along with a surface plot of the test data and the training data points as asterisks. This clearly demonstrates that a kernel using a global bandwidth may severely over-smooth the results. Note also that the test MSE is nearly two times the training MSE.

#### 5.1.1.3 *Starting point Sensitivity of CGLS*

It is a common strategy among researchers/scientist using a gradient based technique to select the starting point randomly. In the absence of any prior knowledge upon which to make the decision, one random guess would be as valid as another. A bandwidth that is very small, with respect to the distance between training points, will result in over-fitting, while a bandwidth that is very large – covering the entire training data space – will result in over-smoothing. The initial guess used for CGLS in this research takes advantage of the fact that the gradient being descended is a function of the distance between data points along a particular input dimension. Given this dependency the standard deviation of the predictor variables, $std(\mathbf{x})$, was used as a basis for the initial guess. A grid of values that approximately covers the range of potential bandwidth parameters – from a very small distance ($10^{-4}$) to the entire search space ($10^{2}$) – (see Figure 5.3(a)) was used to control the starting points for CGLS. This has the crude effect of surveying the error surface. The LOO-CVE output at every evaluation step along the descent path was recorded for each starting point. The results for a grid of 100 different starting points has been superimposed on top of the LOO-CVE surface of sg41_50 and plotted in Figure 5.3(b) and (c). The blue dots indicate the 100 different staring points for CGLS. The smaller black dots represent steps taken along each descent path. Each red x represents a final convergence point for every CGLS starting point. The green dot represents the CGLS initial guess, $std(\mathbf{x})$. The green asterisk indicates the convergence point of the initial guess. The distribution of red x's clearly indicates the presence of many local minima. This figure strongly demonstrates the importance of i) knowing whether or not the underlying error-surface contains local minima and ii) choosing a good starting point. Figure 5.3(d) is a histogram of the smoothing parameters $h_p$ resulting from the 100 different starting points. Notice the rather large standard deviation for both $h1_p$ and $h2_p$, the result of multiple convergence points and harbinger of local minima. Having identified this correlation between multiple local minima and the standard deviation in the smoothing parameters we now have a proverbial smoking gun that indicates the existence of local minima.

Figure 5.4 (a) is the Gaussian kernel produced using the bandwidths $h_1,h_2$ = (0.4643,0.4725) obtained from using the initial guess, $std(\mathbf{x})$, and Figure 5.4 (b) shows the resulting function approximation of S4G1_50 based on these bandwidths with a LOO-CVE = 0.2291, a training MSE = 0.1778, and a test MSE = 0.2181.

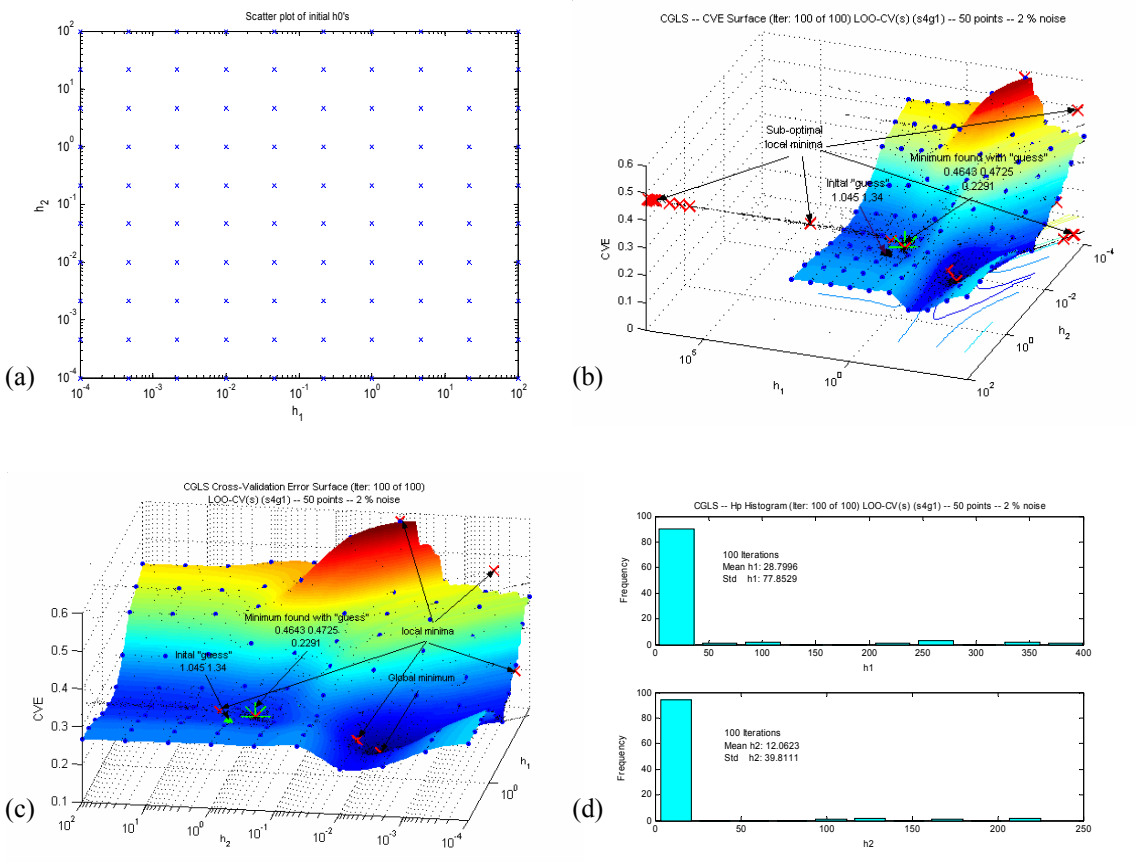



Figure 5.3 ModelM results for the S4G1_50 data set: (a) Grid of initial starting points. (b) ModelM results superimposed over the optimal LOO-CVE surface. The blue dots indicate a starting point for each run. The red X's indicate a point of convergence for each run. The green dot indicates the "initial guess" based on std(x), and the green asterisk indicates the final minimum obtained using the initial guess. Note the large number of red x's to the far left and that the initial guess did not converge to the global minimum. (c) A closer view of the ModelM results layered over the optimal LOO-CVE surface. (d) Smoothing parameter histograms from the 100 CGLS runs, with mean and standard deviations for each smoothing parameter.

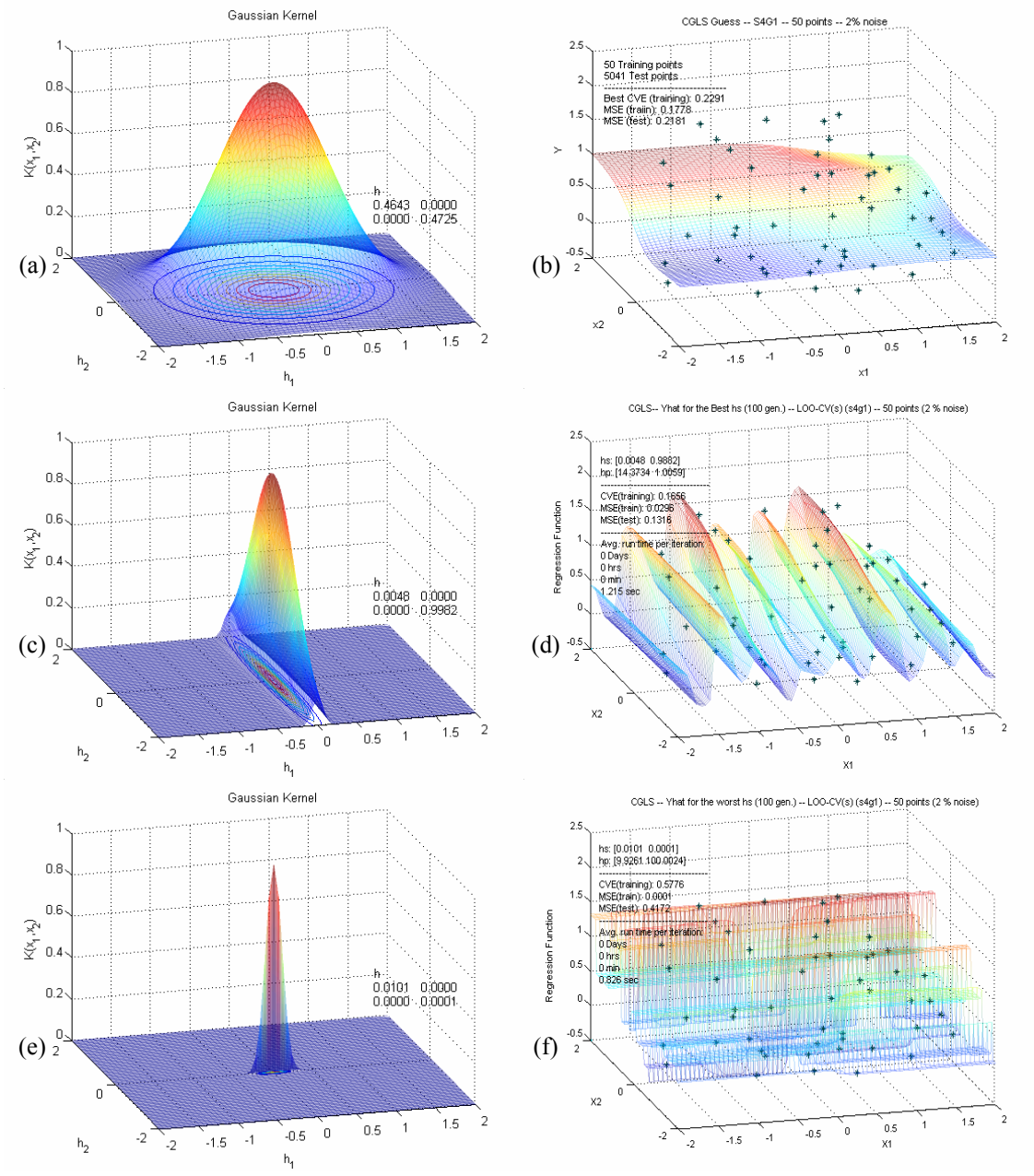



Figure 5.4 Gaussian kernels and function approximations obtained by ModelM for S4G1_50: (a) Gaussian kernel for bandwidths obtained using $std(\mathbf{x})$ as an initial guess. (b) Function approximation produced by using results from the initial guess. Note this was not the optimal solution (global minimum). (c) Gaussian kernel for best bandwidths obtained by ModelM. (d) Function approximation using the best bandwidths obtained by ModelM. (e) Gaussian kernel for worst values of *h*. (f) Function approximation using the worst bandwidths obtained by ModelM.

Figure 5.4 (c) is the Gaussian kernel produced by the best bandwidths $h_1,h_2$ = (0.0048,0.9982) obtained by ModelM, and Figure 5.4 (d) shows the resulting function approximation of S4G1_50 based on the best bandwidths with a LOO-CVE = 0.1656, a training MSE = 0.0296, and a test MSE = 0.1316. These values match the optimal values of $h_1,h_2$ = (0.005, 1.0). Figure 5.4 (e) shows the Gaussian kernel produced by the worst bandwidths $h_1,h_2$ = (0.010,0.0001) obtained by ModelM. The resulting function approximation of with LOO-CVE = 0.5776, a training MSE = 0.0001, and a test MSE = 0.4172 are shown in Figure 5.4 (f). This demonstrates a good example of over fitting the training data as evidenced by the low training MSE.

#### 5.1.1.4 *Detecting the Presence of Local Minima in Higher Dimensional Data*

It is easy enough to visualize the results for a data set with two predictor variables but most real world data consist of more than two input variables. In order to demonstrate how ModelM can be applied to higher dimensional data sets let us take a look at the histogram of smoothing parameters and associated standard deviations. A smoothing parameter $h_p$ is the reciprocal of the bandwidth parameter ($1/h_s$). The smoothing parameter was described by Goutte [158] and was an artifact of AMKREG. When working with data sets that have been scaled the magnitude of a smoothing parameter can serve as an indicator of how relevant the associated input variable is to predicting the output. The smoothing parameters of highly relevant variables will be larger than those of lesser relevance. The smoothing parameter histogram and standard deviation can also serve as a good indicator of whether local minima exist in the error surface. If there were only a single global minimum the histograms would be nearly singular for each parameter and have a small standard deviation. As a result of this finding a method for detecting local minima (ModelM) is proposed that can be applied to data sets of any dimension. This method uses the rapid convergence characteristics of CGLS to affectively survey the error surface by controlling the starting points of a population of CGLS "scouts". Histograms of each bandwidth parameter can be used to visualize results that have more than two input dimensions. If there is only as single minimum then all of the histograms would be nearly singular and have small standard deviations, because all "scouts", even though they start at different points on the error surface or error-hyper-surface for higher dimensions, should converge to the “same” global minimum. A large standard deviation in any of the smoothing parameter histograms would be a strong indicator that local minima exist and another technique such as DEALKR should be used.

#### 5.1.1.5 *$DEALKR_D$ Results*

This section reports on the results of selecting optimal $\mathbf{H} \in D$ bandwidths using DE which will be identified as $DEALKR_D$. The following parameters and initialization options were used for all examples in this section: Strategy 3 (DE/rand-to-best/1) [see 4.2.4 for an explanation] $NP$ = 20, the population was initialized using ‘logInit’, F = 0.8, $CR$ = 0.2; lower boundary constraint > 0. Figure 5.5 (a) shows the LOO-

CVE convergence results obtained after DEALKR$_D$ run of 250 generations. The Gaussian kernel resulting from the bandwidths obtained by DEALKR$_D$ is shown in Figure 5.5 (b). Figure 5.5 (c) presents the function approximation based on diagonal bandwidth parameters of $h_1,h_2$ = (0.0048, 0.9874) which resulted in a LOO-CVE = 0.1656, training MSE = 0.0296 and test MSE = 0.1315.

#### 5.1.1.6 *DEALKR$_F$ Results*

Results based on optimization of the full bandwidth matrix, $\mathbf{H} \in F$ , will be identified as DEALKR$_F$. Figure 5.5 (d) shows the Gaussian kernel produced by the full bandwidth matrix. The resulting function approximation with LOO-CVE = 0.1531, training MSE = 0.0139 and test MSE = 0.1245 is shown in Figure 5.5 (e). Notice that the full bandwidth matrix produced a test MSE that was lower than for the optimal bandwidth parameters which was 0.1321. This can be attributed to the fact that the true global minimum fell between "in the cracks" of the grid of points used to compute the LOO-CVE global optimum. Since CGSL and DEALKRD and DEALKRF are not constrained to the discrete points on the grid it is not surprising that they would find a LOO-CVE that is lower than that found using the grid.

The results for S4G1_100 are summarized in Table 5.1

### 5.1.2 S4G1_100 Data Set

Data set S4G1_100 (see Figure 5.6 (a)) was generated the same way as S4G1_50. First let's look at the optimal solution based on the LOO-CVE surface.

#### 5.1.2.1 *LOO-CVE Based Optimal Solution*

The LOO-CVE surface for S4G1_100 is shown in Figure 5.6 (b). The LCO-CVE was of 0.0636 for bandwidths $h_1,h_2$ = (0.0016, 0.2477). The Gaussian kernel for these bandwidths is shown in Figure 5.6 (c). The function approximation resulted in a test MSE = 0.0693. It is shown as a mesh plot in Figure 5.6 (d), along with the surface plot of the test data and asterisks depicting the training data.

#### 5.1.2.2 *Global Bandwidth Results*

The optimal global bandwidth parameter was determined as before by computing the LOO-CVE for 200 different values of h from $10^{-4}$ to $10^{3.}$ Figure 5.7 (a) shows a plot of the LOO-CVE versus bandwidth parameter for S4G1_100. The minimum LOO-CVE of 0.1965 at $h$ = 0.3037 produced a function approximation with a test MSE = 0.2041 and training MSE = 0.1634. The resulting Gaussian kernel is shown in Figure 5.7 (b) and the function approximation is presented in Figure 5.7 (c).

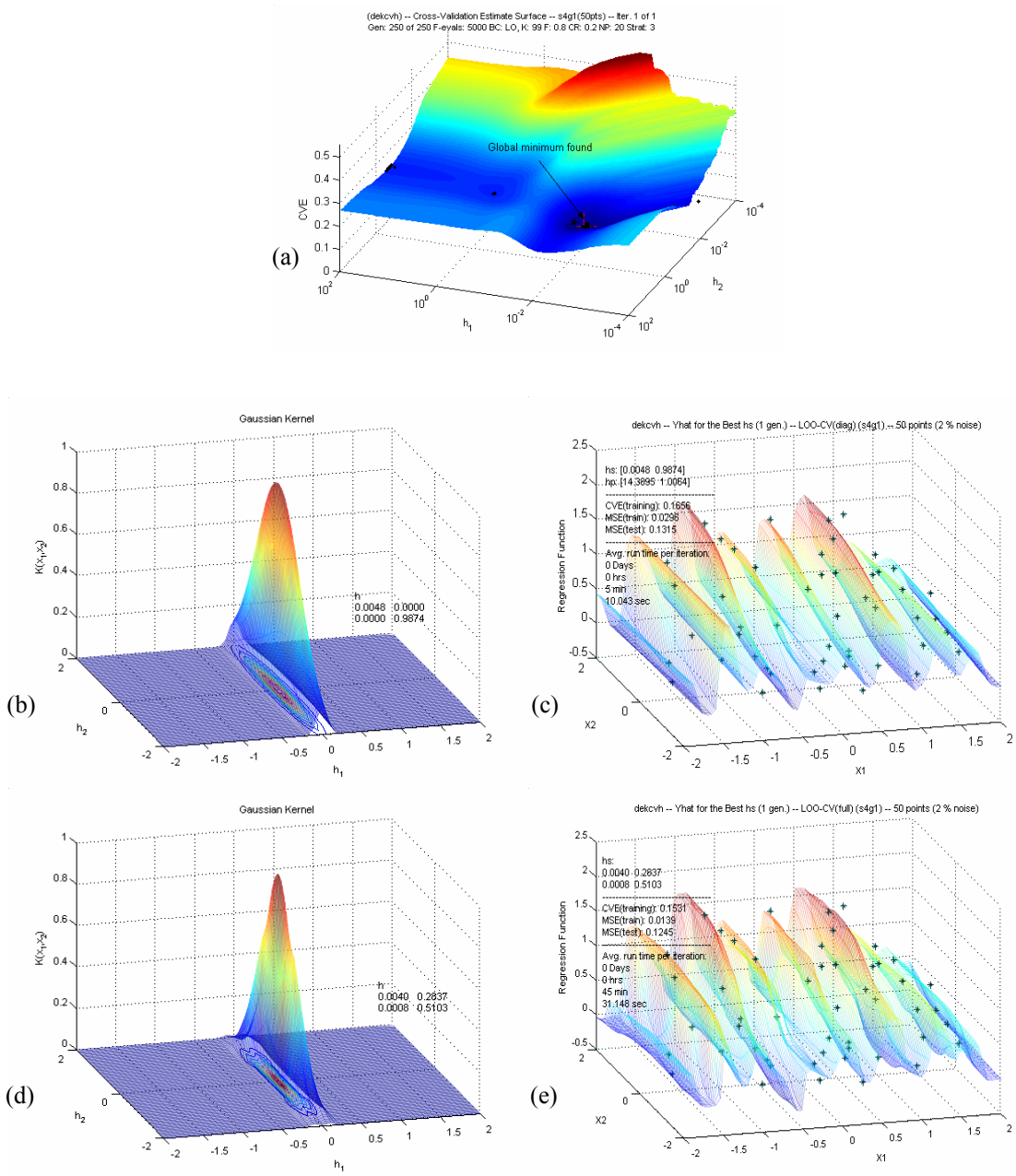



Figure 5.5 DEALKR results for S4G1_50: (a) DEALKRD 250 generation evolutionary history superimposed on the optimal LOO-CVE surface for S4G1_50. The black +'s indicate various "best members" throughout the 250 generation history. The red "+" indicates the final "best member" at the end of 250 generations. Note it is the global minimum. (b) Gaussian kernel for bandwidths obtained by DEALKRD. (c) Resulting function approximation using DEALKRD bandwidths (d) Gaussian kernel for full bandwidth matrix obtained by DEALKRF. (e) Resulting function approximation using full bandwidth matrix.

Table 5.1 Summary of results for S4G1_50 tests for local minima with rankings according to result categories: LOO-CVE., MSE(Train) and MSE(Test).

| Method | Data Set | S4G1_50 | Ranking |
|---|---|---|---|
| **LOO-CVE Optimal** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0050 & 0 \\ 0 & 1.000 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1657 | 3rd |
| | *MSE(Train):* | | |
| | *MSE(Test):* | 0.1321 | 4th |
| **Global** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.4553 & 0 \\ 0 & 0.4553 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.2291 | 4th |
| | *MSE(Train):* | 0.1767 | 4th |
| | *MSE(Test):* | 0.2181 | 5th |
| **CGLS 'std(x)'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.4643 & 0 \\ 0 & 0.4725 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.2291 | 4th |
| | *MSE(Train):* | 0.1778 | 5th |
| | *MSE(Test):* | 0.2181 | 5th |
| **CGLS 'best'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0048 & 0 \\ 0 & 0.9982 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1656 | 2nd |
| | *MSE(Train):* | 0.0296 | 3rd |
| | *MSE(Test):* | 0.1316 | 3rd |
| **CGLS 'worst'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0101 & 0 \\ 0 & 0.0001 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.5776 | 4th |
| | *MSE(Train):* | 0.0001 | 1st |
| | *MSE(Test):* | 0.4172 | 6th |
| **$\mathbf{DEALKR_D}$** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0048 & 0 \\ 0 & 0.9874 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1656 | 2nd |
| | *MSE(Train):* | 0.0296 | 3rd |
| | *MSE(Test):* | 0.1315 | 2nd |
| **$\mathbf{DEALKR_F}$** | $\begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0040 & 0.2837 \\ 0.0008 & 0.5103 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1531 | 1st |
| | *MSE(Train):* | 0.0139 | 2nd |
| | *MSE(Test):* | 0.1245 | 1st |

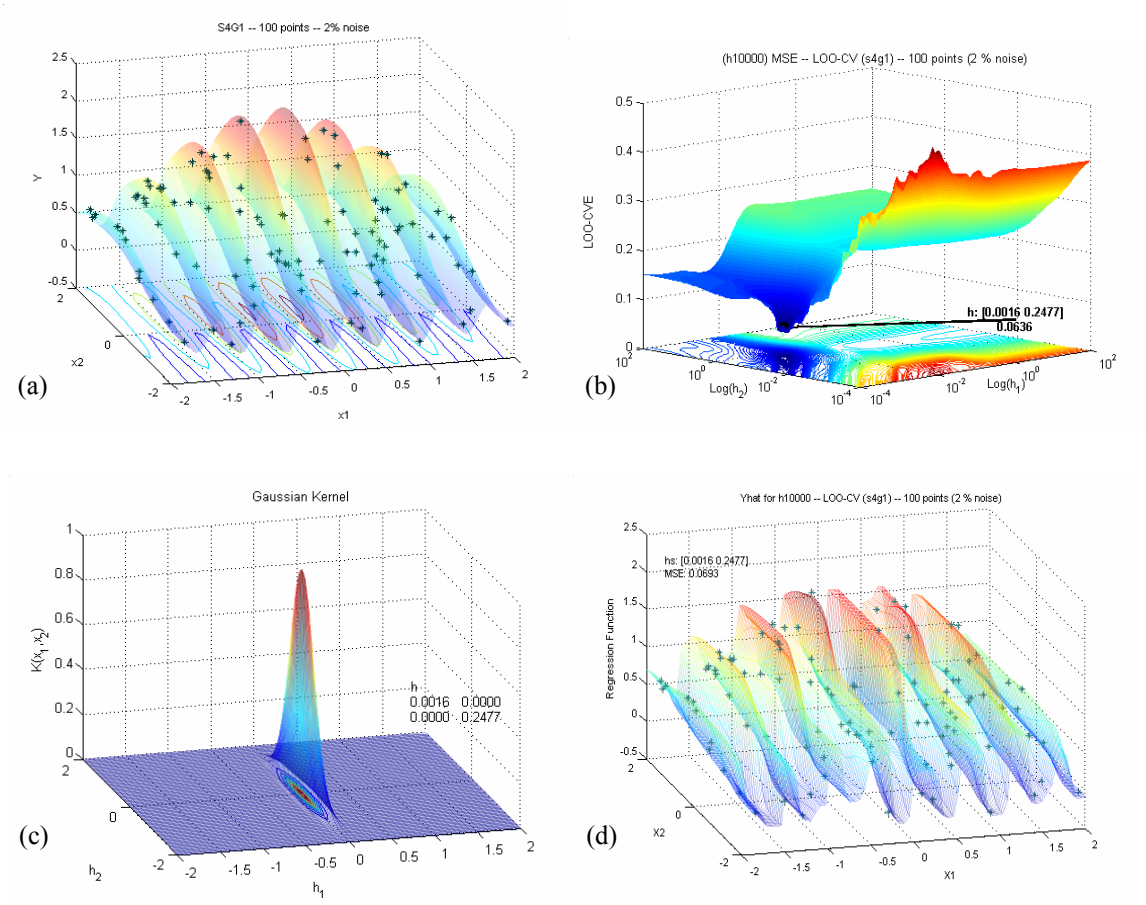



Figure 5.6 S4G1_100 Data Set: (a) Asterisks are training data, surface plot test data. (b) LOO-CVE surface. (c) Gaussian kernel produced by optimal LOO-CVE bandwidths. (d) Function approximation using Gaussian kernel with optimal bandwidths.

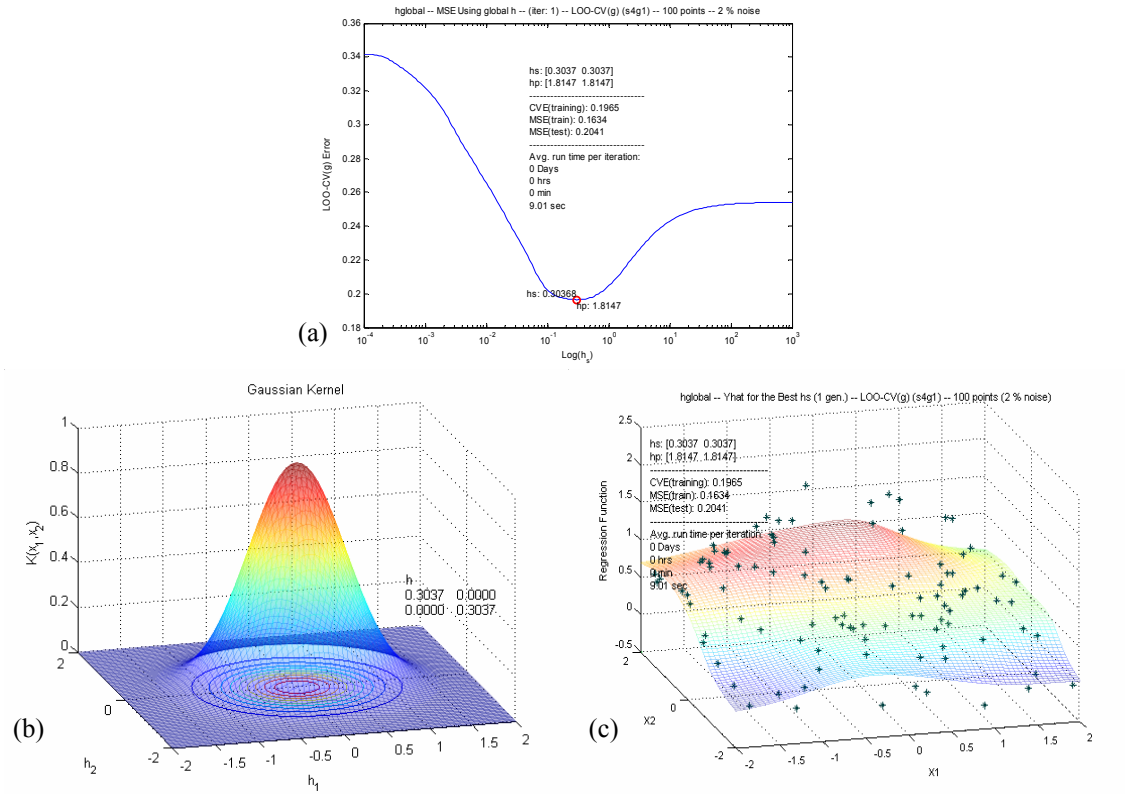



Figure 5.7 Global bandwidth solution for S4G1_100: (a) LOO-CVE plot. (b) Gaussian kernel shape for global bandwidth (0.3037). (c) Function approximation using the global bandwidth LOO-CVE = 1.814.

5.1.2.3 *ModelM and CGLS Results*

ModelM was implemented using a grid of starting points to survey the LOO-CVE surface and look for local minima. Figure 5.8 (a) shows the results superimposed on top of the LOO-CVE surface for sg41_100. The blue dots designate the 100 different staring points for CGLS. The smaller black dots are LOO-CVE values at steps taken along convergence paths. Each red x is a final convergence point. The black dot represents the "informed" initial guess based on the distribution of the underlying data. And the black asterisk indicates the convergence point of the informed initial guess. Notice that presence of multiple local minima represented by the read x's and that the initial guess did not result in converge to the optimal solution.

The initial guess produced a LOO-CVE = 0.1962 for bandwidth parameters $h_1,h_2$ = (0.8892, 0.2483), a training MSE =0.1721 and a test MSE =0.2087. Figure 5.8 (b) shows the Gaussian kernel produced by these bandwidth values.

The smoothing parameter histograms in Figure 5.8 (c) reveal large standard deviations, an indication that multiple minima are present. The function approximation produced by this set of bandwidths is shown as a mesh plot mesh plot in Figure 5.8 (d) along with a surface plot of the test data and the training data points as asterisks.

The Gaussian kernel for best bandwidth values obtained by ModelM $h_1,h_2$ = (0.0017, 0.2470) is shown in Figure 5.9 (a) the resulting function approximation with a LOO-CVE = 0.0635, a training MSE = 0.0036 and a test MSE = 0.0690 is shown in Figure 5.9 (b). Figure 5.9 (c).shows the Gaussian kernel for thy worst bandwidth parameters $h_1,h_2$ = (0.0003, 0.00001) obtained using ModelM and the resulting function approximation, which had a LOO-CVE = 0.4024, a training MSE = 0.0001 and a test MSE = 0.3712 is presented in Figure 5.9 (d). Again notice the affects of over fitting the training data.

5.1.2.4 *DEALKR$_D$ Results*

Figure 5.10 (a) shows the LOO-CVE convergence results obtained of DEALKR$_D$ after 250 generations superimposed on the LOO-CVE surface for S4G1_100. All parameters were the same as described previously. Notice the optimal solution was found as indicated by the red cross. Figure 5.10 (b) shows the Gaussian kernel produced by the bandwidths obtained by DEALKR$_D$. The resulting function approximation using $h_1,h_2$ = (0.0017, 0.2461) shown in Figure 5.10 (c) had a LOO-CVE = 0.0635, a training MSE = 0.0036 and a test MSE = 0.0690.

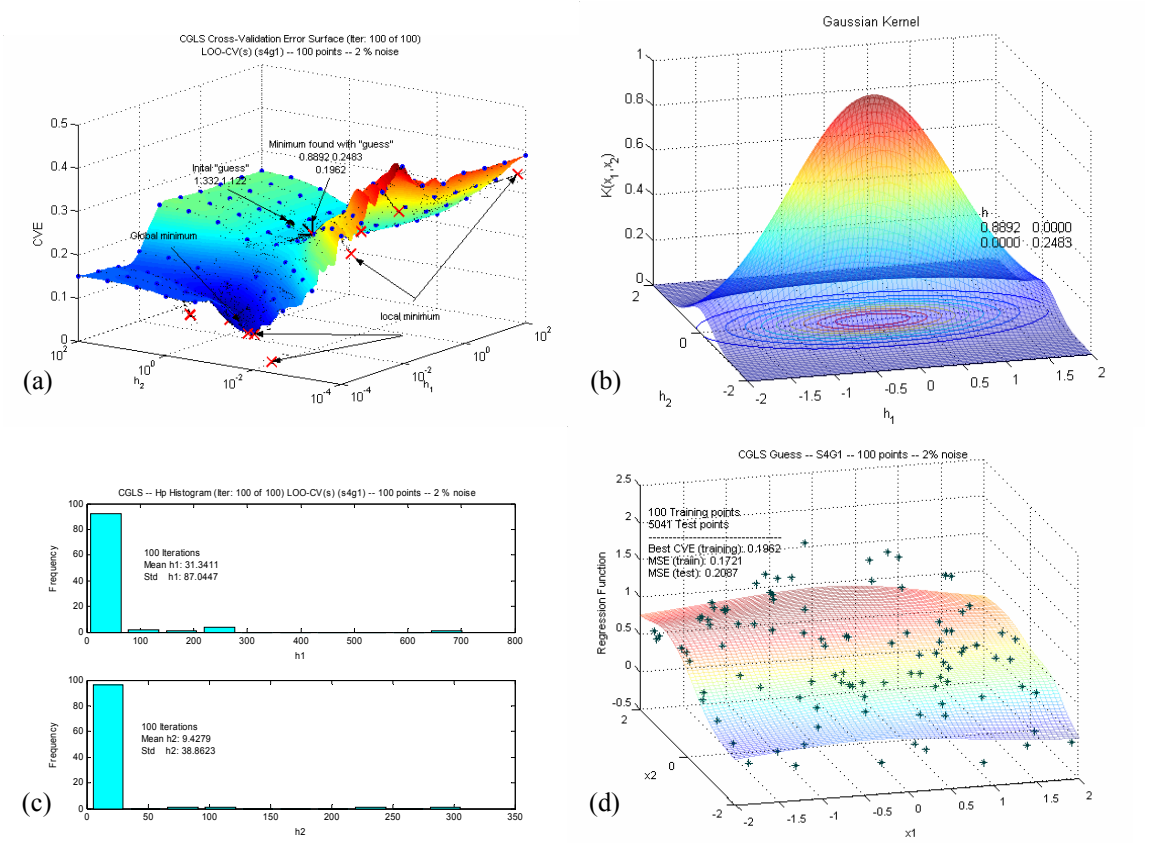



Figure 5.8 ModelM results for the S4G1_100 data set: (a) ModelM results superimposed over the optimal LOO-CVE surface. The blue dots indicate a starting point for each run. The red X's indicate a point of convergence for each run. The green dot indicates the "initial guess" based on std(x), and the green asterisk indicates the final minimum obtained using the initial guess. Note the initial guess did not converge to the global minimum. (b) Gaussian kernel for bandwidths obtained using $std(\mathbf{x})$ as an initial guess. (c) Smoothing parameter histograms from the 100 CGLS runs, with mean and standard deviations for each smoothing parameter. (d) Function approximation produced with results obtained by the initial guess.

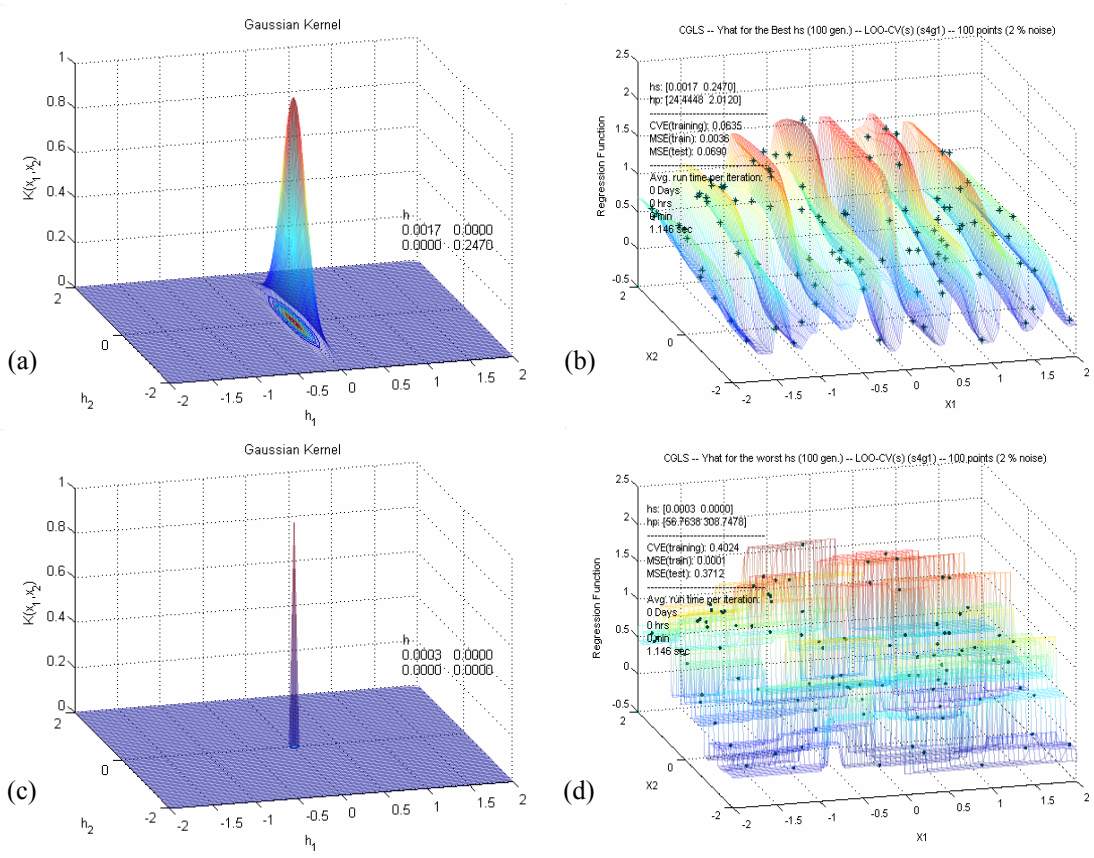



Figure 5.9 Gaussian kernels and function approximations obtained by ModelM for S4G1_100: (a) Gaussian kernel for best bandwidths obtained by ModelM. (b) Function approximation using the best bandwidths obtained by ModelM. (c) Gaussian kernel for worst values of *h*. (d) Function approximation using the worst bandwidths obtained by ModelM.

5.1.2.5 *$DEALKR_F$ Results*

Figure 5.10 (d) shows the Gaussian kernel produced by the bandwidths obtained by $DEALKR_F$. The resulting function approximation is shown in Figure 5.10 (e) with a LOO-CVE = 0.0634, training MSE = 0.0035 and a test MSE = 0.0691. $DEALKR_F$ had the lowest LOO-CVE of all methods.Table 5.2 contains a summary of the results for S4G1_100.

### 5.1.3 S4G1_200 Data Set

Data set S4G1_200 was generated just as S4G1_50 and S4G1_100.

5.1.3.1 *LOO-CVE Based Optimal Solution*

The optimal bandwidth parameters $h_1,h_2$ = (0.0012, 0.2477) were selected by minimizing the LOO-CVE surface shown in Figure 5.11 (b). The resulting LOO-CVE was 0.0302. Figure 5.11 (c) show the Gaussian kernel produced by these bandwidths and the resulting function approximation which had a test MSE = 0.0271 is presented in Figure 5.11 (d).

5.1.3.2 *Global Bandwidth Results*

The best global bandwidth parameter was selected by minimizing the LOO-CVE values computed for 200 different values of h from between $10^{-4}$ to $10^{3.}$ Figure 5.12 (a) shows the LOO-CVE curve. Notice there are two distinct minima even in this 2-D error surface. A bandwidth of 0.0073 resulted in the minimum LOO-CVE of 0.1786. Figure 5.12 (b) shows the Gaussian kernel for these bandwidths. The resulting function approximation with a training MSE = 0.0117 and a test MSE = 0.1724 is shown in Figure 5.12 (c) Notice the affect of over fitting the training data.

5.1.3.3 *ModelM and the CGLS Results*

ModelM was used to survey the LOO-CVE surface and search for local minima. Figure 5.13 (a) shows the ModelM results superimposed on top of the optimal LOO-CVE surface for S4G1_200. The black dot, indicates the "informed" initial guess. Notice that it did not converge to the optimal solution. It gets trapped in a local minimum. Also notice the multiple local minima indicated by the red x's. Figure 5.13 (b) shows the Gaussian kernel for the bandwidths obtained using the initial guess. The standard deviations in the smoothing parameter histograms in Figure 5.13 (c) also suggest the presence of local minima. The function approximation shown in Figure 5.13 (d) resulted in a LOO-CVE = 0.1843, a training MSE = 0.1599 and a test MSE = 0.1874.

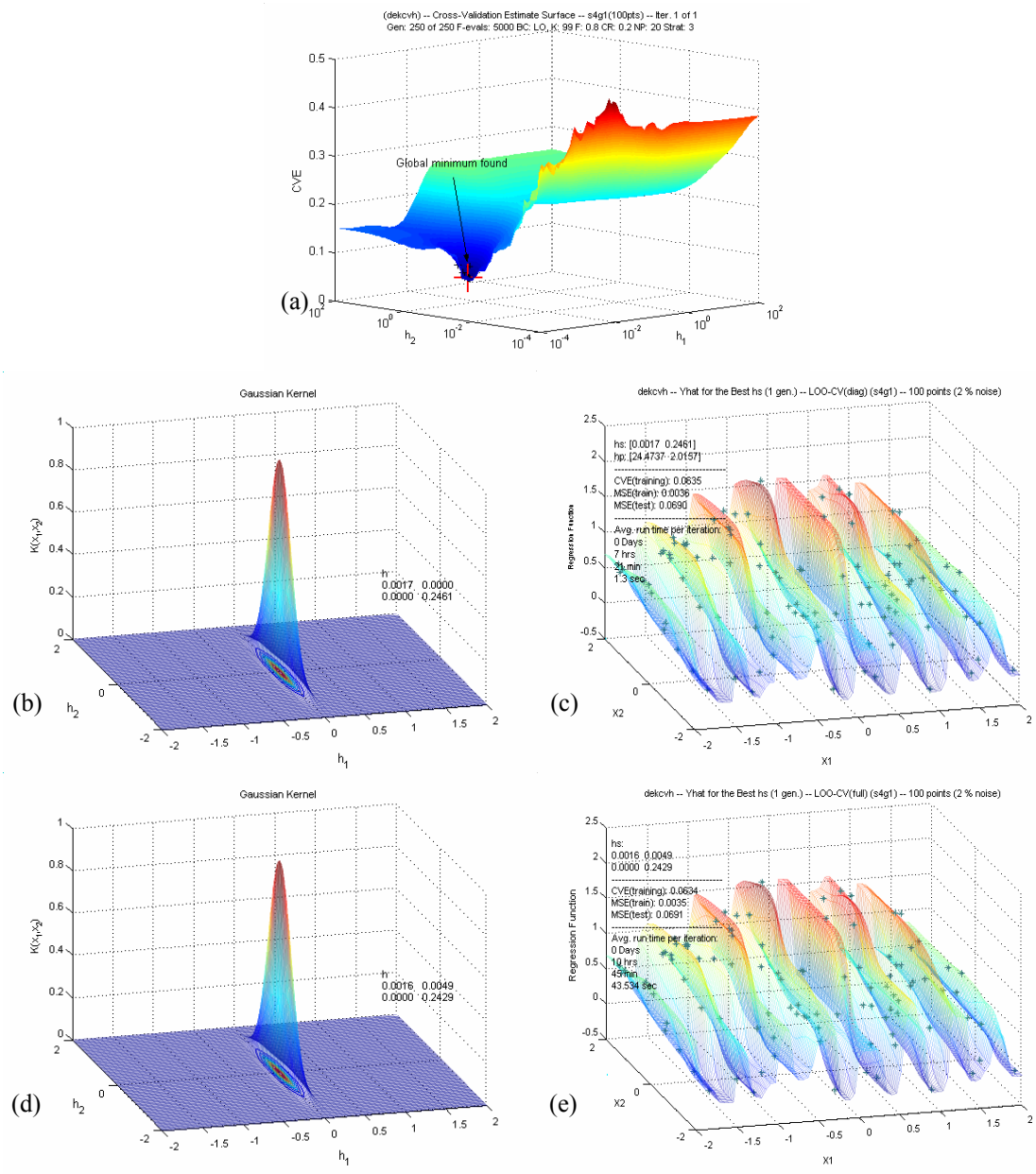



Figure 5.10 DEALKR results for S4G1_100: (a) DEALKRD 250 generation evolutionary history superimposed on the optimal LOO-CVE surface for S4G1_100. The black +'s indicate various "best members" throughout the 250 generation history. The red "+" indicates the final "best member" at the end of 250 generations. Note it is the global minimum. (b) Gaussian kernel for bandwidths obtained by $DEALKR_D$. (c) Resulting function approximation using $DEALKR_D$ bandwidths (d) Gaussian kernel for full bandwidth matrix obtained by $DEALKR_F$. (e) Resulting function approximation using full bandwidth matrix.

Table 5.2 Summary of results for S4G1_100 tests for local minima with rankings according to result categories: LOO-CVE., MSE(Train) and MSE(Test).

| Method | Data Set | S4G1_100 | Ranking |
|---|---|---|---|
| **LOO-CVE Optimal** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0016 & 0 \\ 0 & 0.2477 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0636 | 3rd |
| | *MSE(Train):* | | |
| | *MSE(Test):* | 0.0693 | 3rd |
| **Global** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.3037 & 0 \\ 0 & 0.3037 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1965 | 5th |
| | *MSE(Train):* | 0.1634 | 4th |
| | *MSE(Test):* | 0.2041 | 4th |
| **CGLS 'std(x)'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.8892 & 0 \\ 0 & 0.2483 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1962 | 4th |
| | *MSE(Train):* | 0.1721 | 5th |
| | *MSE(Test):* | 0.2087 | 5th |
| **CGLS 'best'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0017 & 0 \\ 0 & 0.2470 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0635 | 2nd |
| | *MSE(Train):* | 0.0036 | 3rd |
| | *MSE(Test):* | 0.0690 | 1st |
| **CGLS 'worst'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0003 & 0 \\ 0 & 0.0000 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.4024 | 6th |
| | *MSE(Train):* | 0.0001 | 1st |
| | *MSE(Test):* | 0.3712 | 6th |
| **DEALKR$_D$** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0017 & 0 \\ 0 & 0.2461 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0635 | 2nd |
| | *MSE(Train):* | 0.0036 | 3rd |
| | *MSE(Test):* | 0.0690 | 1st |
| **DEALKR$_F$** | $\begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0016 & 0.0049 \\ 0.0000 & 0.2429 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0634 | 1st |
| | *MSE(Train):* | 0.0035 | 2nd |
| | *MSE(Test):* | 0.0691 | 2nd |

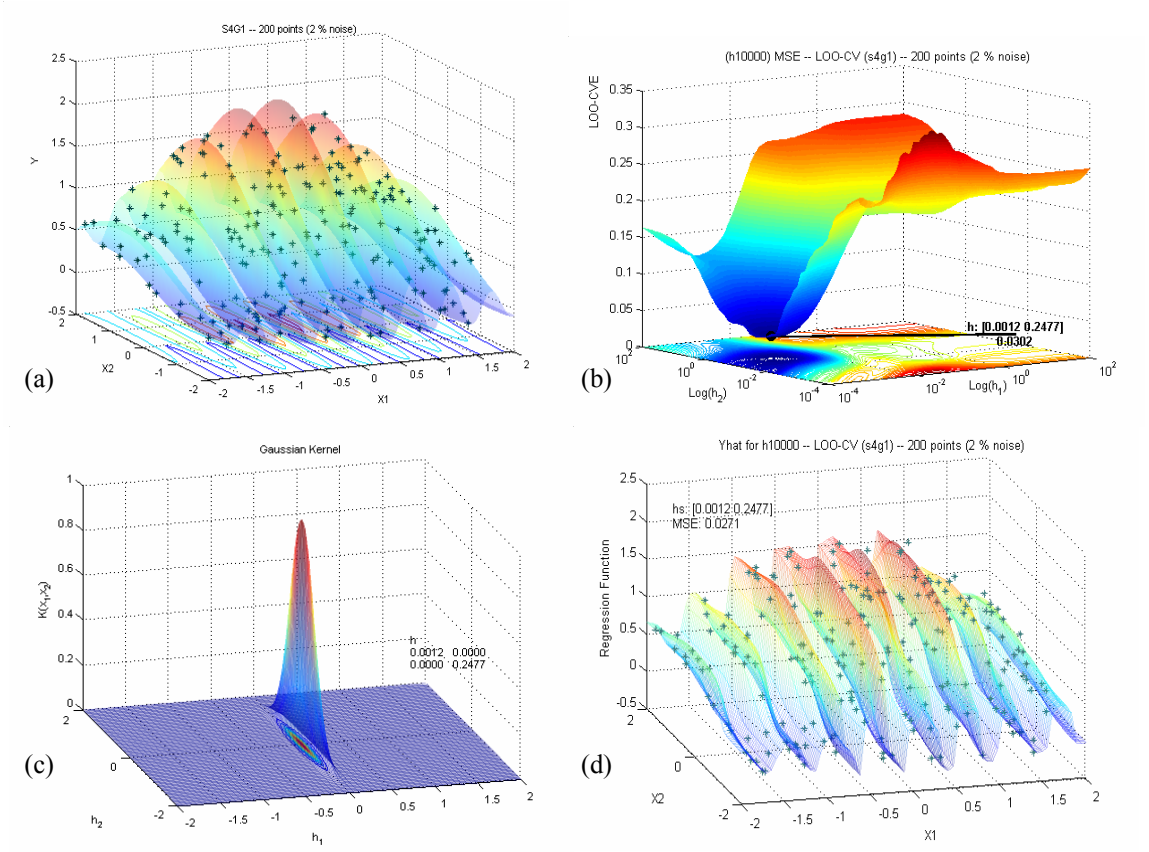



Figure 5.11 S4G1_200 Data Set: (a) Asterisks are training data, surface test data (a) LOO-CVE surface. (b) Gaussian kernel produced by optimal LOO-CVE bandwidths $h_1,h_2$ = (0.0012, 0.2477). (c) Function approximation using Gaussian kernel with optimal bandwidths with LOO-CVEE = 0.0302.

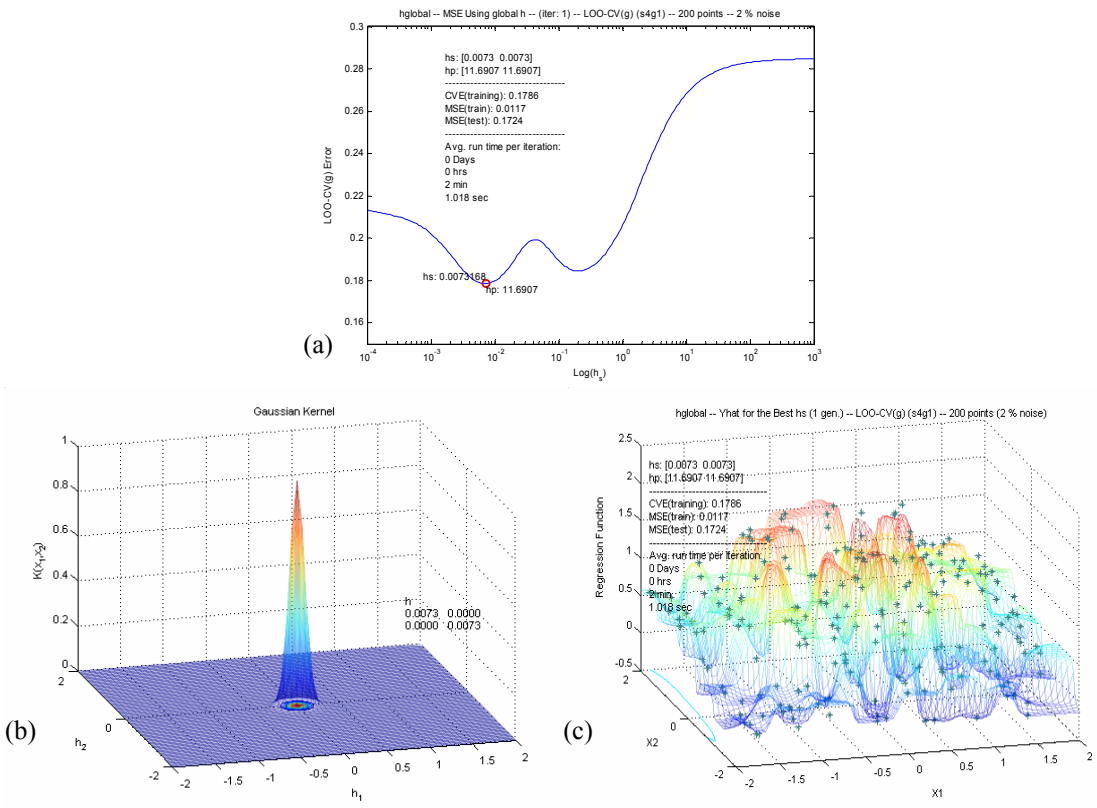



Figure 5.12 Global bandwidth solution for S4G1_200: (a) LOO-CVE plot. Notice the two distinct local minima. (b) Gaussian kernel shape for global bandwidth (0.0073). (c) Function approximation using the global bandwidth LOO-CVE = 0.1786.

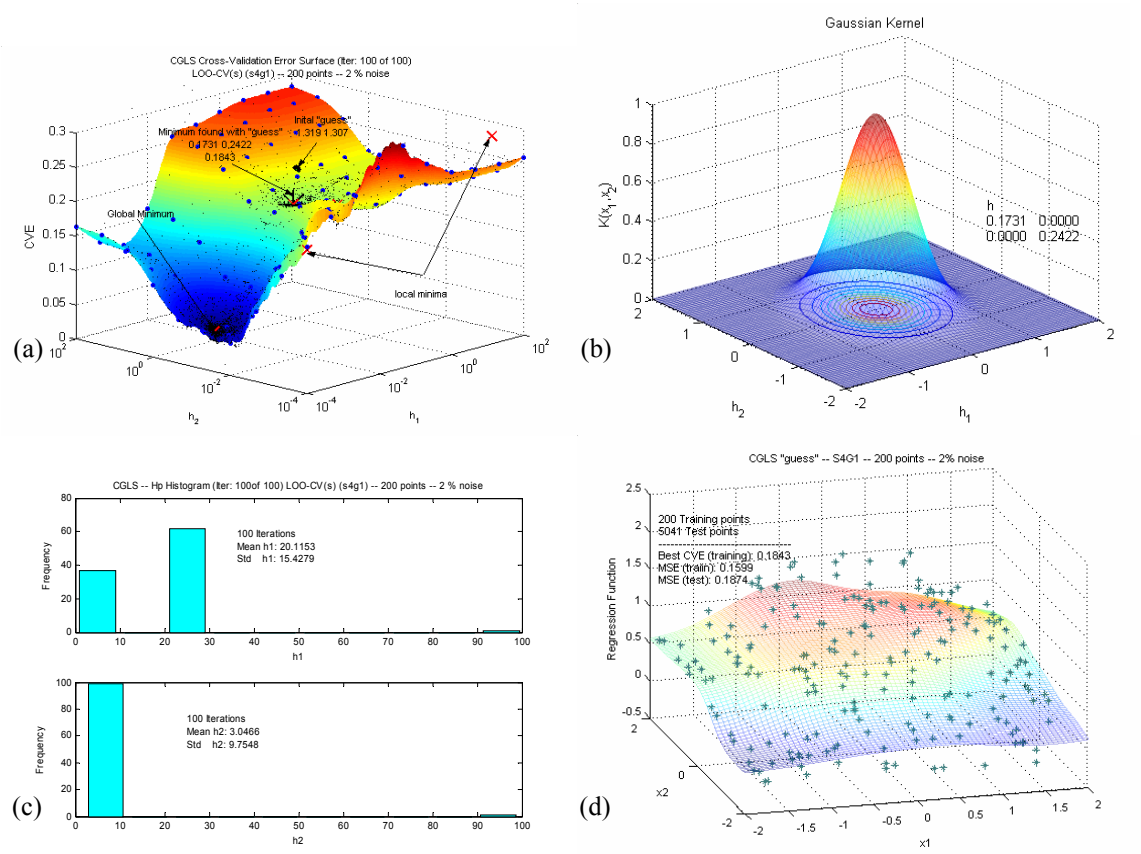



Figure 5.13 ModelM results for the S4G1_200 data set: (a) ModelM results superimposed over the optimal LOO-CVE surface. The blue dots indicate a starting point for each run. The red X's indicate a point of convergence for each run. The green dot indicates the "initial guess" based on std(x), and the green asterisk indicates the final minimum obtained using the initial guess. Note the initial guess did not converge to the global minimum. (b) Gaussian kernel for bandwidths obtained using $std(\mathbf{x})$ as an initial guess. (c) Smoothing parameter histograms from the 100 CGLS runs, with mean and standard deviations for each smoothing parameter. (d) Function approximation produced with results obtained by the initial guess.

The Gaussian kernel produced by the best bandwidths obtained by ModelM is shown in Figure 5.14 (a) Figure 5.14 (b) shows the resulting function approximation produced by best bandwidths which had a LOO-CVE = 0.0302, a training MSE = 0.0042 and a test MSE = 0.0274 Figure 5.14 (c) shows the Gaussian kernel for the worst bandwidths obtained by ModelM and Figure 5.14 (d) shows the function approximation used the worst bandwidths parameters with a LOO-CVE = 0.2075, a training MSE = 0.0001 and a test MSE = 0.1978. Once again notice that the training data is over fit.

#### 5.1.3.4 *DEALKR$_D$ Results*

The evolution history for a 250 generation run of DEALKR$_D$ is superimposed on top of the LOO-CVE surface for S4G1_200 in Figure 5.15 (a). DE again found the global minimum. Figure 5.15 (b) contains the Gaussian kernel for the bandwidths obtained by DEALKR$_D$. The function approximation based on these bandwidths is presented in Figure 5.15 (c) It had a LOO-CVE = 0.0302, a training MSE = 0.0042 and a test MSE, = 0.0274.

#### 5.1.3.5 *DEALKR$_F$ Results*

The Gaussian kernel produced by the full bandwidth matrix obtained by DEALKR$_F$ is shown in Figure 5.15 (d). The resulting function approximation shown in Figure 5.15 (a) had a LOO-CVE = 0.0295, training MSE = 0.0042 and a test MSE = 0.027.

The results for S4G1_200 are summarized in Table 5.3.

### 5.1.4 S4G1_500 Data Set

The S4G1_500 data set shown in Figure 5.16 (a) was generated just like the previous version of S4G1, but with 500 input-output pairs.

#### 5.1.4.1 *LOO-CVE Based Optimal Solution*

The optimal bandwidth parameters were selected by minimizing the LOO-CVE surface for the S4G1_500 training data set. The LOO-CVE reached a minimum at $h_1,h_2$ = (0.0011, 0.1630) with a LOO-CVE = 0.0097 and is shown in Figure 5.16 (b). The Gaussian kernel for the LOO-CVE optimal bandwidths is shown in Figure 5.16 (c) and the resulting function approximation with a test MSE = 0.0097 is presented in Figure 5.16 (d).

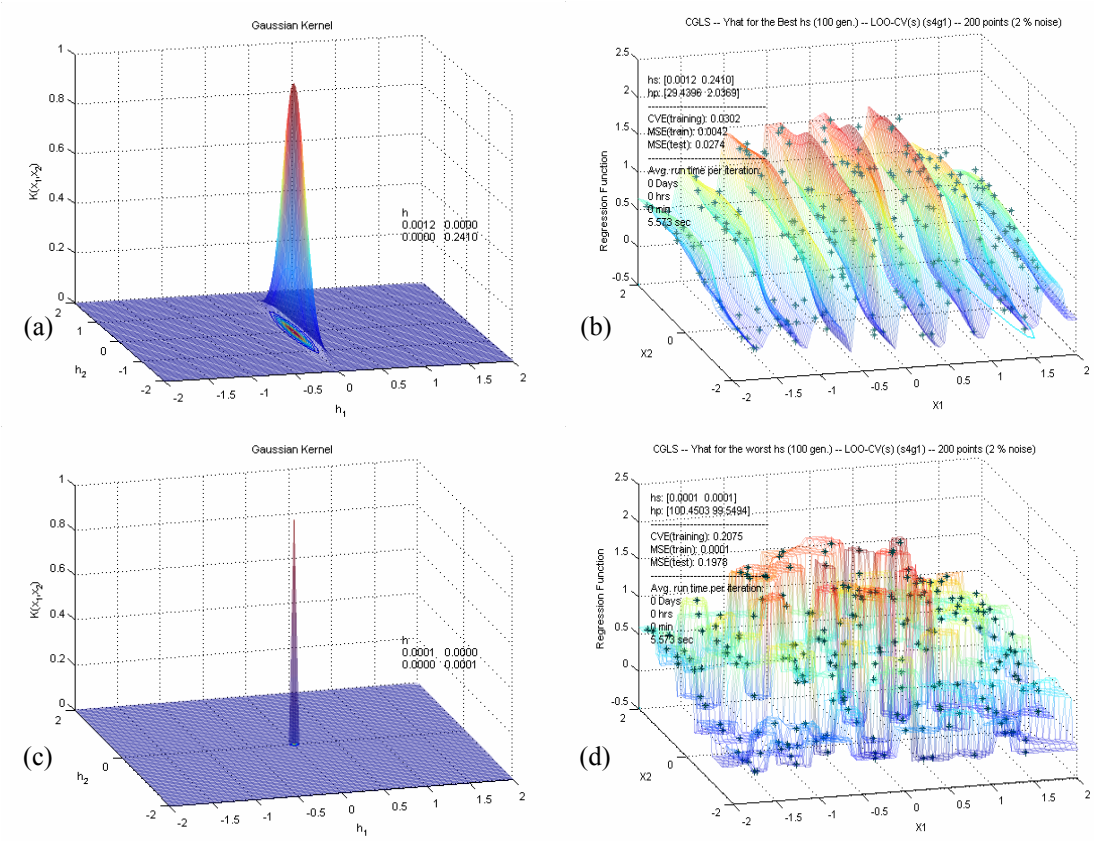
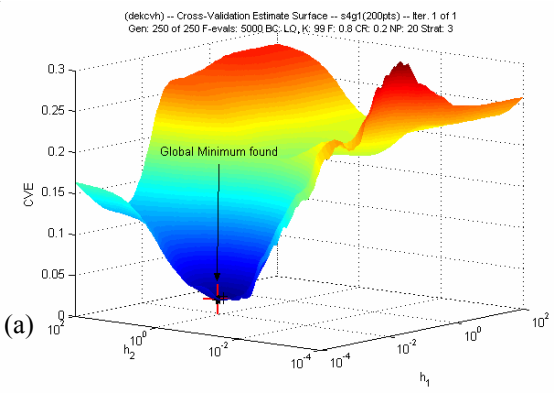
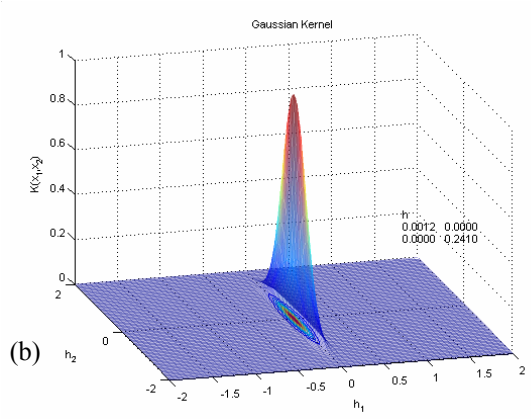


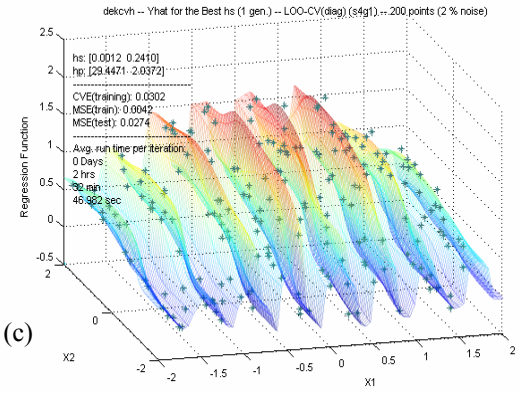

Figure 5.14 Gaussian kernels and function approximations obtained by ModelM for S4G1_200: (a) Gaussian kernel for best bandwidths obtained by ModelM. (b) Function approximation using the best bandwidths obtained by ModelM. (c) Gaussian kernel for worst values of *h*. (d) Function approximation using the worst bandwidths obtained by ModelM.


(dekcvh) -- Cross-Validation Estimate Surface -- s4g1(200pts) -- Iter. 1 of 1
Gen: 250 of 250 F-evals: 5000 BC: LO, K: 99 F: 0.8 CR: 0.2 NP: 20 Strat: 3
Global Minimum found
CV/E
h2
h1

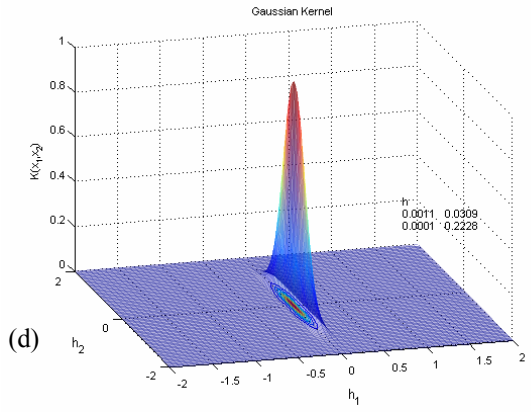
(a)


Gaussian Kernel
K(x1,x2)
h
0.0012 0.0000
0.0000 0.2410
h2
h1

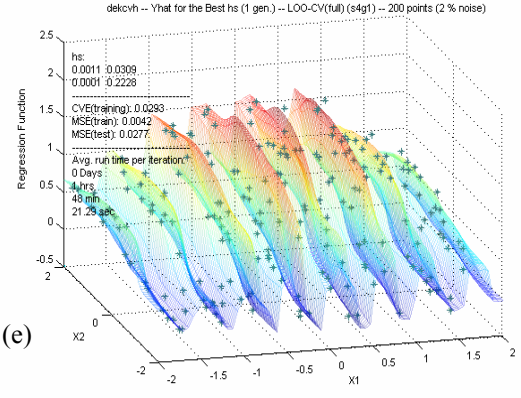
(b)


dekcvh -- Yhat for the Best hs (1 gen.) -- LOO-CV(diag) (s4g1) -- 200 points (2 % noise)
hs: [0.0012 0.2410]
hp: [29.4471 2.0372]
CVE(training): 0.0302
MSE(train): 0.0042
MSE(test): 0.0274
Avg. run time per iteration:
0 Days
2 hrs
32 min
46.952 sec
Regression Function
X2
X1

(c)


Gaussian Kernel
K(x1,x2)
h
0.0011 0.0309
0.0001 0.2228
h2
h1

(d)


dekcvh -- Yhat for the Best hs (1 gen.) -- LOO-CV(full) (s4g1) -- 200 points (2 % noise)
hs:
0.0011 0.0309
0.0001 0.2228
CVE(training): 0.0293
MSE(train): 0.0042
MSE(test): 0.0277
Avg. run time per iteration:
0 Days
4 hrs
48 min
21.29 sec
Regression Function
X2
X1

(e)

Figure 5.15 DEALKR results for S4G1_200: (a) $DEALKR_D$ 250 generation evolutionary history superimposed on the optimal LOO-CVE surface for S4G1_200. The black +'s indicate various "best members" throughout the 250 generation history. The red "+" indicates the final "best member" at the end of 250 generations. Note it is the global minimum. (b) Gaussian kernel for bandwidths obtained by $DEALKR_D$. (c) Resulting function approximation using $DEALKR_D$ bandwidths (d) Gaussian kernel for full bandwidth matrix obtained by $DEALKR_F$. (e) Resulting function approximation using full bandwidth matrix.

Table 5.3 Summary of results for S4G1_200 tests for local minima with rankings according to result categories: LOO-CVE., MSE(Train) and MSE(Test).

| Method | Data Set | S4G1_200 | Ranking |
|---|---|---|---|
| **LOO-CVE Optimal** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0012 & 0 \\ 0 & 0.2477 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0302 | 2nd |
| | *MSE(Train):* | | |
| | *MSE(Test):* | 0.0271 | 1st |
| **Global** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0073 & 0 \\ 0 & 0.0073 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1786 | 3rd |
| | *MSE(Train):* | 0.0117 | 5th |
| | *MSE(Test):* | 0.1724 | 4th |
| **CGLS 'std(x)'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.1731 & 0 \\ 0 & 0.2422 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1843 | 4th |
| | *MSE(Train):* | 0.1599 | 3rd |
| | *MSE(Test):* | 0.1874 | 5th |
| **CGLS 'best'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0012 & 0 \\ 0 & 0.2410 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0302 | 2nd |
| | *MSE(Train):* | 0.0042 | 2nd |
| | *MSE(Test):* | 0.0274 | 2nd |
| **CGLS 'worst'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0001 & 0 \\ 0 & 0.0001 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.2075 | 5th |
| | *MSE(Train):* | 0.0001 | 1st |
| | *MSE(Test):* | 0.1978 | 6th |
| **$DEALKR_D$** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0012 & 0 \\ 0 & 0.2410 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0302 | 2nd |
| | *MSE(Train):* | 0.0042 | 2nd |
| | *MSE(Test):* | 0.0274 | 2nd |
| **$DEALKR_F$** | $\begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0011 & 0.0309 \\ 0.0001 & 0.2228 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0293 | 1st |

| | | | |
|---|---|---|---|
| | *MSE(Train):* | 0.0042 | 2nd |
| | *MSE(Test):* | 0.0277 | 3rd |

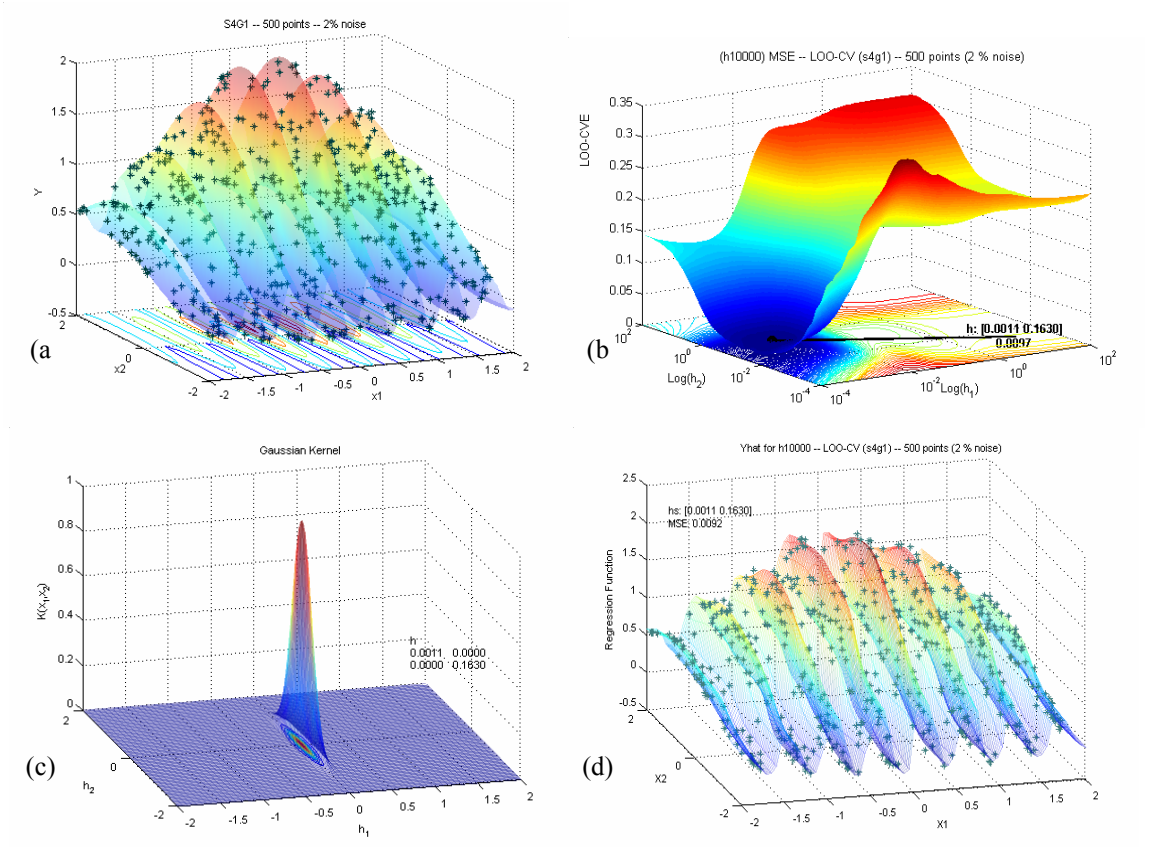



Figure 5.16 S4G1_500 Data Set: (a) Asterisks are training data; surface plot is test data. (b) LOO-CVE surface. (c) Gaussian kernel produced by optimal LOO-CVE bandwidths $h_1,h_2$ = (0.0011, 0.1630). (d) Function approximation using Gaussian kernel with optimal bandwidths with LOO-CVE = 0.0097.

5.1.4.2 *Global Bandwidth Results*

The best global bandwidth parameter was selected by computing the LOO-CVE for 200 different values of h from $10^{-4}$ to $10^{3}$. Figure 5.17 (a) presents a plot of LOO-CVE versus bandwidth for and it still has two minima. The global minimum of 0.0981 was for $h = 0.0067$. The function approximation produced using this global bandwidth parameter resulted in a training MSE = 0.0149 and a test MSE = 0.0776. Figure 5.17 (b) shows the Gaussian kernel and Figure 5.17 (c) the resulting function approximation.

5.1.4.3 *ModelM and CGLS Results*

ModelM was used to survey the LOO-CVE surface for S4G1_500 looking for local minima. Figure 5.18 (a) presents the ModelM results superimposed on the LOO-CVE surface for S4G1_500. The convergence path that started at the black dot, which is the value of the "informed" initial guess, did not result in finding the optimal solution but got trapped in a local minimum. Again note the multiple local minima represented by the red x's. Figure 5.18 (b) show the Gaussian kernel for the bandwidths obtained using the initial guess. The smoothing parameter spectrum and standard deviations is plotted in Figure 5.18 (c) The resulting function approximation shown in Figure 5.18 (d) had a LOO-CVE = 0.175, a training MSE = 0.1625 and a test MSE = 0.1818.

The Gaussian kernel for the best bandwidth parameters obtained by ModelM, $h_1,h_2$ = (0.0010, 0.1539), is shown in Figure 5.19 (a) and the resulting function approximation with a LOO-CVE = 0.0096, a training MSE = 0.0032 and a test MSE = 0.0091 is presented in Figure 5.19 (b).

The Gaussian kernel for the worst bandwidth parameters obtained by ModelM, $h_1,h_2$ = (43M, 0.0067), is shown in Figure 5.19 (c) and the resulting function approximation with a LOO-CVE = 0.2045, a training MSE = 0.1890 and a test MSE = 0.22 is shown in Figure 5.19 (d ).

#### 5.1.4.4 *DEALKR$_D$ Results*

The evolutionary history for a 100 generation run of DEALKR$_D$ is shown superimposed on the LOO-CVE surface for S4G1_500 in Figure 5.20 (a). Once again it converged to the global minimum. Figure 5.20 (b) shows the Gaussian kernel using the bandwidths obtained by DEALKR$_D$, $h_1,h_2$ = (0.0019, 0.1548). The resulting function approximation shown in Figure 5.20 (c) had a LOO-CVE = 0.0096, a training MSE = 0.0032 and a test MSE = 0.0091.

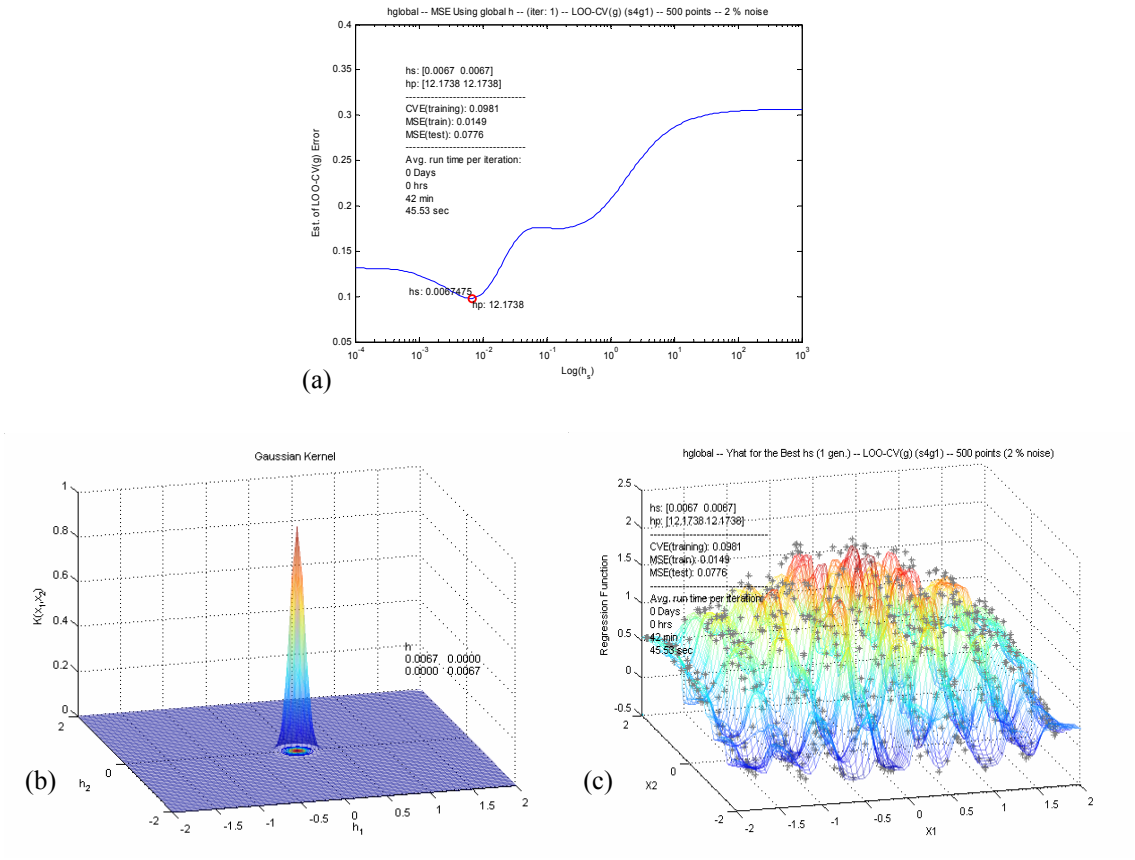



Figure 5.17 Global bandwidth solution for S4G1_500: (a) LOO-CVE plot. Notice there are still two local minima. (b) Gaussian kernel shape for global bandwidth (0.0067 ). (c) Function approximation using the global bandwidth LOO-CVE = 0.0981.

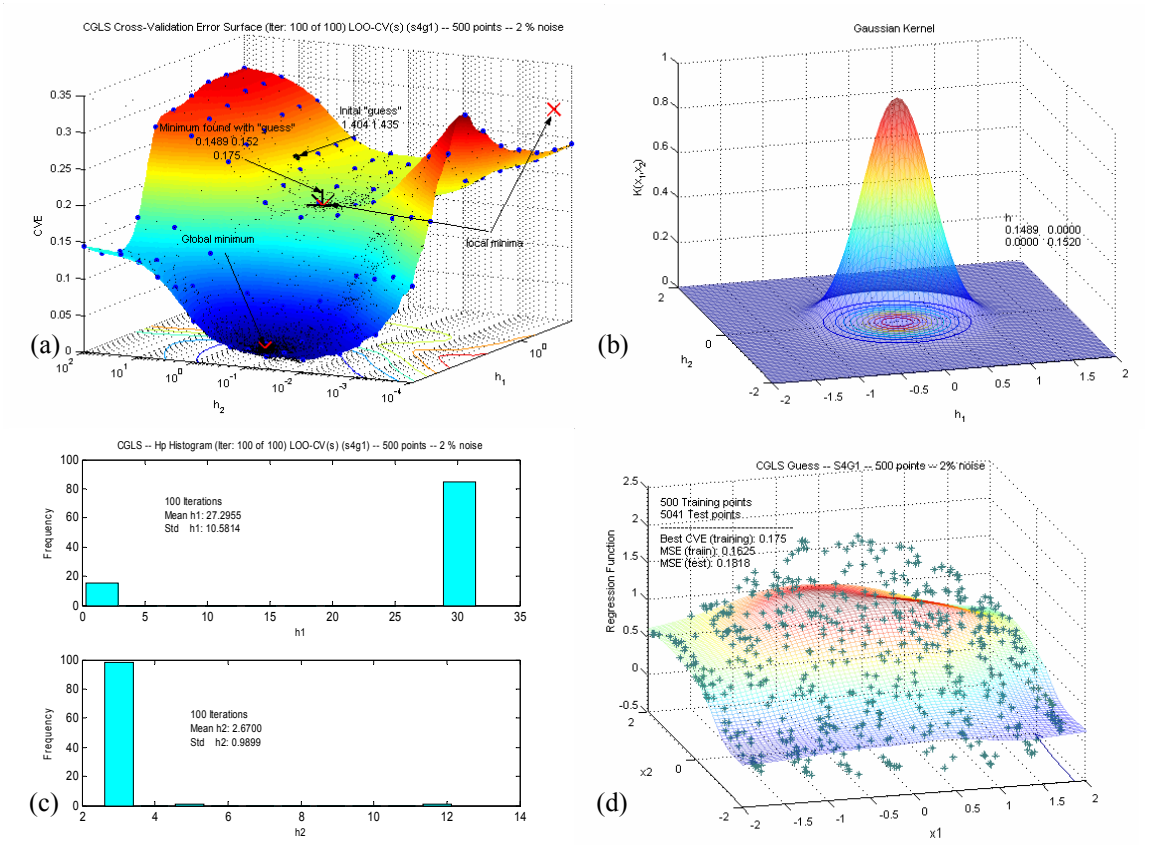



Figure 5.18 ModelM results for the S4G1_500 data set: (a) ModelM results superimposed over the optimal LOO-CVE surface. The blue dots indicate a starting point for each run. The red X's indicate a point of convergence for each run. The green dot indicates the "initial guess" based on std(x), and the green asterisk indicates the final minimum obtained using the initial guess. Note the initial guess still did not converge to the global minimum. (b) Gaussian kernel for bandwidths obtained using $std(\mathbf{x})$ as an initial guess. (c) Smoothing parameter histograms from the 100 CGLS runs, with mean and standard deviations for each smoothing parameter. (d) Function approximation produced with results obtained by the initial guess.

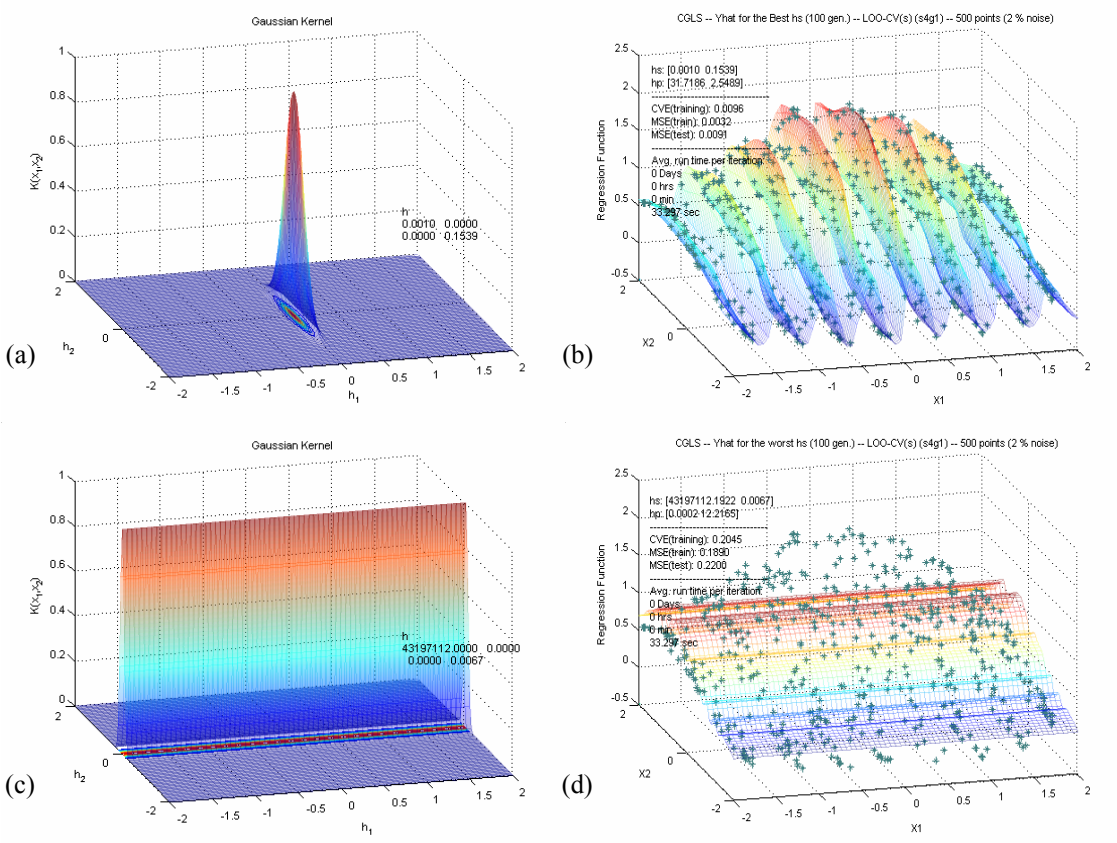



Figure 5.19 Gaussian kernels and function approximations obtained by ModelM for S4G1_500: (a) Gaussian kernel for best bandwidths obtained by ModelM. (b) Function approximation using the best bandwidths obtained by ModelM. (c) Gaussian kernel for worst values of *h*. (d) Function approximation using the worst bandwidths obtained by ModelM.

5.1.4.5 *$DEALKR_F$ Results*

Figure 5.20 (d) shows the Gaussian kernel produced by the full bandwidth matrix obtained by $DEALKR_F$ after 160 generations. The resulting function approximation, shown in Figure 5.20 (e), had a LOO-CVE = 0.0094, training MSE = 0.0029 and a test MSE of 0.0091.

Table 5.4 provides a summary of the results for S4G1_500 test data. $DEALKR_D$ and $DEALKR_F$ outperformed Global for all case and CGLS for all cases when the initial guess was used as the starting point.

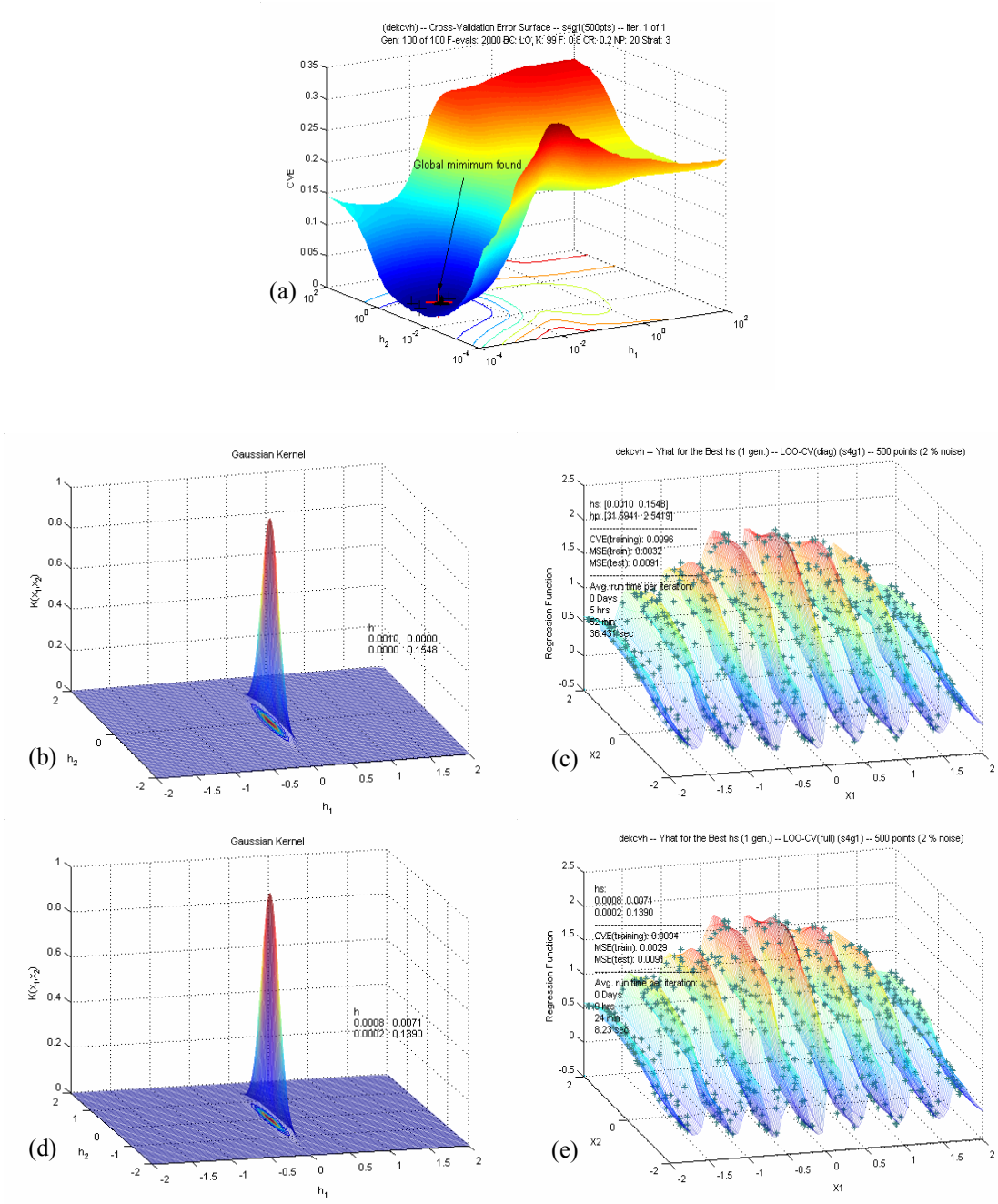



Figure 5.20 DEALKR results for S4G1_500: (a) $DEALKR_D$ 100 generation evolutionary history superimposed on the optimal LOO-CVE surface for S4G1_500. The black +'s indicate various "best members" throughout the 100 generation history. The red "+" indicates the final "best member" at the end of 250 generations. Note it is the global minimum. (b) Gaussian kernel for bandwidths obtained by $DEALKR_D$. (c) Resulting function approximation using $DEALKR_D$ bandwidths (d) Gaussian kernel for full bandwidth matrix obtained by $DEALKR_F$. (e) Resulting function approximation using full bandwidth matrix.

Table 5.4 Summary of results for S4G1_500 tests for local minima with rankings according to result categories: LOO-CVE., MSE(Train) and MSE(Test).

| Method | Data Set | S4G1_500 | Ranking |
|---|---|---|---|
| **LOO-CVE Optimal** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0011 & 0 \\ 0 & 0.1630 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0097 | 3rd |
| | *MSE(Train):* | | |
| | *MSE(Test):* | 0.0092 | 2nd |
| **Global** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0067 & 0 \\ 0 & 0.0067 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0981 | 4th |
| | *MSE(Train):* | 0.0149 | 3rd |
| | *MSE(Test):* | 0.0776 | 3rd |
| **CGLS 'std(x)'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.1489 & 0 \\ 0 & 0.1520 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1750 | 5th |
| | *MSE(Train):* | 0.1625 | 4th |
| | *MSE(Test):* | 0.1818 | 4th |
| **CGLS 'best'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0019 & 0 \\ 0 & 0.1539 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0096 | 2nd |
| | *MSE(Train):* | 0.0032 | 2nd |
| | *MSE(Test):* | 0.0091 | 1st |
| **CGLS 'worst'** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 43{,}197{,}112 & 0 \\ 0 & 0.0067 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.2045 | 6th |
| | *MSE(Train):* | 0.1890 | 5th |
| | *MSE(Test):* | 0.2200 | 5th |
| **DEALKR$_D$** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0019 & 0 \\ 0 & 0.1548 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0096 | 2nd |
| | *MSE(Train):* | 0.0032 | 2nd |
| | *MSE(Test):* | 0.0091 | 1st |
| **DEALKR$_F$** | $\begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0008 & 0.0071 \\ 0.0002 & 0.1390 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0094 | 1st |
| | *MSE(Train):* | 0.0029 | 2nd |
| | *MSE(Test):* | 0.0091 | 1st |

Chapter

6

# Case Studies

In this chapter the effectiveness of DEALRR, DELLKR$_D$ and DELLKR$_F$ are demonstrated for artificial and real world data sets. The LKR results for both the diagonal, $\mathbf{H} \in D$, and full, $\mathbf{H} \in F$, bandwidth matrix and the full bandwidth matrix obtained by DE, designated DEALKR$_D$ and DEALKR$_F$ respectively, are compared with results using a global bandwidth, $\mathbf{H} \in S$, and LKR with diagonal bandwidth parameters obtained using the conjugate-gradient + line-search (CGLS) technique described in section 3.3.1.

Section 6.1 provides a brief introduction. Section 6.2 describes the data sets. Section 6.3 presents the results for three artificial data sets with two input variables, Banana3, Bumps1, and Peaks1. Section 6.4 describes the results of DEALKR$_D$ and DEALKR$_F$ for Cascade1, an artificial data set of four input variables and a single output target variable, comparing them against global and CGLS results. The use of DEALKR for input variable selection is demonstrated in section 6.5. It is shown to accurately select relevant input variables and predict the output even when the input data is dominated by noisy or irrelevant input variables. Results are presented for Cascade18 (four relevant and four irrelevant inputs), G10D1 (five relevant and five irrelevant inputs), S4G1u10 (two relevant and eight irrelevant inputs), and S4G1u20 (two relevant and eight-teen irrelevant inputs). Section 6.6 looks at the performance of DEALRR and DEALKR when applied to real world data sets and compares them to Ordinary least-squares, (OLS) principle component regression (PCR), and ordinary ridge regression (ORR).

## 6.1 Introduction

Two different types of data sets are used to demonstrate the capabilities of the DEALRR and DEALKR$_D$ and DEALKR$_F$, one set is artificial and the other is obtained from real world measurements. There are advantages and disadvantages to both types of data. An advantage of using artificial data is the "true" answer is known and thus an accurate assessment of performance can be made. The disadvantage of using

artificial data is the results obtained may not accurately indicate performance of the technique under consideration in the "real world". The advantage of using real world data sets is they are representative of the "real world". But the disadvantage is the underlying model and even the 'true" answer is seldom if ever known.

What follows is a brief overview of the data sets used.

## 6.2 Data sets

The artificial data sets are described first followed by a brief description of the real world data sets.

Four artificial data sets with two-inputs were created to allow visualization of the functions, their approximations and the LOO-CVE surfaces: Banana3, Bumps1, Peaks1, and S4G1.

Five higher dimensional input data sets were created specifically to evaluate input variable selection: Cascade1_100, G10D1_100, Cascade81_100, S4G1U10_100, and S4G1U20_100. The S4G1_100 data set was altered by including a number of noisy/irrelevant input variables, eight in the case of S4G1u10 and eight-teen for S4G1u20. The irrelevant input values were randomly selected from a uniform random distribution and they do not contribute to the output variable at all. Since the output variable is only a function of two input variables the LOO-CVE surface and function approximation for each technique could be visually compared to the true output, just as we did for the 2D data sets.

All training data sets considered contained 100 input-output pairs. The test data sets contained either 5,000 or 5,041 input-output pairs. The two-input test sets had 5,041 input-output pairs [71x71] because they were created using the mesh grid function in MATLAB to facilitate visualization of the results.

Noise was added to the output vector, **y**, for all training data. For all data sets in this study, except for G10D, 2% noise was added to $y$ using:

$$y\_n = [n * (std(y)/100) * rand(length(y),1)] + y \tag{6.1}$$

where, $n$ = 2 and *rand* is a uniform random distribution between zero and one. For the g10d1 data set Gaussian noise, N(0,1), was added to the output vector.

Comparison of the results can be conducted made on the basis of the training LOO-CVE, the training MSE, and the test MSE. LOO-CVE provides a good indication on the effectiveness of the objective function and

the parameter selection/optimization method, while the training and test MSE can serve as indicators of whether or not the model has been properly regularized.

## 6.3 Artificial data sets with two input variables

This section describes the result for three, two-dimensional, artificial data sets, Banana3_100, Bumps1_100, and Peaks1_100. The results of the function estimations and the LOO-CVE surfaces will be plotted for a visual comparison. After a description of each data set the results will be reported, first the optimal solution obtainable by minimizing the LOO-CVE surface, second, the global bandwidth parameter results, third the ModelM results and last the results obtained for both the diagonal and full bandwidth matrix differential evolution, designated $\text{DEALKR}_\text{D}$ and $\text{DEALKR}_\text{F}$.

### 6.3.1 Banana3_100 Data Set

The Banana3_100 data set is also known as Rosenbrock's Saddle or the second De Jong function [44]. The equation describing it is:

$$y = 100(x_2 - x_1^2)^2 - (1 - x_1)^2, \quad with \ -2 \le x_i \le 2. \tag{6.2}$$

The 100 training points are plotted in Figure 6.1 (a) as asterisks and the 5,041 test points as a mesh plot.

#### 6.3.1.1 *Optimal LOO-CVE Solution*

The optimal solution for the 100 training points is selected by minimizing the LOO-CVE. The LOO-CVE surface for Banana3 is shown in Figure 6.1 (b). The smallest LOO-CVE value occurs at 19,182 for the bandwidths $h_1,h_2$ = (0.0266, 0.1630). The Gaussian kernel produced by these bandwidths is shown in Figure 6.1 (c). The resulting function approximation with a test MSE = 28,463 is shown in Figure 6.6 (a).

#### 6.3.1.2 *Global Bandwidth Results*

The optimal global bandwidth parameter was determined by computing the LOO-CVE for 200 different values of $h$ from $10^{-4}$ to $10^{3}$ and selecting the $h$ the produced the minimum LOO-CVE value. Figure 6.2 (a) shows the LOO-CVE plot with a minimum of 32,371 at $h$ = 0.0314. The Gaussian kernel produced by this bandwidth is shown Figure 6.2 (b). The resulting function approximation, shown in Figure 6.6 (b), had test MSE = 49,446 and a training MSE = 3,976.

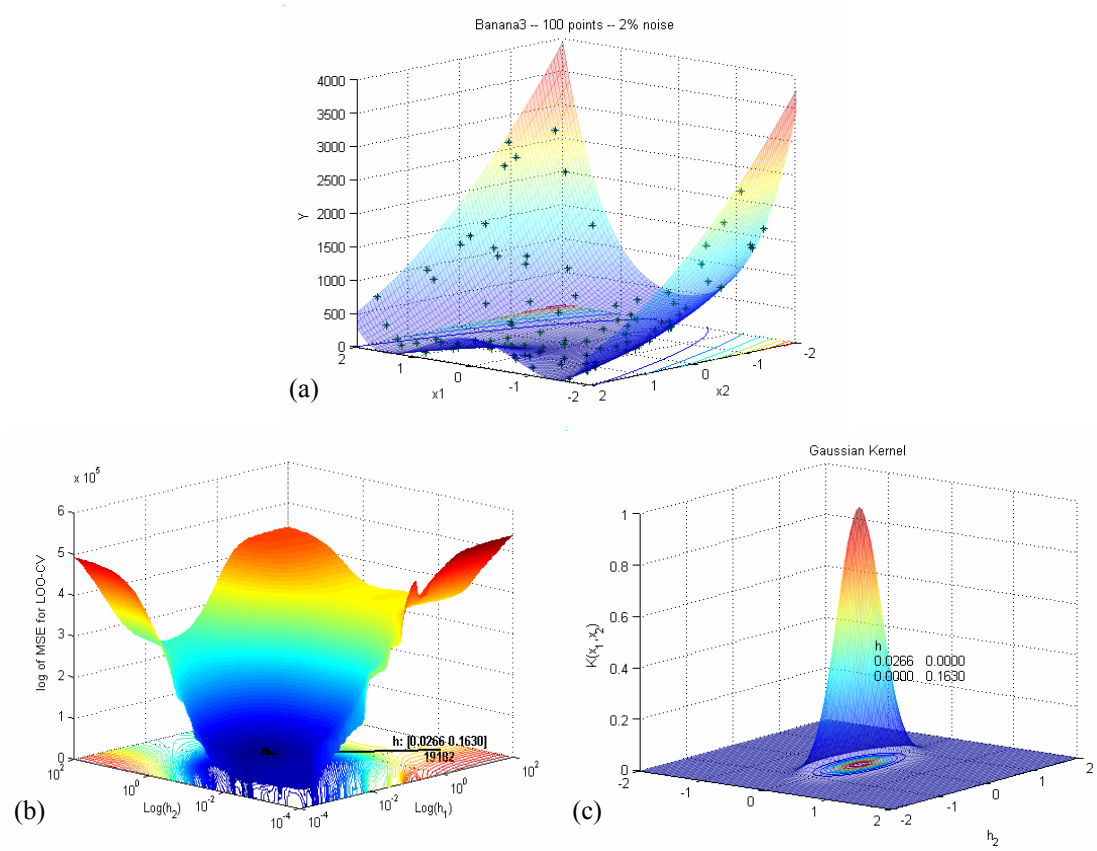



Figure 6.1 Banana3_100 data set: (a) Asterisks are the training data (100 points) and the mesh plot is the test data (5,041 points). (b) LOO-CVE surface computed on a 100x100 grid. Global minimum at (0.0266, 0.1630) with LOO-CVE = 19,182. (c) Gaussian kernel produced by LOO-CVE optimal bandwidths.

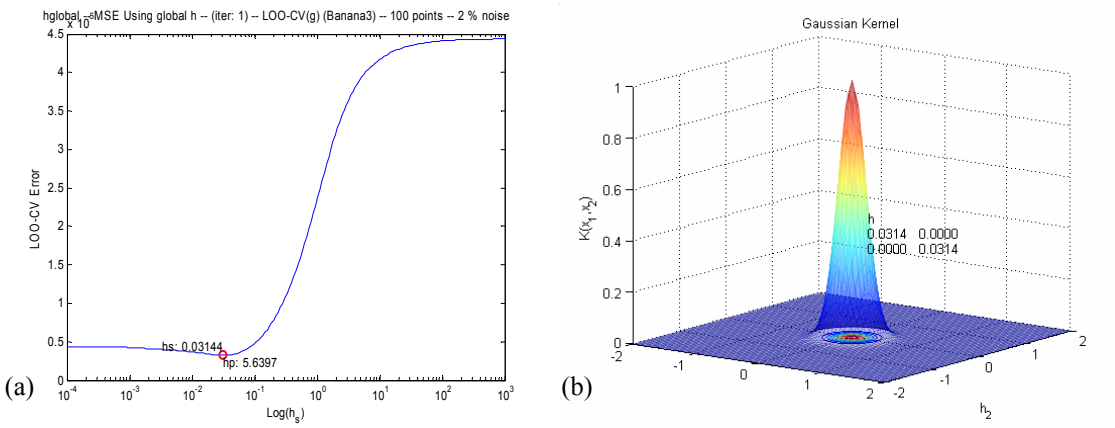



Figure 6.2 Global bandwidth solution for Banana3_100: (a) LOO-CVE plot minimum at (0.0314) with a LOO-CVE = 32,372. (b) Gaussian kernel for optimal global bandwidth.

6.3.1.3 *ModelM and CGLS LKR Results*

ModelM was used with a 10x10 grid of starting points to "map" the LOO-CVE surface and look for local minima as described in section 5.1. Figure 6.3 (a) show the results superimposed over the LOO-CVE surface. The blue dots designate the 100 staring points. The smaller black dots are LOO-CVE values at steps taken along each convergence path. Each red x is a final convergence point. The large black dot is the initial guess based on the standard deviation of the input variables, and the green asterisk is resulting convergence point. The presence of local minima are indicated by the scattered red x's.

The smoothing parameter histogram for Banana3_100 is shown in Figure 6.3 (b). The large standard deviations std($h_1$) = 16.98 and std($h_2$) = 30.07 are an indication that multiple minima are present. The initial guess did converge to the global minima at $h_1,h_2$ = (0.0254, 0.1511), with a LOO-CVE of 19,125, a training MSE = 5,958 and a test MSE = 28,062. The worst set of bandwidths $h_1,h_2$ = (385,352, 0.0218) obtained by ModelM, had a LOO-CVE = 342,088, a training MSE =270,103 and a test MSE = 3,75,461. The Gaussian kernels produced by the best and worst bandwidths are shown in Figure 6.3 (c) and (d). The resulting function approximations for both sets of bandwidths are presented Figure 6.6 (c) and (d).

6.3.1.4 *DEALKR$_D$ Results*

Figure 6.4 (a) shows the results from 100 separate iterations of 100 generations each superimposed over the LOO-CVE surface for Banana3_100. The black +'s indicate the best members of each generation during the evolutionary history of all 100 iterations. The red +'s indicate the best member ever or the point of final convergence for each iteration. There are actually 100 red +'s even though there appear to be only one – they all converged to the LOO-CVE global minimum. The Gaussian kernel produced by the bandwidths $h_1,h_2$ = (0.0254, 0.1511) obtained by DEALKR$_D$ is shown in Figure 6.4 (b). The resulting

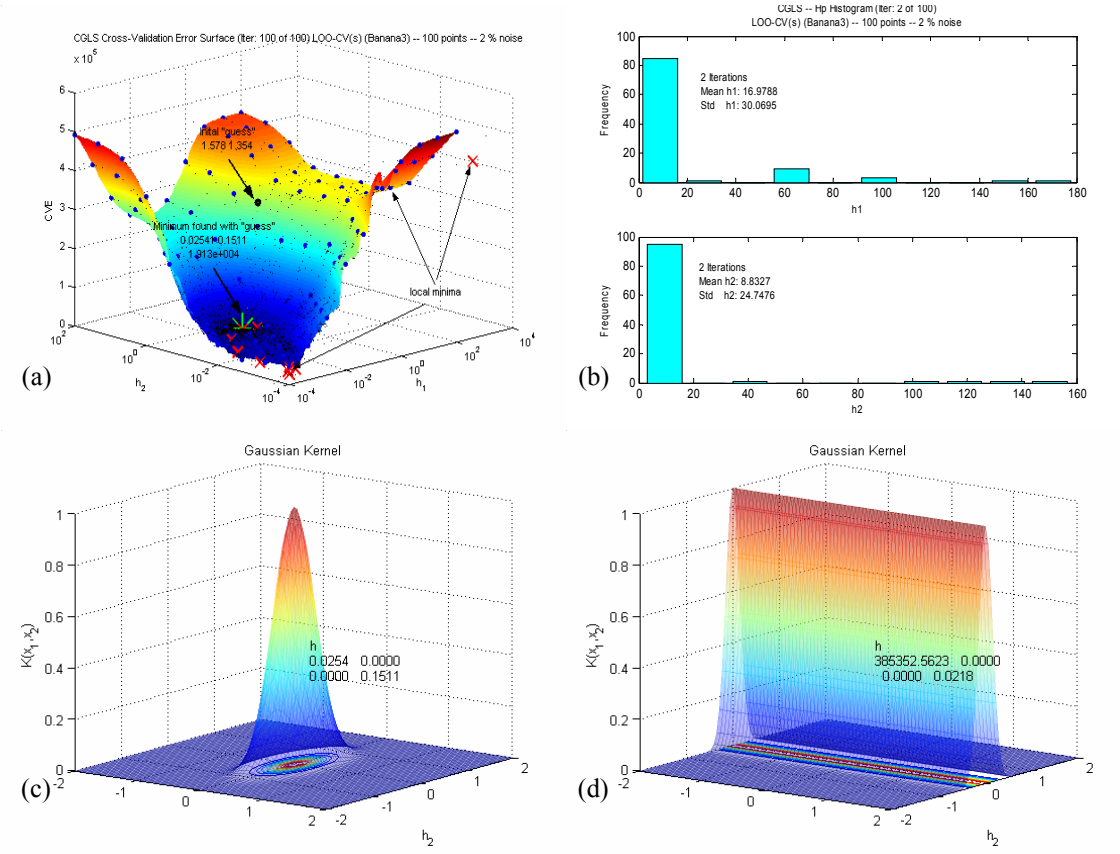



Figure 6.3 ModelM results for the Banana3_100 data set: : (a) ModelM results superimposed over the optimal LOO-CVE surface. The blue dots indicate a starting point for each run. The red X's indicate a point of convergence for each run. The green dot indicates the "initial guess" based on std(x), and the green asterisk indicates the final minimum obtained using the initial guess. Note the initial guess did converge to the global minimum. (b) Smoothing parameter histograms from the 100 CGLS runs, with mean and standard deviations for each smoothing parameter. (c) Gaussian kernel for best bandwidths obtained by ModelM. (d) Gaussian kernel for worst bandwidths obtained by ModelM.

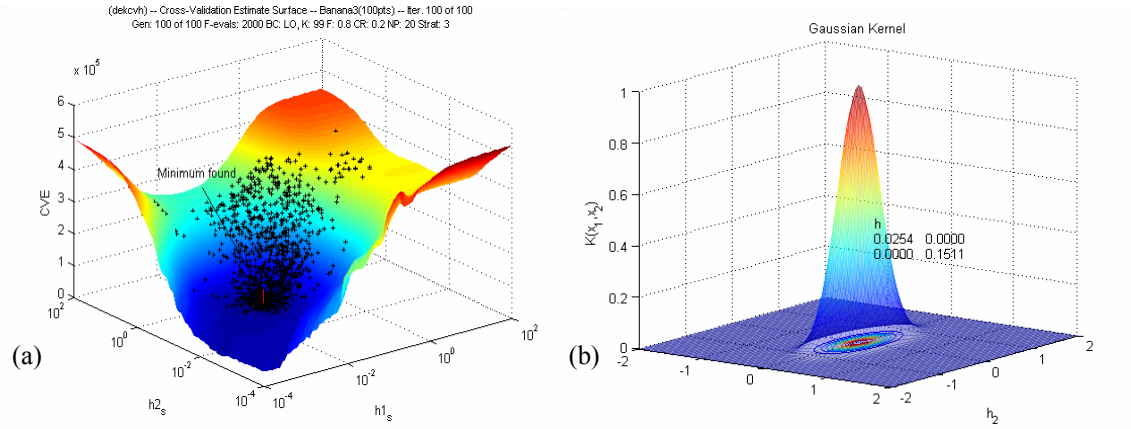



Figure 6.4 DEALKR$_D$ results for Banana3_100: (a) DEALKR$_D$ 100 generation evolutionary history superimposed on the optimal LOO-CVE surface for Banana3_100. The black +'s indicate various "best members" throughout the 100 generation history. The red "+" indicates the final "best member" at the end of 250 generations. Note it is the global minimum. (b) Gaussian kernel for bandwidths obtained by DEALKR$_D$.

function approximation with a LOO-CVE = 19,125, a training MSE = 5,958 and a test MSE = 28,062 is shown in Figure 6.6 (e).

#### 6.3.1.5 *DEALKR$_F$ Results*

The full bandwidth matrix was also optimized by minimizing the LOO-CVE. The smoothing parameter spectrum and Gaussian kernel obtained by DEALKR$_F$ are presented in Figure 6.5 (a) and (b). The resulting function approximation which had a LOO-CVE = 19,046, a training MSE = 5,894, and test MSE = 27,978 is shown in Figure 6.6 (f). Notice it has the lowest LOO-CVE and test MSE of all the approaches.

Table 6.1 provides a summary of the results for Bannana3_100. Note DEALKR$_D$ and DEALKR$_F$ outperformed Global for all case and CGLS for all cases when the initial guess was used as the starting point.

### 6.3.2 Bumps1_100 Data Set

Data Bumps1_100 is a bimodal function described by:

$$y = 0.7 \times e^{(-3((x_1+0.8)^2+8(x_2-0.5)^2))} + e^{(-3((x_1+0.8)^2+8(x_2-0.5)^2))}, \; -2 \le x_1 \le 2, \quad 0 \le x_2 \le 1. \tag{6.3}$$

Figure 6.7 (a) shows the Bumps1_100 test set. The surface plot is of the 5,041 point test set and the training data are shown as asterisks.

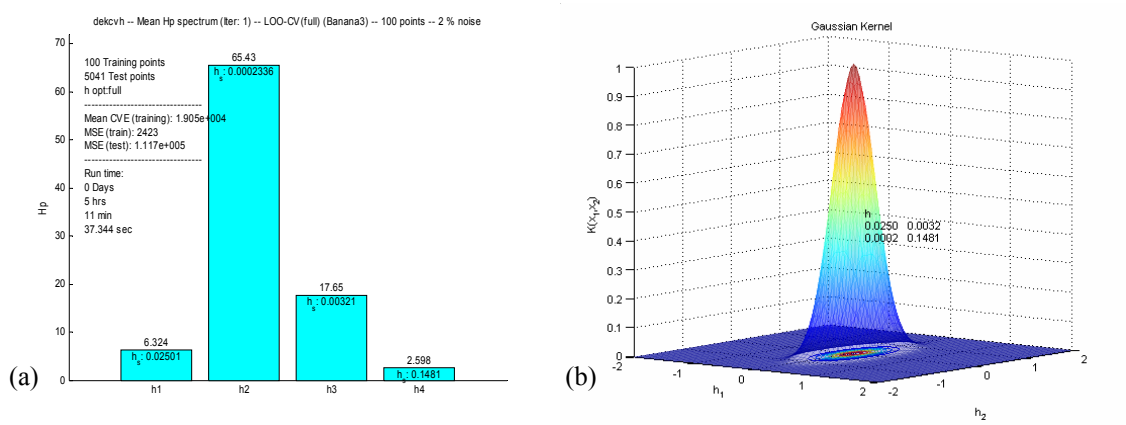



Figure 6.5 DEALKR$_F$ results for Banana3_100: (a) Smoothing parameter spectrum. (b) Gaussian kernel for bandwidths obtained by DEALKR$_F$.

### 6.3.2.1 *Optimal LOO-CVE Solution*

The optimal solution for Bumps1_100 was selected by minimizing the LOO-CVE. The LOO-CVE surface for Bumps1_100 is shown in Figure 6.7 (b). The smallest LOO-CVE value occurs at 0.0035 for the bandwidth parameters $h_1,h_2$ = (0.0175, 0.0016). The function approximation of Bumps1_100 based on these optimal bandwidth parameters results in a test MSE of 0.0038. Figure 6.7 (b) is the Gaussian kernel for the optimal bandwidths. Figure 6.12 (a) shows the function approximation as a mesh plot along with as surface plot of the test data and the training data as asterisks.

### 6.3.2.2 *Global Bandwidth Results*

The optimal global bandwidth parameter for Bumps1_100 was determined by computing the LOO-CVE for 200 different values of $h$ from $10^{-4}$ to $10^{3}$ and selecting the value of $h$ the produced the minimum LOO-CVE value 0.0078. Figure 6.8 (a) shows the plot of LOO-CVE versus bandwidth parameter for Bumps1_100 with a minimum at $h$ = 0.0086. The Gaussian kernel for this bandwidth is shown in Figure 6.8 (b). The function approximation based on the global bandwidth parameter resulted in a training MSE of 0.0011 and a test MSE = 0.0069. Figure 6.12 (b) shows this approximation to Banana3 as a mesh plot along with as surface plot of the test data and the training data as asterisks.

### 6.3.2.3 *ModelM and CGLS LKR Results*

The ModelM survey technique was used with a 10 x 10 grid of starting points to "map" the LOO-CVE surface and look for local minima. Figure 6.9 (a) shows the results superimposed on top of the LOO-CVE surface for Bumps1. The blue dots designate the 100 staring points. The smaller black dots are LOO-CVE values at steps taken along convergence paths. Each red x is a final convergence point. The large black dot

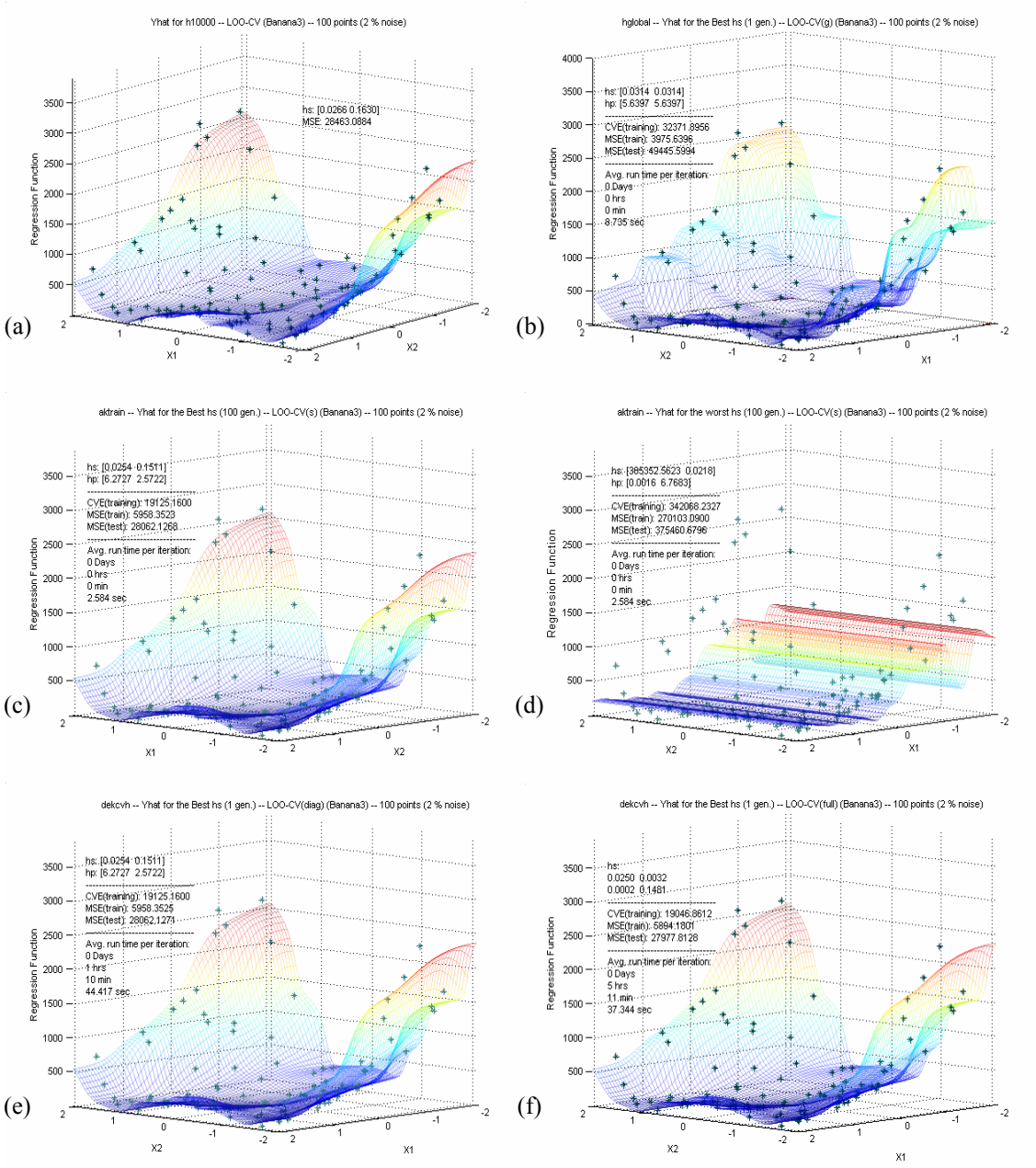



Figure 6.6 Function approximations of Banana3_100: (a) Using optimal LOO-CVE bandwidths. (b) Using the optimal global bandwidth. (c) Using the best bandwidths by ModelM. (d) Using the worst bandwidths by ModelM. (e) Using bandwidths obtained by DEALKR$_D$. (f) Using bandwidths obtained by DEALKR$_F$.

Table 6.1 Summary of results for artificial data set Banna3_100 with rankings according to result categories: LOO-CVE., MSE(Train) and MSE(Test).

| Method | Data Set | Banana3_100 | Ranking |
|---|---|---|---|
| **LOO-CVE Optimal** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0266 & 0 \\ 0 & 0.1630 \end{bmatrix}$ | |
| | *LOO-CVE:* | 19,182 | 3rd |
| | *MSE(Train):* | | |
| | *MSE(Test):* | 28,463 | 3rd |
| **Global** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0314 & 0 \\ 0 & 0.0314 \end{bmatrix}$ | |
| | *LOO-CVE:* | 32,372 | 4th |
| | *MSE(Train):* | 3,976 | 1st |
| | *MSE(Test):* | 49,446 | 4th |
| **CGLS *'best'*** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0254 & 0 \\ 0 & 0.1511 \end{bmatrix}$ | |
| | *LOO-CVE:* | 19,125 | 2nd |
| | *MSE(Train):* | 5,958 | 3rd |
| | *MSE(Test):* | 28,062 | 2nd |
| **CGLS *'worst'*** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 385{,}352 & 0 \\ 0 & 0.0218 \end{bmatrix}$ | |
| | *LOO-CVE:* | 342,088 | 5th |
| | *MSE(Train):* | 270,103 | 4th |
| | *MSE(Test):* | 375,461 | 5th |
| **DEALKR$_D$** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0254 & 0 \\ 0 & 0.1511 \end{bmatrix}$ | |
| | *LOO-CVE:* | 19,125 | 2nd |
| | *MSE(Train):* | 5,958 | 3rd |
| | *MSE(Test):* | 28,062 | 2nd |
| **DEALKR$_F$** | $\begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0250 & 0.0032 \\ 0.0002 & 0.1481 \end{bmatrix}$ | |
| | *LOO-CVE:* | 19,046 | 1st |
| | *MSE(Train):* | 5,894 | 2nd |
| | *MSE(Test):* | 27,978 | 1st |

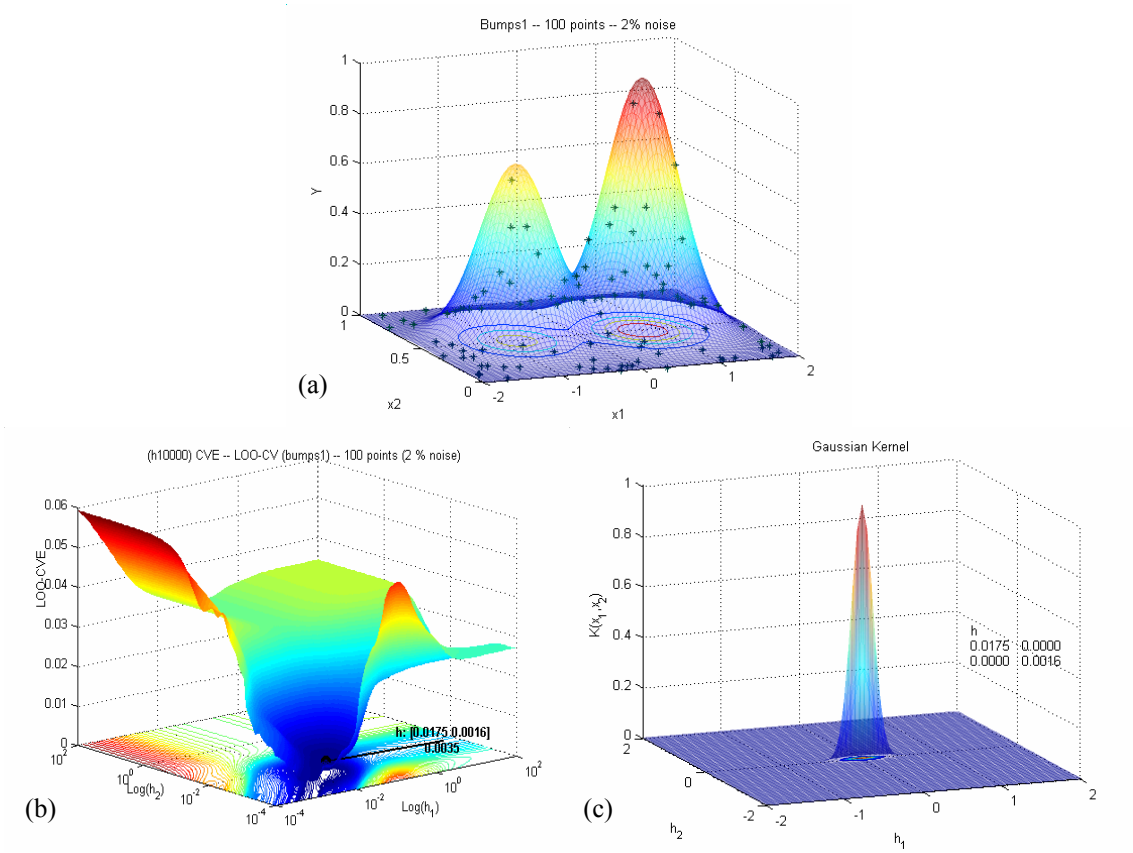



Figure 6.7 Bumps1_100 data set. (a) Asterisks are the training data (100 points) and the mesh plot is the test data (5,041 points). (b) LOO-CVE surface computed on a 100x100 grid. Global minimum at (0.0175, 0.0016 ) with LOO-CVE = 0.00352. (c) Gaussian kernel produced by LOO-CVE optimal bandwidths.

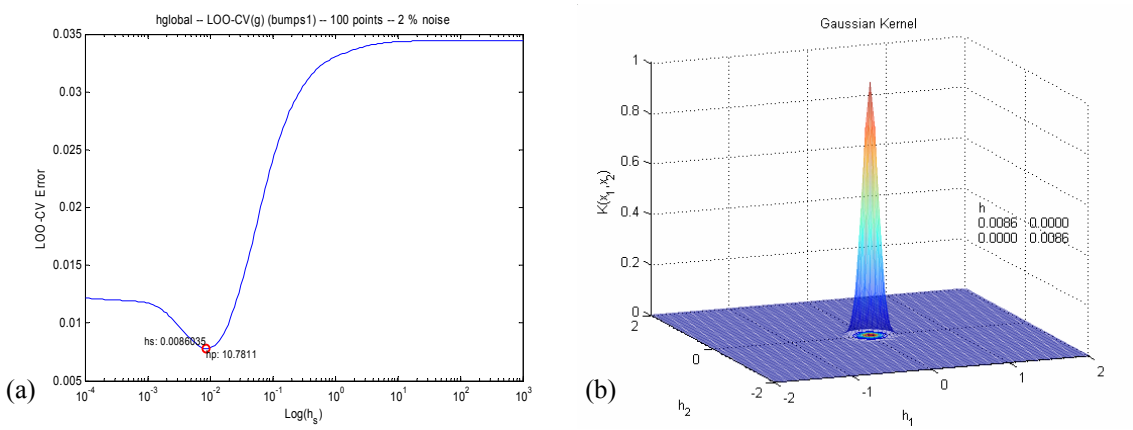



Figure 6.8 Global bandwidth solution for Bumps1_100: (a) LOO-CVE plot minimum at $h$ = 0.0086, with a LOO-CVE = 0.0078. (b) Gaussian kernel for optimal global bandwidth.

is the "informed" initial guess based on the underlying data. And the green asterisk is the convergence point of the informed initial guess. The presence of local minima are designated by scattered red x's.

The smoothing parameter histogram of the ModelM results for Bumps1 is in Figure 6.9 (b). The large standard deviations are an indication that multiple minima are present. The informed initial guess converged to the global minimum and resulted in a LOO-CVE = 0.0035 for bandwidth parameters $h_1,h_2$ = (0.0171, 0.0017), a training MSE = 0.0002 and a test MSE = 0.0038. The Gaussian kernel for these bandwidths is shown in Figure 6.9 (c). The function approximation produced from this set of bandwidth parameters is also the best set and is shown as a mesh plot in Figure 6.12 (c), with a surface plot of the test data and the training data as asterisks.

The Gaussian kernel for worst bandwidths obtained by ModelM is shown in Figure 6.9 (c). Figure 6.12 (d) shows the function approximation for the worst bandwidth parameters $h_1,h_2$ = (0.1109, 6.7464) obtained using ModelM, with a corresponding LOO-CVE = 0.03, a training MSE = 0.0272 and test MSE = 0.0381.

### 6.3.2.4 *DEALKR$_D$ Results*

Figure 6.10 (a) presents the evolutionary history of 100 generations shown superimposed on top of the LOO-CVE surface for Bumps1_100. The black +'s indicate the best members at differing times during the evolution of the population. The red +'s indicate the best member ever or the point of final convergence. Notice the global minimum was found every time. The Gaussian kernel for the DEALKR$_D$ obtained bandwidths is shown in Figure 6.10 (a) The function approximation resulting from the bandwidth parameters $h_1,h_2$ = (0.0172, 0.0017) is shown in Figure 6.12 (e). The LOO-CVE = 0.0035, the training MSE = 0.0002 and the test MSE = 0.0038.

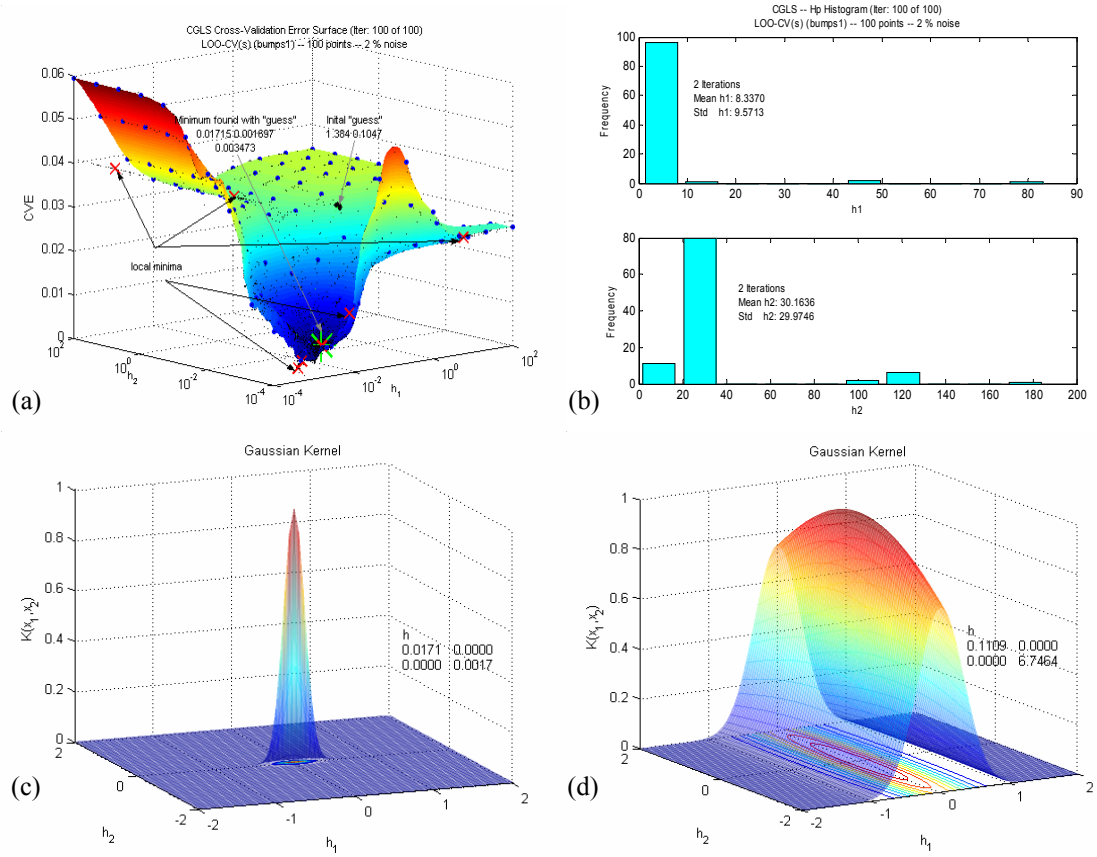



Figure 6.9 ModelM results for the Bumps1_100 data set: : (a) ModelM results superimposed over the optimal LOO-CVE surface. The blue dots indicate a starting point for each run. The red X's indicate a point of convergence for each run. The green dot indicates the "initial guess" based on std(x), and the green asterisk indicates the final minimum obtained using the initial guess. Note the initial guess did converge to the global minimum. (b) Smoothing parameter histograms from the 100 CGLS runs, with mean and standard deviations for each smoothing parameter. (c) Gaussian kernel for best bandwidths obtained by ModelM. (d) Gaussian kernel for worst bandwidths obtained by ModelM.

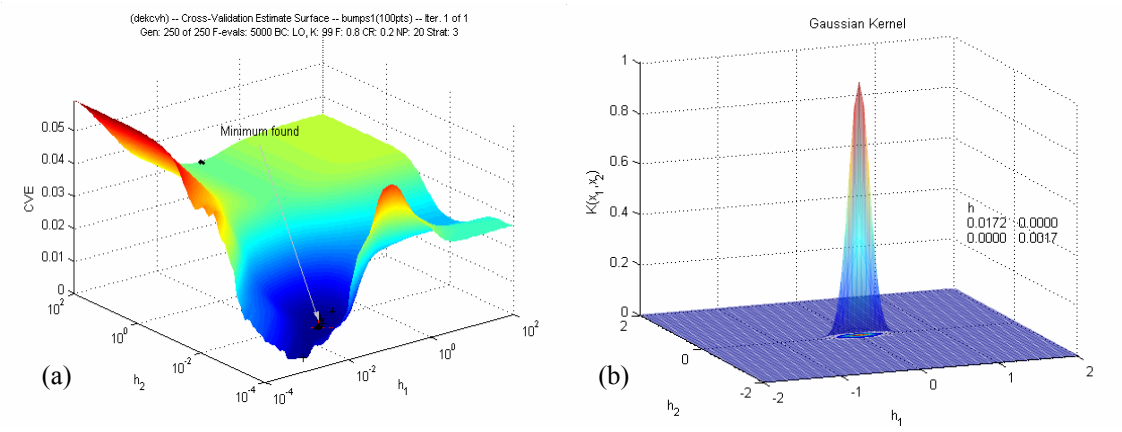



Figure 6.10 $DEALKR_D$ results for Bumps1_100: (a) $DEALKR_D$ 250 generation evolutionary history superimposed on the optimal LOO-CVE surface for Banana3_100. The black +'s indicate various "best members" throughout the 250 generation history. The red "+" indicates the final "best member" at the end of 250 generations. Note it is the global minimum. (b) Gaussian kernel for bandwidths obtained by $DEALKR_D$.

#### 6.3.2.5 *$DEALKR_F$ Results*

The full bandwidth matrix was optimized by minimizing the LOO-CVE. The smoothing parameter spectrum of the $DEALKR_F$ results is shown in The function approximation of Bumps1Figure 6.11_100 for the full bandwidth matrix is presented in Figure 6.11 (a) and associated Gaussian kernel is in Figure 6.11 (b) . The function approximation based on the full bandwidth matrix yielded a LOO-CVE = 0.0032, a training MSE = 0.0003, and test MSE = 0.0043 and is presented in Figure 6.12 (f).

Table 6.2 provides a summary of the results for all Bumps1_100.

### 6.3.3 Peaks1_100 Data Set

Data set Peaks1_100 was generated using the peaks.m function that ships with MATLAB. The output of peaks is generated by translating and scaling several Gaussian distributions. Figure 6.13 (a) shows Peaks1_100 test data as a surface plot and the training data as asterisks.

#### 6.3.3.1 *Optimal LOO-CVE Solution*

The optimal solution for Peaks1_100 was selected by minimizing the LOO-CVE. The LOO-CVE surface for Peaks1_100 is shown in Figure 6.13 (b). The smallest LOO-CVE value occurs at 0.3514 for the bandwidth parameters $h_1,h_2$ = (0.0705, 0.0404). The Gaussian kernel produced by optimal LOO-CVE bandwidths is shown in Figure 6.13 (c). The resulting function approximation had a test MSE = 0.4047 and is shown n Figure 6.18 (a).

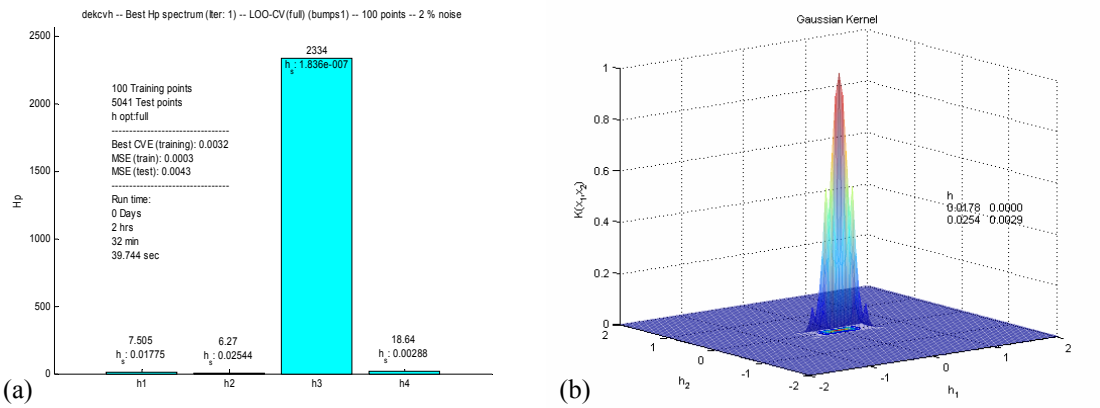



Figure 6.11 DEALKR$_F$ results for Bumps1_100: (a) Smoothing parameter spectrum. (b) Gaussian kernel for bandwidths obtained by DEALKR$_F$.

### 6.3.3.2 *Global Bandwidth Results*

The optimal global bandwidth parameter for Peaks1_100 was determined by computing the LOO-CVE for 200 different values of $h$ from $10^{-4}$ to $10^{3}$ and selecting the value of $h$ the produced the minimum LOO-CVE = 0.4087. Figure 6.14 (a) shows the LOO-CVE plot for Peaks1_100 with a minimum at $h = 0.0554$. The associated Gaussian kernel is plotted it Figure 6.14 (a). The resulting function approximation shown in Figure 6.18 (b).had a training MSE = 0.0317 and a test MSE = 0.3709.

### 6.3.3.3 *ModelM and CGLS LKR Results*

The ModelM survey technique was used to look for local minima in the LOO-CVE surface. Figure 6.15 (a) shows the results superimposed on top of the LOO-CVE surface. The blue dots designate the 100 staring points. The smaller black dots are LOO-CVE values at steps taken along convergence paths. Each red x is a final convergence point. The large black dot is the "informed" initial guess based on the underlying data. And the black asterisk is the convergence point of the informed initial guess. The presence of local minima are represented by scattered red x's.

The smoothing parameter histogram of the ModelM results is shown Figure 6.15 (b). Again notice the large standard deviations, which are an indicator of multiple minima. The initial guess converged to global minimum resulting in a LOO-CVE = 0.3506 for bandwidth parameters $h_1,h_2$ = (0.0728, 0.0430), a training MSE = 0.0375, a test MSE = 0.3945. The worst bandwidths were $h_1,h_2$ = (0.0000, 0.3459) with a LOO-CVE – 4.8873, a training MSE = 0.0070 and a test MSE = 6.2248.

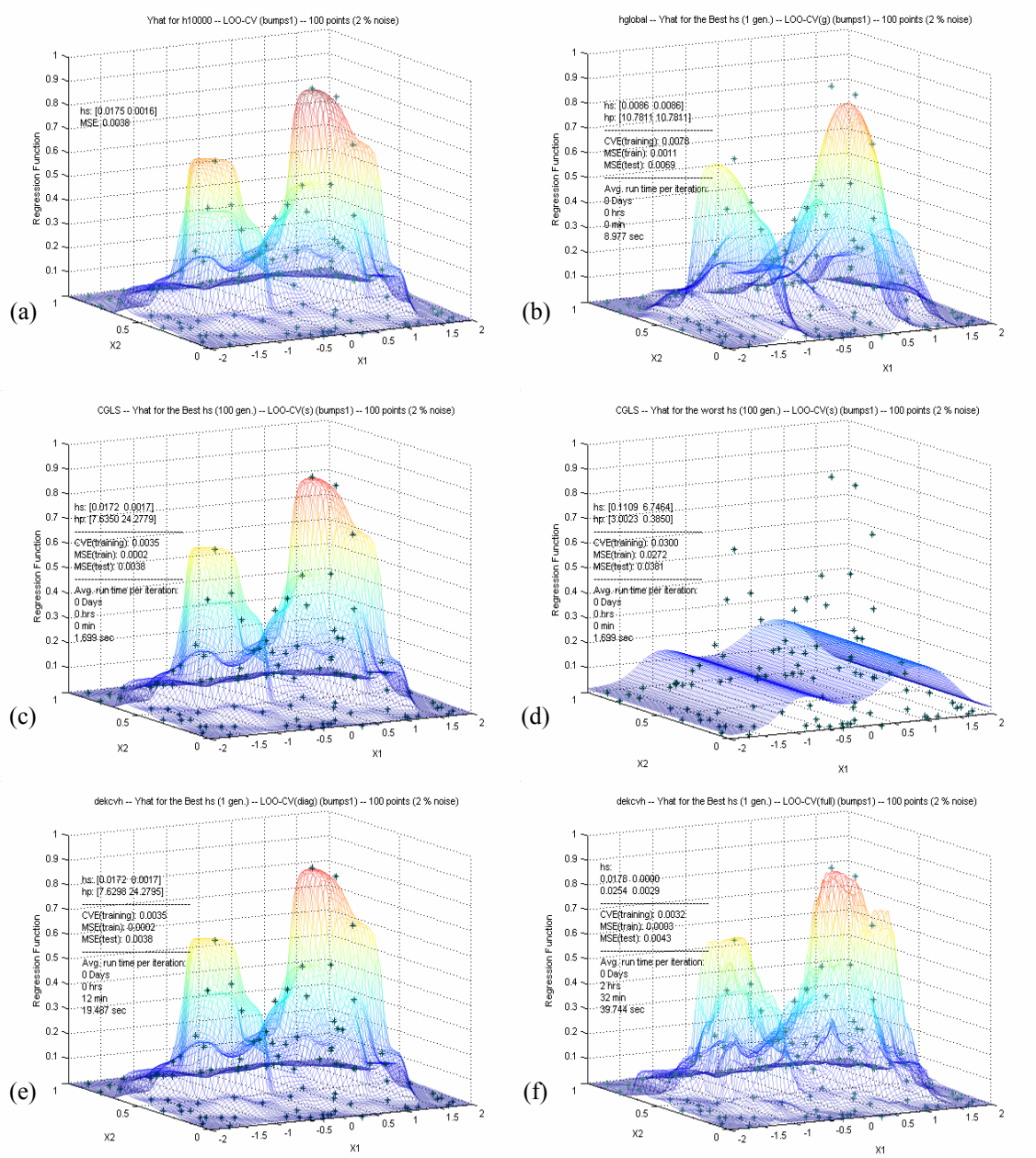



Figure 6.12 Function approximations of Bumps1_100: (a) Using optimal LOO-CVE bandwidths. (b) Using the optimal global bandwidth. (c) Using the best bandwidths by ModelM. (d) Using the worst bandwidths by ModelM. (e) Using bandwidths obtained by DEALKR$_D$. (f) Using bandwidths obtained by DEALKR$_F$.

Table 6.2 Summary of results for artificial data set Bumps1_100 with rankings according to result categories: LOO-CVE., MSE(Train) and MSE(Test).

| Method | Data Set | Bumps1_100 | Ranking |
|---|---|---|---|
| **LOO-CVE Optimal** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0175 & 0 \\ 0 & 0.0016 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0035 | 2nd |
| | *MSE(Train):* | | |
| | *MSE(Test):* | 0.0038 | 1st |
| **Global** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0086 & 0 \\ 0 & 0.0086 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0078 | 4th |
| | *MSE(Train):* | 0.0011 | 3rd |
| | *MSE(Test):* | 0.0069 | 3rd |
| **CGLS *'best'*** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0172 & 0 \\ 0 & 0.0017 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0035 | 2nd |
| | *MSE(Train):* | 0.0002 | 1st |
| | *MSE(Test):* | 0.0038 | 1st |
| **CGLS *'worst'*** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.1109 & 0 \\ 0 & 6.7464 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0300 | 3rd |
| | *MSE(Train):* | 0.0272 | 4th |
| | *MSE(Test):* | 0.0381 | 4th |
| **DEALKR$_D$** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0172 & 0 \\ 0 & 0.0017 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0035 | 2nd |
| | *MSE(Train):* | 0.0002 | 1st |
| | *MSE(Test):* | 0.0038 | 1st |
| **DEALKR$_F$** | $\begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0178 & 0.0000 \\ 0.0254 & 0.0029 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.0032 | 1st |
| | *MSE(Train):* | 0.0003 | 2nd |
| | *MSE(Test):* | 0.0043 | 2nd |

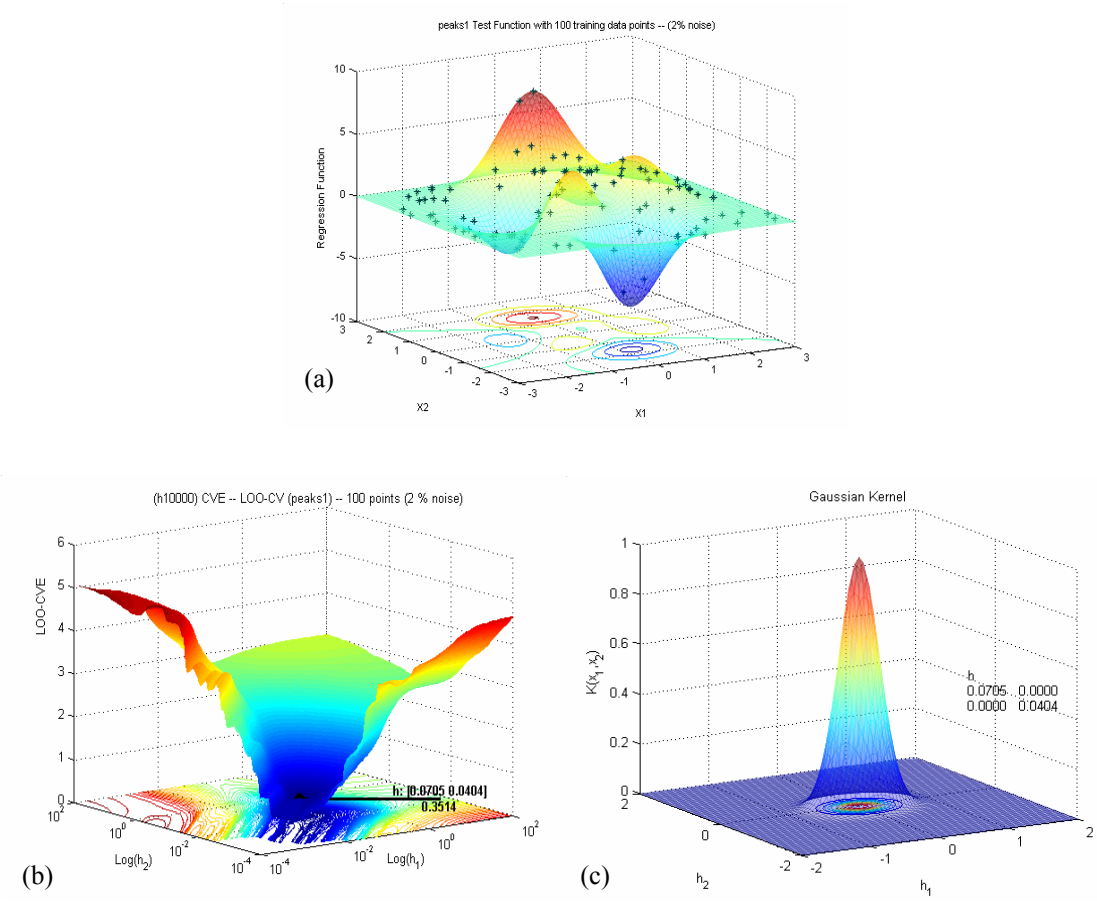



Figure 6.13 Peaks1_100 data set: (a) Asterisks are the training data (100 points) and the mesh plot is the test data (5,041 points). (b) LOO-CVE surface computed on a 100x100 grid. Global minimum at $h_1,h_2$ = (0.0705, 0.0404) with LOO-CVE = 0.3514. (c) Gaussian kernel produced by LOO-CVE optimal bandwidths.

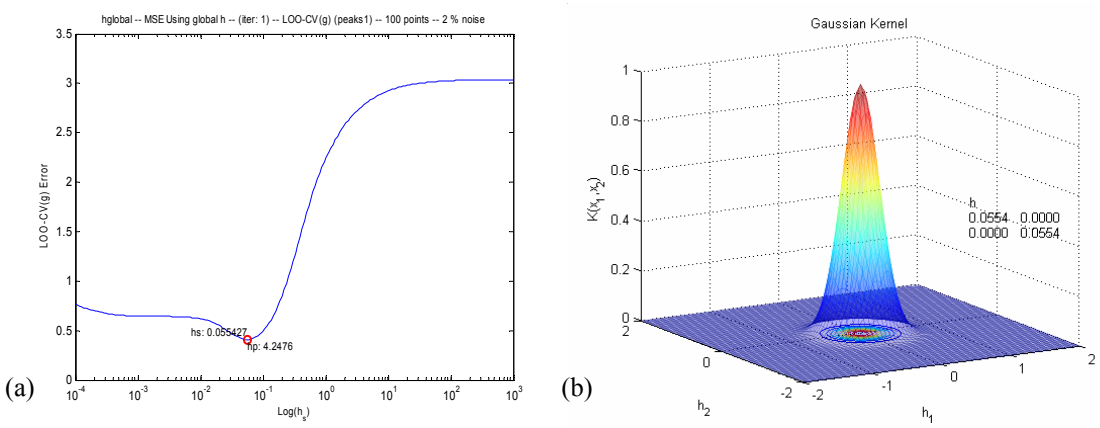



Figure 6.14 Global bandwidth solution for Peaks1_100: (a) LOO-CVE plot minimum at $h$ = 0.0554 with a LOO-CVE = 0.4087. (b) Gaussian kernel for optimal global bandwidth.

The resulting Gaussian kernels for the best and worst bandwidths obtained by ModelM are presented in Figure 6.15 (c) and Figure 6.15 (d). The function approximations produced by the best and worst bandwidths are presented in Figure 6.18 (c) and Figure 6.18 (d) respectively.

### 6.3.3.4 *DEALKR$_D$ Results*

Figure 6.16 (a) presents the evolutionary history of 250 generations superimposed on the LOO-CVE surface for peaks1_100. The black +'s indicate the best members at differing times during the evolution of the population. The red +'s indicate the best member ever or the point of final convergence. The Gaussian kernel for bandwidths $h_1,h_2$ = (0.0728, 0.0430) obtained by DEALKR$_D$ is shown in Figure 6.16 (b). The resulting function approximation with a LOO-CVE = 0.3506, a training MSE = 0.0375 and test MSE = 0.3945 is shown in Figure 6.18 (e).

### 6.3.3.5 *DEALKR$_F$ Results*

The full bandwidth matrix was optimized by minimizing the LOO-CVE. The smoothing parameter spectrum is presented in Figure 6.17 (a). The Gaussian kernel produced by the DEALKR$_F$ full bandwidth matrix I shown in Figure 6.17 (b). Notice the squared off shape indicating the influence from off-diagonal parameters. The resulting function approximation for the full matrix of bandwidth parameters is presented in Figure 6.18 (f). It yielded a LOO-CVE of 0.3193, a training MSE of 0.0247, and test MSE of 0.4426.

Table 6.3 provides a summary of the results for Peaks1_100.

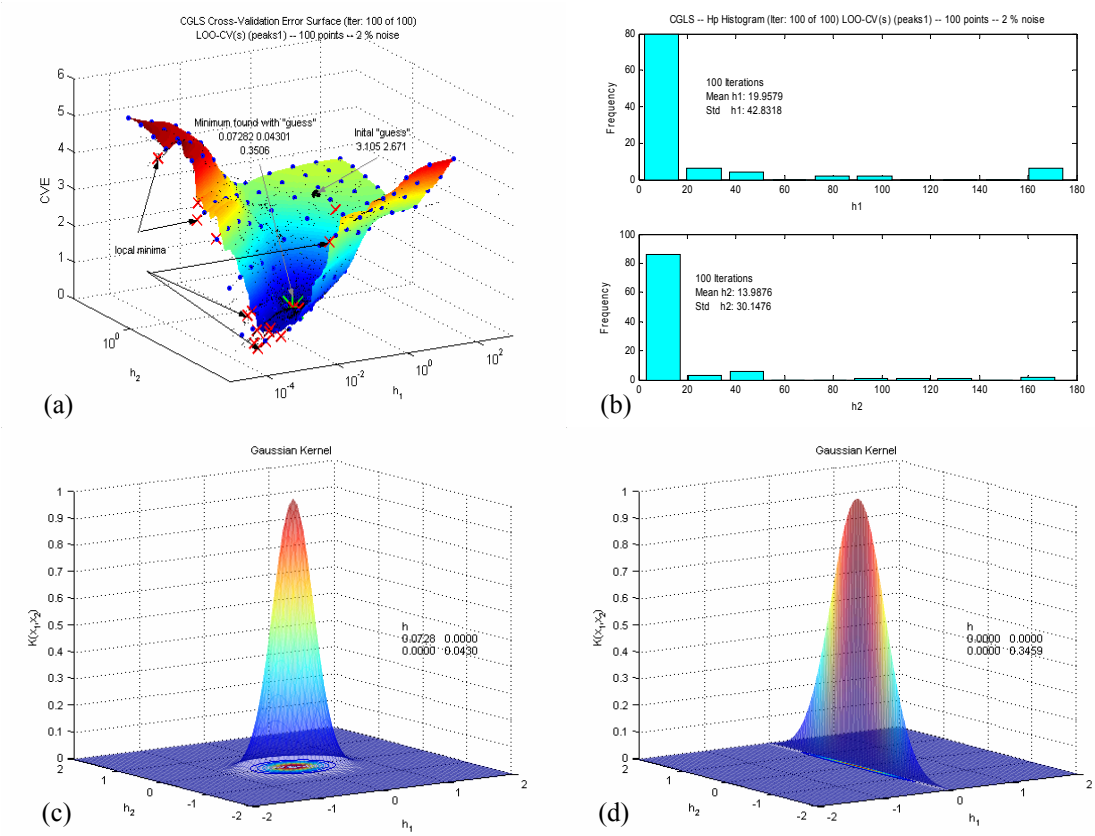



Figure 6.15 ModelM results for the Peaks1_100 data set: : (a) ModelM results superimposed over the optimal LOO-CVE surface. The blue dots indicate a starting point for each run. The red X's indicate a point of convergence for each run. The green dot indicates the "initial guess" based on std(x), and the green asterisk indicates the final minimum obtained using the initial guess. Note the initial guess did converge to the global minimum. (b) Smoothing parameter histograms from the 100 CGLS runs, with mean and standard deviations for each smoothing parameter. (c) Gaussian kernel for best bandwidths obtained by ModelM. (d) Gaussian kernel for worst bandwidths obtained by ModelM.

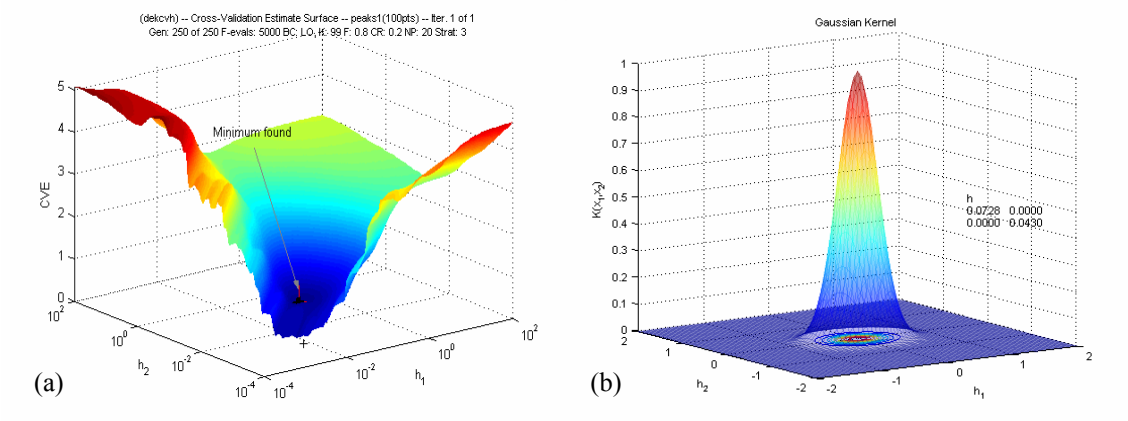



Figure 6.16 DEALKR$_D$ results for Peaks1_100: (a) DEALKR$_D$ 250 generation evolutionary history superimposed on the optimal LOO-CVE surface for Peaks1_100. The black +'s indicate various "best members" throughout the 250 generation history. The red "+" indicates the final "best member" at the end of 250 generations. Note it is the global minimum. (b) Gaussian kernel for bandwidths obtained by DEALKR$_D$.

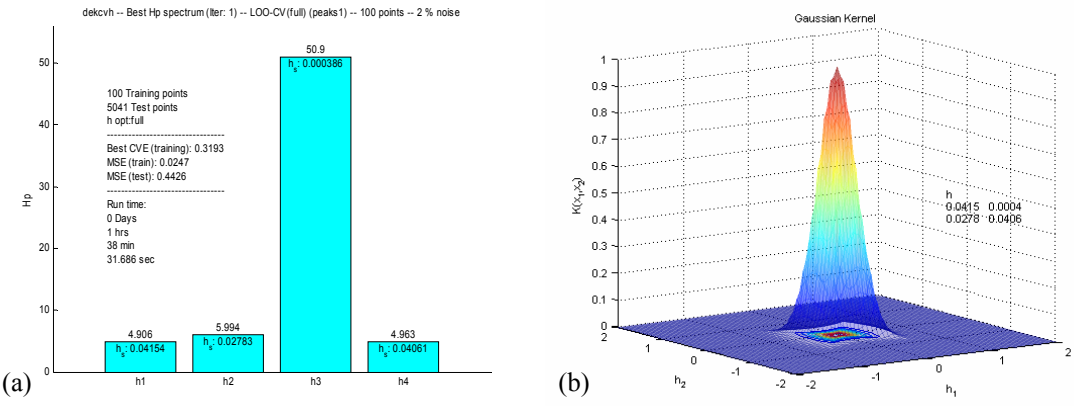



Figure 6.17 DEALKR$_F$ results for Peaks1_100: (a) Smoothing parameter spectrum. (b) Gaussian kernel for bandwidths obtained by DEALKR$_F$.

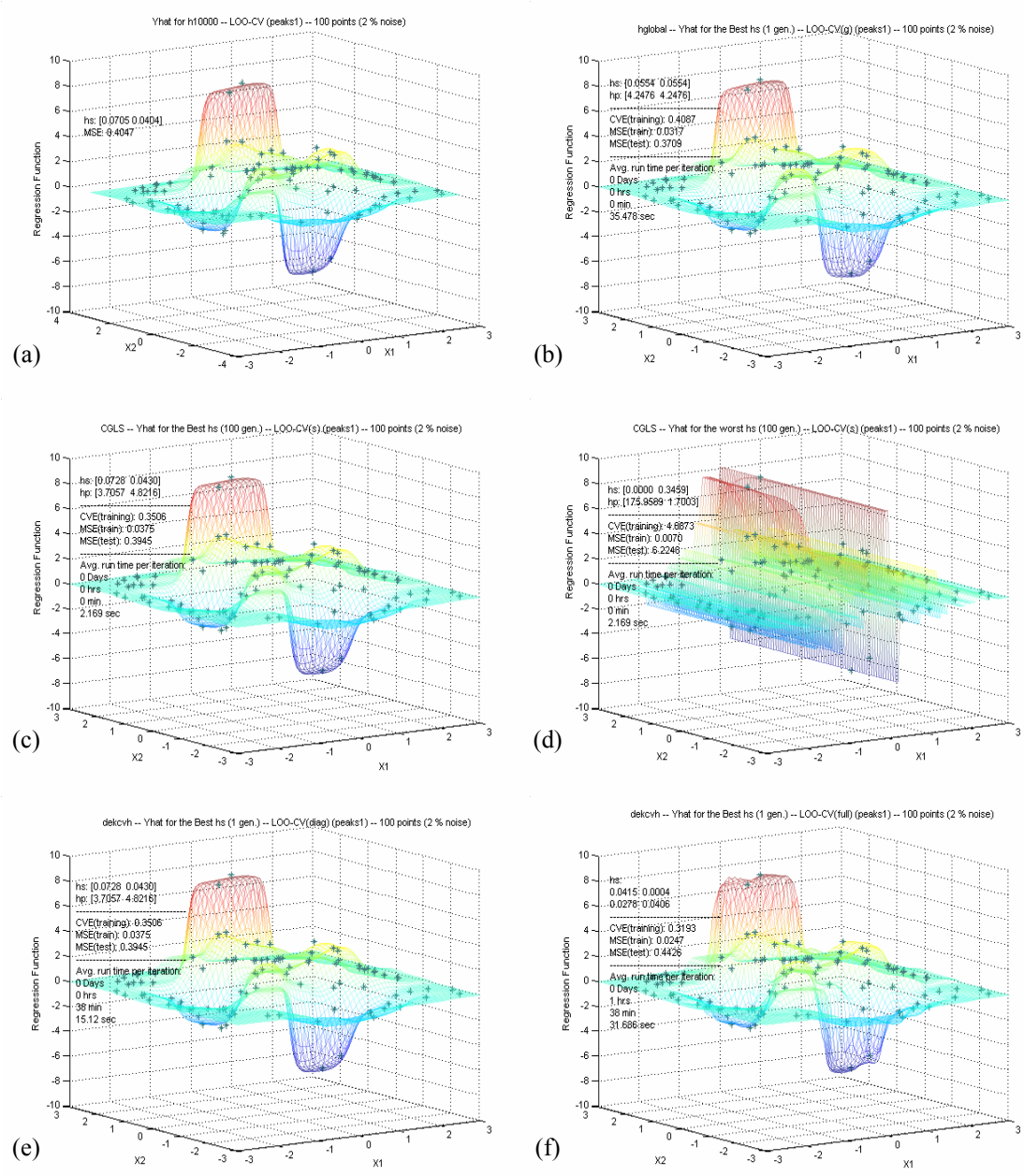



Figure 6.18 Function approximations of Peaks1_100: (a) Using optimal LOO-CVE bandwidths. (b) Using the optimal global bandwidth. (c) Using the best bandwidths by ModelM. (d) Using the worst bandwidths by ModelM. (e) Using bandwidths obtained by DEALKR$_D$. (f) Using bandwidths obtained by DEALKR$_F$.

Table 6.3 Summary of results for artificial data set Peaks1_100 with rankings according to result categories: LOO-CVE., MSE(Train) and MSE(Test).

| Method | Data Set | Peaks1_100 | Ranking |
|---|---|---|---|
| **LOO-CVE Optimal** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0705 & 0 \\ 0 & 0.0404 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.3514 | 3rd |
| | *MSE(Train):* | | |
| | *MSE(Test):* | 0.4047 | 3rd |
| **Global** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0554 & 0 \\ 0 & 0.0554 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.4087 | 4th |
| | *MSE(Train):* | 0.0317 | 3rd |
| | *MSE(Test):* | 0.3709 | 1st |
| **CGLS *'best'*** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0728 & 0 \\ 0 & 0.0430 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.3506 | 2nd |
| | *MSE(Train):* | 0.0375 | 4th |
| | *MSE(Test):* | 0.3945 | 2nd |
| **CGLS *'worst'*** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0000 & 0 \\ 0 & 0.3459 \end{bmatrix}$ | |
| | *LOO-CVE:* | 4.8873 | 5th |
| | *MSE(Train):* | 0.0070 | 1st |
| | *MSE(Test):* | 6.2248 | 5th |
| **DEALKR$_D$** | $\begin{bmatrix} h_{1,1} & 0 \\ 0 & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0728 & 0 \\ 0 & 0.0430 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.3506 | 2nd |
| | *MSE(Train):* | 0.0375 | 4th |
| | *MSE(Test):* | 0.3945 | 2nd |
| **DEALKR$_F$** | $\begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{1,2} \end{bmatrix}$ | $\begin{bmatrix} 0.0415 & 0.0004 \\ 0.0278 & 0.0406 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.3193 | 1st |
| | *MSE(Train):* | 0.0247 | 2nd |
| | *MSE(Test):* | 0.4426 | 4th |

## 6.4 Artificial Data Set with Four Input Variables

### 6.4.1 Cascade1_100 Data Set

The Cascade1_100 data set is a cascaded function that was generated as follows. First, four x variables were created with 100 uniformly spaced values from –0.5 to 0.5. Next, the four variables A, B, C, D which are functions of $x_1$, $x_2$, $x_3$ and $x_4$, were computed:

$$\mathrm{A} = \sin(10\mathrm{x}_1);\ B = x_2;\ C = \sin(5\mathrm{x}_3);\ D = e^{\sin(15\mathrm{x}_4)}. \tag{6.4}$$

And finally the output $y$ is created:

$$y = e^{(2\mathrm{A}\cdot\sin(\pi\,\mathrm{D}))} + \sin(50\mathrm{BC}). \tag{6.5}$$

Figure 6.19 (a) shows the inputs A, B, C and D of the training data as diamonds and the test set as solid lines. Figure 6.19 (b) is a plot of test output $y$ as a solid line and the training data as asterisks.

#### 6.4.1.1 *Global Bandwidth Parameters*

The optimal global bandwidth parameter for Cascade1_100 was determined by computing the LOO-CVE for 200 different values of $h$ from $10^{-4}$ to $10^{3}$ and selecting the value of $h$ the produced the minimum LOO-CVE = 0.1685. Figure 6.20 (a) shows the LOO-CVE plot with a minimum at $h$ = 0.0073. The function approximation based on the global bandwidth parameter resulted in a LOO-CVE = 0.1685, a training MSE = 0.0106 and a test MSE = 0.0164. Figure 6.20 (b) shows the resulting function approximation.

#### 6.4.1.2 *ModelM and CGLS LKR Results*

ModelM was used to "map" the LOO-CVE hyper-surface and look for local minima. But a new way of generating the initial guesses was required. Instead of a single grid of two variables each input dimension was sampled using a logarithmic permutation described by this snippet of MATLAB code:

```
nGrid=floor(numIter/numH)
hs0=zeros(numH,numIter)
for i = 1:numH
     ind=randperm(numIter);
     seedHs  = logspace(-4,2,numIter);
     hs0(i,:) = seedHs(:,ind);

end
```

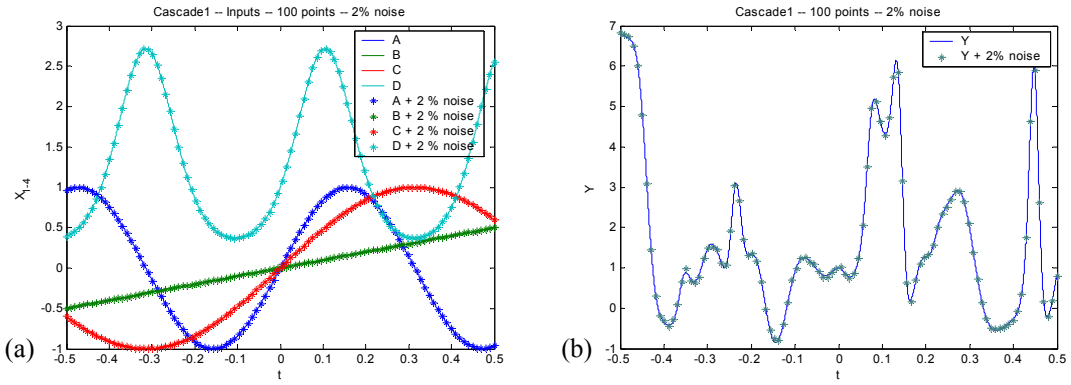



Figure 6.19 Cascade1_100 data set: (a) Four input variables (b) Target output variable.

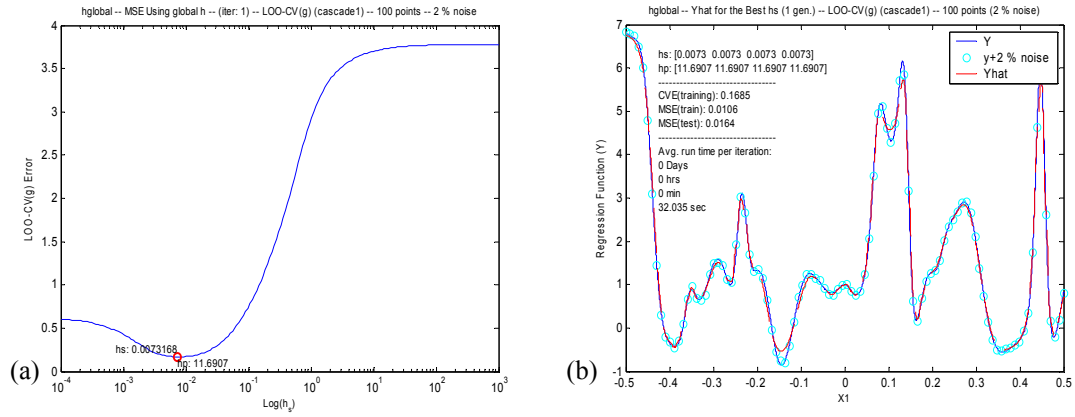



Figure 6.20 Global bandwidth solution for Cascade1_100: (a) LOO-CVE plot. (b) Function approximation using optimal global bandwidth parameter.

where, **`numIter`** is the number of points to be generated along each input dimension; and **`numH`** is the number of bandwidth parameters to be optimized, which is four for Cascade1. Figure 6.21 is a scatter plot showing starting point exemplars for $h_1$ vs. $h_2$, $h_1$ vs. $h_3$ and $h_1$ vs. $h_4$. Similar scatter plots could be generated for all parameter combinations. Since the LOO-CVE surface is a hyper-surface it cannot be visualized but we can look at the smoothing parameter histogram to try and determine whether local minima exist.

The smoothing parameter histogram results for Cascade1_100 are in plotted in Figure 6.22. Included on histogram of each parameter are the value of the informed initial guess and the value it converged to. Notice the large spread of values in the histograms and the large standard deviations, both are indicators that multiple minima are present.

The smallest LOO-CVE obtained by ModelM was 0.1438 for bandwidth parameters diag(90,590, 0.00001, 0.0028, 0.0171), with a training MSE = 0. 0.0014 and a test MSE = 0.0339. The smoothing parameter plot for the best set of bandwidth parameters is presented in Figure 6.23 (a). The resulting function approximation produced by the best bandwidth parameters is shown in Figure 6.23 (b).

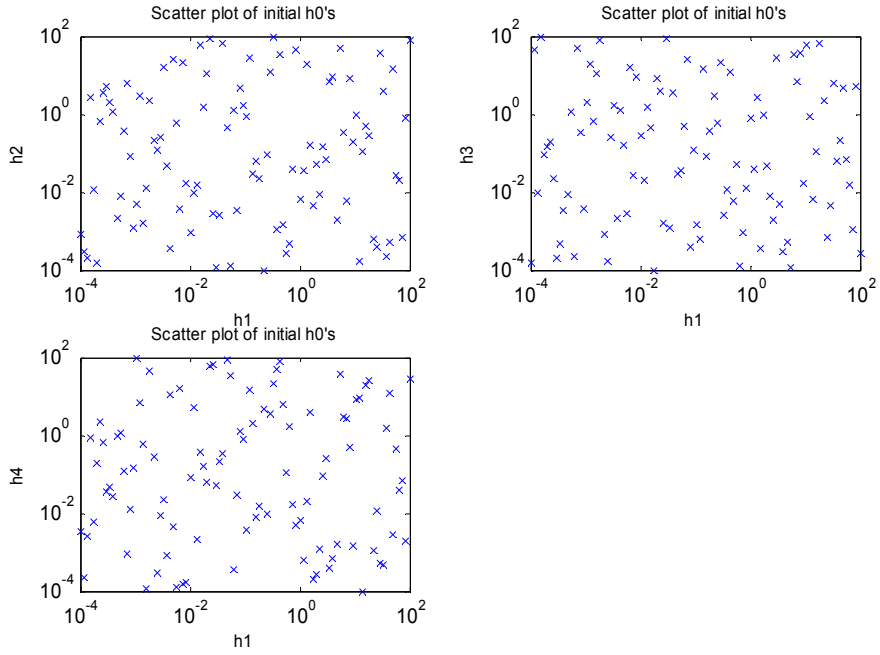



Figure 6.21 Scatter plot example of initial values used by ModelM.

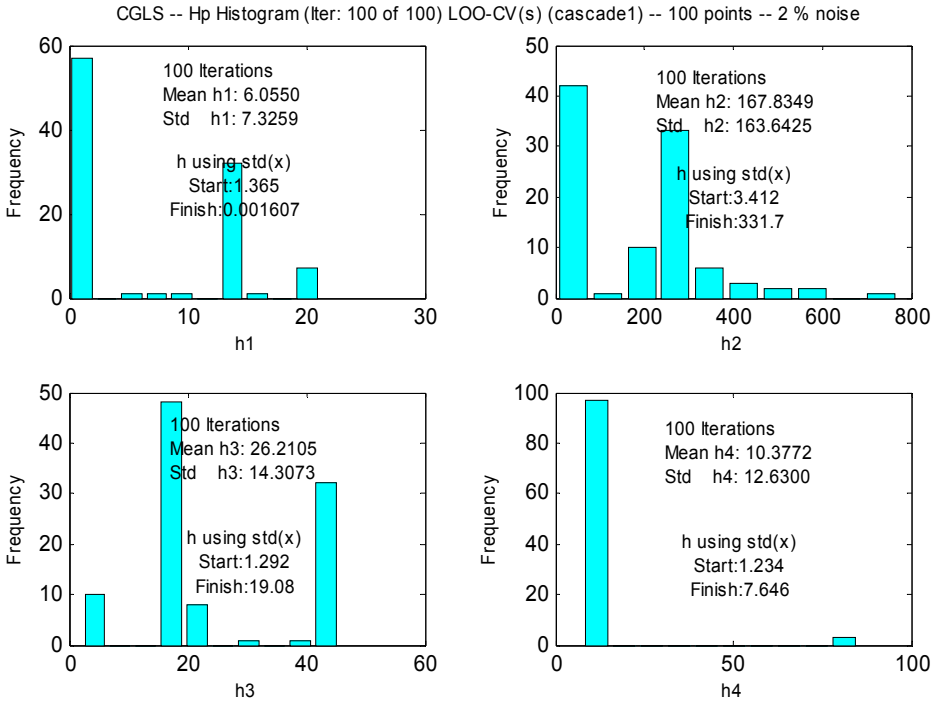
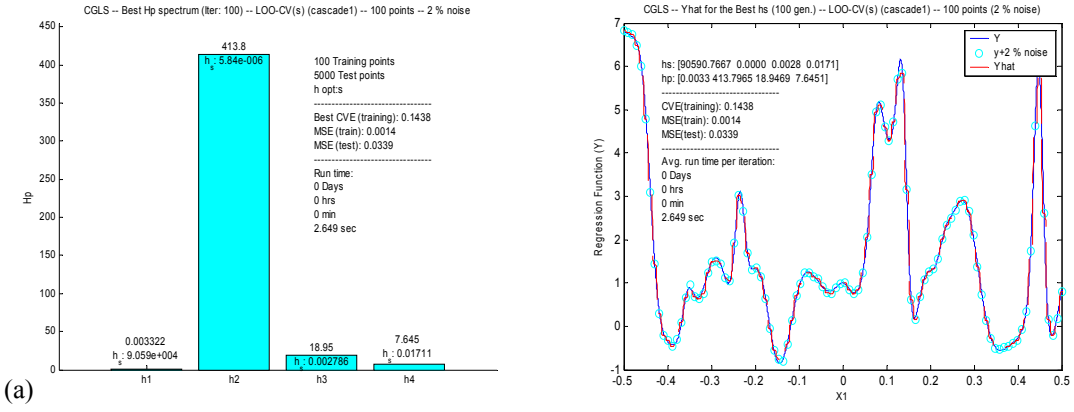



Figure 6.22 Smoothing parameter histograms for Cascade1_100 obtained by ModelM for 100 CGLS runs.




(a)

Figure 6.23 ModelM results for Cascade1_100: (a) Smoothing parameter spectrum for the best bandwidths obtained by ModelM. (b) The function approximation using the best bandwidth parameters.

6.4.1.3 *$DEALKR_D$ Results*

The smoothing parameter spectrum of the results obtained by $DEALKR_D$ is shown in Figure 6.24 (a). The resulting function approximation shown in Figure 6.24 (b) produce a LOO-CVE = 0.1438, a training MSE = 0.0014 and a test MSE = 0.0626.

6.4.1.4 *$DEALKR_F$ Results*

The smoothing parameter spectrum for the full bandwidth matrix obtained by $DEALKR_F$ is shown in Figure 6.25 (a). The resulting function approximation which is shown in Figure 6.25 (b) had a LOO-CVE = 0.1291, a training MSE = 0.0082, and a test MSE = 0.0403.

A summary of the results for Cascade1_100 is presented in Table 6.4.

## 6.5 Selection of Relevant Input Variables

This section demonstrates the ability of using $DEALKR_D$ to select relevant input variables while damping or minimizing the influence of irrelevant or noisy input variables.

### 6.5.1 G10D1_100 Data Set: Five Relevant and Five Irrelevant Variables

The g10d data set was generated by a system described in [63]. The target vector is defined by:

$$y = 10\sin(\pi x_1 x_2) + 20\left(x_3 - \frac{1}{2}\right)^2 + 10x_4 + 5x_5 + \varepsilon \tag{6.6}$$

with the Gaussian noise added $\varepsilon \sim N(0,1)$. The input vector contained 10 sets of random values selected from the uniform distribution $x_1 \ldots x_{10},\ x_i \sim U([0,1])$.

Form equation 5.7 it is apparent that only the first five inputs contribute to the output variable in any way and therefore relevant, while the last five are clearly irrelevant. A training data set of 100 samples and a test data set of 5,000 samples were generated. For kernel regression the 100 data samples is sparse for 10 dimensions. Figure 6.26 (a) shows a scatter plot of the training data for the 10 input variables and Figure 6.26 (b) is the target output.

(b)



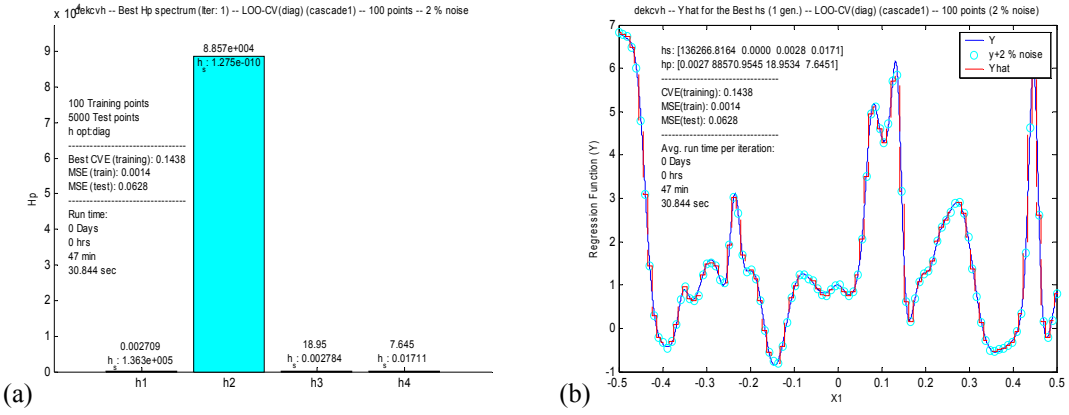



Figure 6.24 $DEALKR_D$ results for Cascade1_100: (a) Smoothing parameter spectrum for bandwidths obtained by $DEALKR_D$ (b) Function approximation using the best bandwidth parameters.

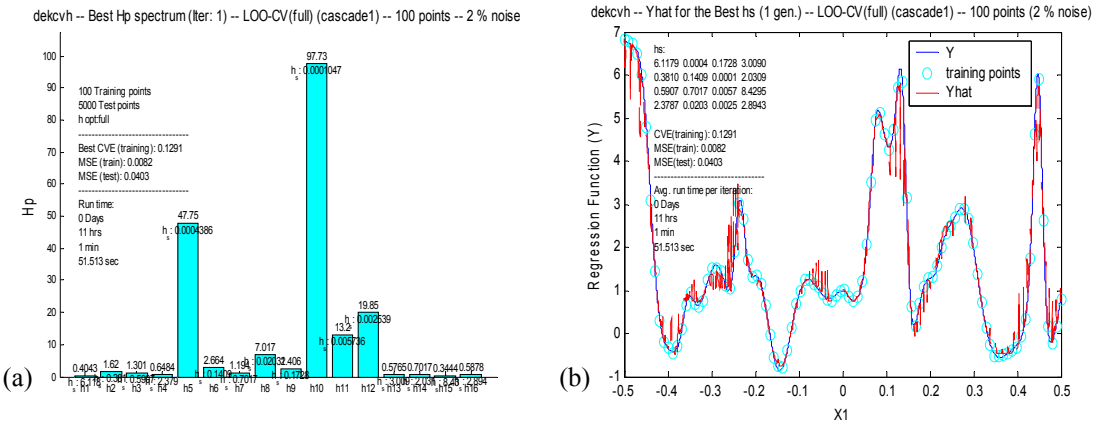
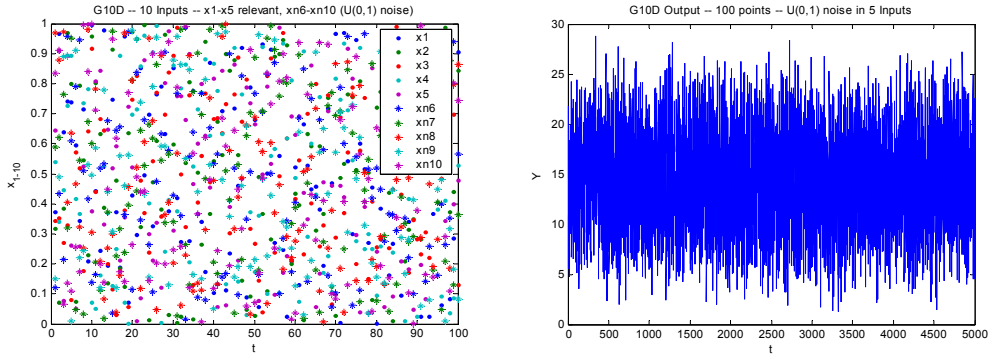



Figure 6.25 $DEALKR_F$ results for Cascade1_100: (a) Smoothing parameter spectrum for bandwidths obtained by $DEALKR_F$ (b) Function approximation using the best bandwidth parameters.

Table 6.4 Summary of results for Cascade1_100 with rankings according to result categories: LOO-CVE., MSE(Train) and MSE(Test).

| Method | Data Set | Cascade1_100 | Ranking |
|---|---|---|---|
| **Global** | $\begin{bmatrix} h_{1,1} & 0 & 0 & 0 \\ 0 & h_{2,2} & 0 & 0 \\ 0 & 0 & h_{3,3} & 0 \\ 0 & 0 & 0 & h_{4,4} \end{bmatrix}$ | $\begin{bmatrix} 0.0073 & 0 & 0 & 0 \\ 0 & 0.0073 & 0 & 0 \\ 0 & 0 & 0.0073 & 0 \\ 0 & 0 & 0 & 0.0073 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1685 | 3rd |
| | *MSE(Train):* | 0.0106 | 3rd |
| | *MSE(Test):* | 0.0164 | 1st |
| **CGLS 'best'** | $\begin{bmatrix} h_{1,1} & 0 & 0 & 0 \\ 0 & h_{2,2} & 0 & 0 \\ 0 & 0 & h_{3,3} & 0 \\ 0 & 0 & 0 & h_{4,4} \end{bmatrix}$ | $\begin{bmatrix} 90{,}591 & 0 & 0 & 0 \\ 0 & 0.0000 & 0 & 0 \\ 0 & 0 & 0.0028 & 0 \\ 0 & 0 & 0 & 0.0171 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1438 | 2nd |
| | *MSE(Train):* | 0.0014 | 1st |
| | *MSE(Test):* | 0.0339 | 2nd |
| **DEALKR$_D$** | $\begin{bmatrix} h_{1,1} & 0 & 0 & 0 \\ 0 & h_{2,2} & 0 & 0 \\ 0 & 0 & h_{3,3} & 0 \\ 0 & 0 & 0 & h_{4,4} \end{bmatrix}$ | $\begin{bmatrix} 136{,}267 & 0 & 0 & 0 \\ 0 & 0.0000 & 0 & 0 \\ 0 & 0 & 0.0028 & 0 \\ 0 & 0 & 0 & 0.0171 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1438 | 2nd |
| | *MSE(Train):* | 0.0014 | 1st |
| | *MSE(Test):* | 0.0626 | 4th |
| **DEALKR$_F$** | $\begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} \\ h_{3,1} & h_{3,2} & h_{3,3} & h_{3,4} \\ h_{4,1} & h_{4,2} & h_{4,3} & h_{4,4} \end{bmatrix}$ | $\begin{bmatrix} 6.1179 & 0.0004 & 0.1728 & 3.0090 \\ 0.3810 & 0.1409 & 0.0001 & 2.0309 \\ 0.5907 & 0.7017 & 0.0057 & 8.4295 \\ 2.3787 & 0.0203 & 0.0025 & 2.8943 \end{bmatrix}$ | |
| | *LOO-CVE:* | 0.1291 | 1st |
| | *MSE(Train):* | 0.0082 | 2nd |
| | *MSE(Test):* | 0.0403 | 3rd |




(a) (b)

Figure 6.26 G10D1_100 data set: (a) Scatter plot of the 10 input variables (training set). (b) Target output variable (test set).

#### 6.5.1.1 *Global Bandwidth Results*

The global bandwidth parameter was selected by minimizing the LOO-CVE. A plot of the LOO-CVE vs. bandwidth is presented in Figure 6.27. The minimum LOO-CVE was 13.2184 for $h = 0.0977$ and training MSE = 1.3879 and test MSE = 9.9720. Clearly use of a global bandwidth is useless for input selection.

#### 6.5.1.2 *ModelM/CGLS LKR Results*

ModelM was used to "map" the LOO-CVE hyper-surface for G10D1_100 and look for local minima. Figure 6.28 is a scatter plot of exemplar data points for $h_1$ vs. $h_2$-$h_{10}$. Similar scatter plots could be generated for all parameter combinations.

The smoothing parameter histogram is presented in Figure 6.29. Included on the histogram of each parameter are the value of the informed initial guess and the value it converged to. The spread in the smoothing parameter values and the large standard deviations suggest there are local minima in the error hyper-surface. The spectrum for the best set of smoothing parameters obtained by ModelM is presented in Figure 6.30. Observe that CGLS attenuates the smoothing parameters associated with only three of the five irrelevant input dimensions, 6, 7, and 10. A nearly equal smoothing parameter value was assigned to inputs 1, 2 and 4 which accurately reflect the balanced roll of x1 and x2 in equation (5.1). It did not however attenuate 8 and 9. The LOO-CVE was 4.6811, the training MSE 0.8611 and the test MSE 5.7377.

#### 6.5.1.3 *DEALKR$_D$ Results*

The DEALKR$_D$ smoothing parameter spectrum is presented in Figure 6.31. Notice that DEALKR$_D$ accurately identified dimensions 1-5.as the most significant, and dimensions 6, 7, 8 and 10 as insignificant.

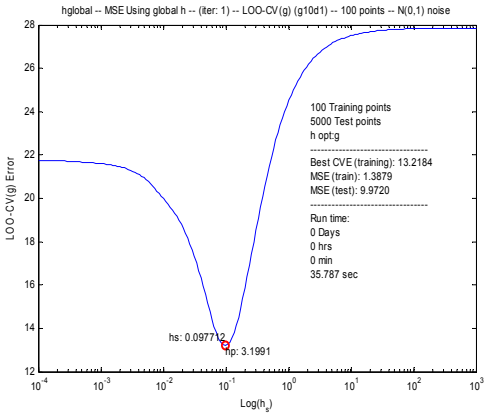



Figure 6.27 Global bandwidth LOO-CVE plot for G10D1_100 data set.

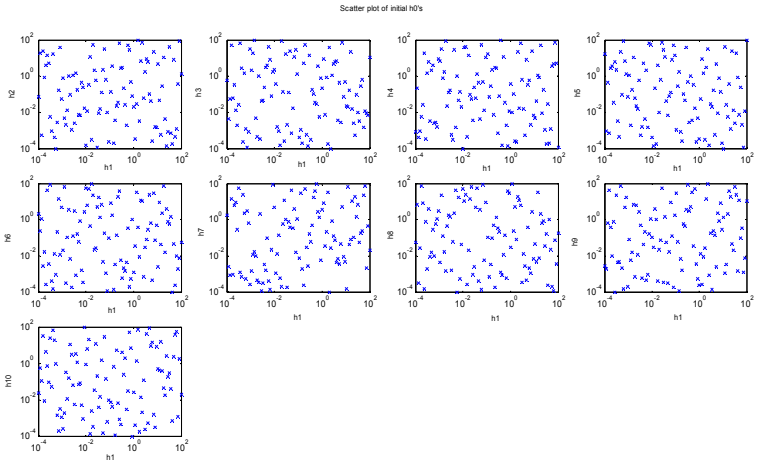



Figure 6.28 Scatter plot of exemplar initialization points used by ModelM .

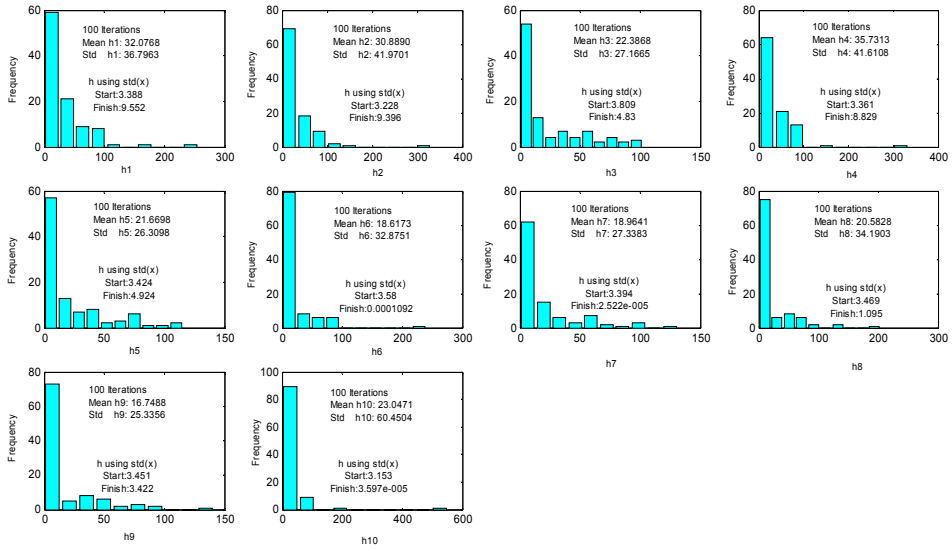



Figure 6.29 Smoothing parameter histogram for G10D1_100 obtained by ModelM for 100 CGLS runs.

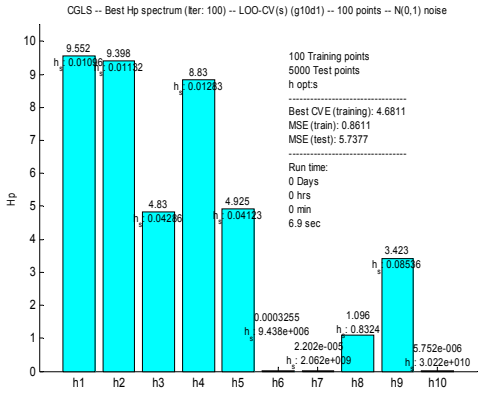



Figure 6.30 ModelM results for G10D1_100. Smoothing parameter spectrum for best bandwidths.

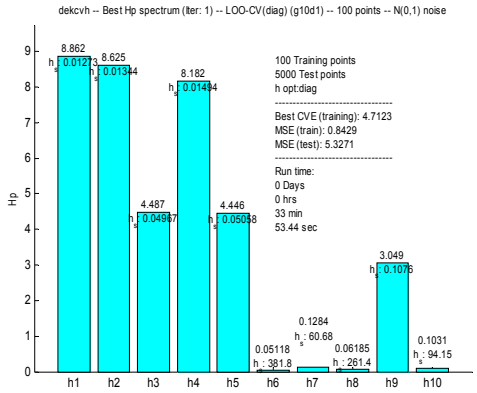



Figure 6.31 DEALKR$_D$ smoothing parameter spectrum of best bandwidths obtained for G10D1_100.

Although variable 9 was not accurately identified as insignificant it is worth noting that given the scarcity of the data, only 100 samples for 10 dimensions, the random nature of the values in dimension 9 are correlated with the output which is indicated by the correlation coefficients: (0.3918, 0.4241, 0.0936, 0.6579, 0.2063,-0.0399, 0.0391, -0.0967, 0.1387, -0.0192). Variable 9 (0.1387) is more highly correlated than variable 3 (0.0936). But DEALK$_D$ did assign a bandwidth to variable 9 (0.1076) nearly an order of magnitude larger than any of the bandwidths for 1-5. The LOO-CVE for DEALKR$_D$ was 4.1648, the training MSE was 0.8429, and the test MSE was 5.3271. The LOO-CVE was slightly higher than that for CGLS, while both the training and test MSE were slightly lower.

A summary of the results for G10D1 is provided in Table 6.5

## 6.5.2 Cascade81_100 Data Set: Four Relevant and Four Irrelevant Variables

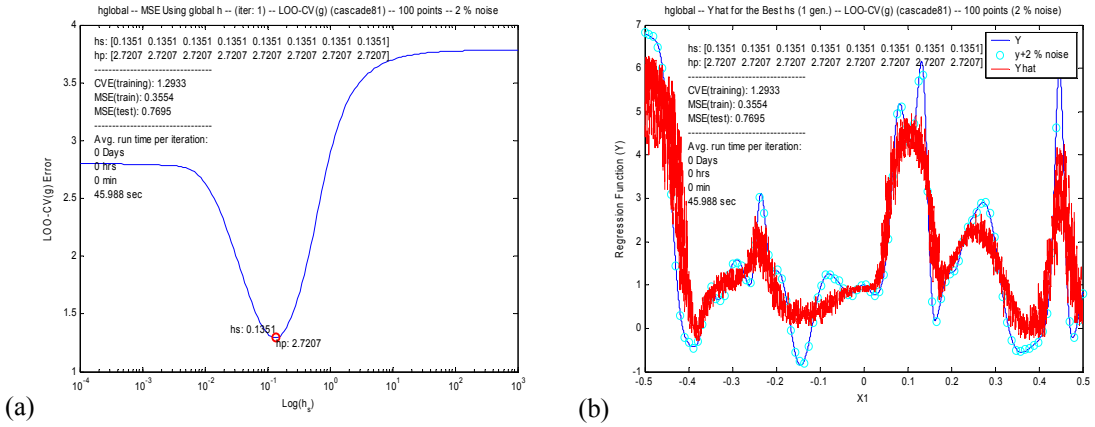
The Cascade81_100 data set has eight input dimension (4 relevant + 4 irrelevant). It takes the original Cascade1_100 and adds four noisy input dimensions which are randomly sampled form a uniform random distribution, $x_5 \cdot x_8 \sim U([0,1])$.

### 6.5.2.1 *Global h*

The global bandwidth parameter was selected by minimizing the LOO-CVE. A plot of the LOO-CVE curve is presented in Figure 6.32 (a). The minimum LOO-CVE was 1.2933 for $h$ = 0.1351 with training MSE = 0.3554 and test MSE = 0.7695. Clearly use of a global bandwidth is useless for input selection as the function approximation presented in Figure 6.32 (b) clearly indicates.

Table 6.5 Summary of results for G10D1_100: 5 relevant and 5 irrelevant input variables.

| Method | Data Set | G10D1_100 |
|---|---|---|
| **Global** | $h \cdot eye(10,10)$ : | $0.0977 \cdot eye(10,10)$ |
| | *LOO-CVE:* | 13.2184 |
| | *MSE(Train):* | 1.3879 |
| | *MSE(Test):* | 9.9720 |
| **CGLS 'best'** | $\begin{bmatrix} h_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h_{2,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_{3,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{4,4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{5,5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_{6,6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_{7,7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{8,8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{9,9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{10,10} \end{bmatrix}$ | $\begin{bmatrix} 0.01096 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0113 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0429 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0128 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1412 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9.4e^{6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.24e^{9} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8324 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0854 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.0e^{10} \end{bmatrix}$ |
| | *LOO-CVE:* | 4.6811 |
| | *MSE(Train):* | 0.8611 |
| | *MSE(Test):* | 5.7377 |
| **DEALKR$_D$** | $\begin{bmatrix} h_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h_{2,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_{3,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{4,4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{5,5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_{6,6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_{7,7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{8,8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{9,9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{10,10} \end{bmatrix}$ | $\begin{bmatrix} 0.01273 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0134 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0497 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0149 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0506 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 381.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 60.68 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 261.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1076 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 94.15 \end{bmatrix}$ |
| | *LOO-CVE:* | 4.1648 |
| | *MSE(Train):* | 0.8429 |
| | *MSE(Test):* | 5.3271 |




(a) (b)

Figure 6.32 Global bandwidth results for Cascade81_100: (a) Plot of LOO-CVE curve. (b) Function approximation of cascade81_100 using the optimal global bandwidth parameter.

### 6.5.2.2 *ModelM and CGLS Results*

ModelM was used to check of local minima. The smoothing parameter histogram presented in Figure 6.33 and standard deviations of the smoothing parameter histogram are large suggesting the presence of multiple minima. The smoothing parameter spectrum for the best set of bandwidth parameters is presented in Figure 6.34 (a) and the resulting function approximation is shown in Figure 6.34 (b). CGLS successfully identified variables 5-8 as insignificant and variables 2-4 as significant. It also identified variable 1 as having little correlation with the target variable. It had a LOO-CVE = 0.1375, a training MSE = 0.0015 and test MSE = 0.0126

### 6.5.2.3 *DEALKR$_D$*

The DEALKR$_D$ smoothing parameter spectrum is presented Figure 6.35 (a). DEALKR$_D$ accurately identified variables 5-8 as irrelevant and 2-4 as relevant. It also identified variable 1 as having very little relevance to predicting the output. The LOO-CVE = 0.1375, training MSE = 0.0014 and test MSE = 0.0298.

A summary of the results for Cascade81_100 is provided in Table 6.6

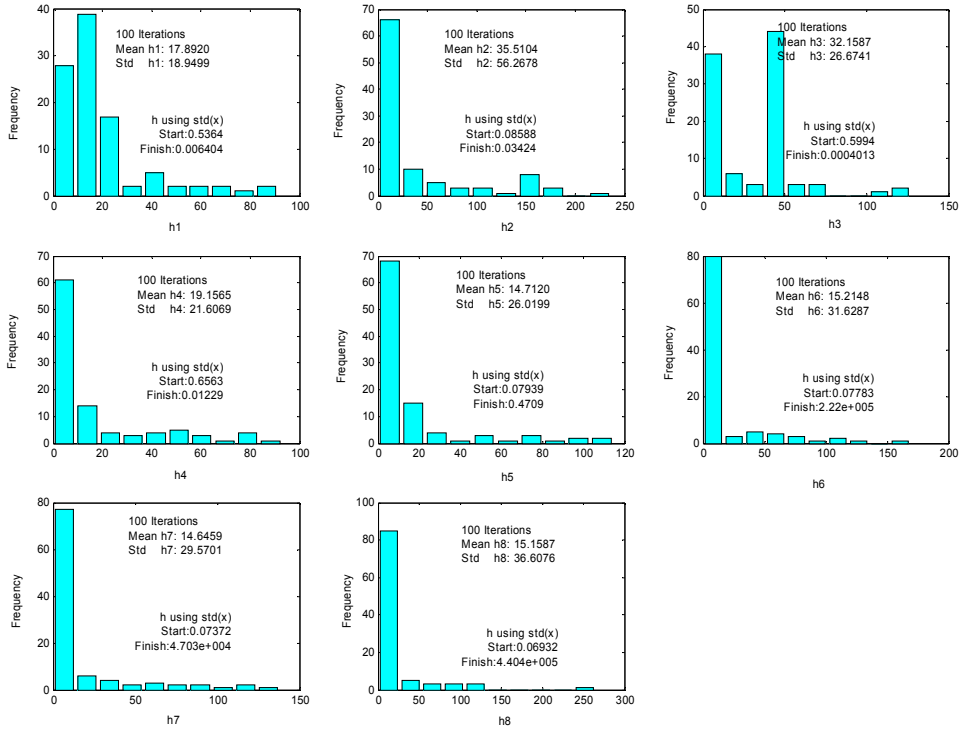



Figure 6.33 Smoothing parameter histogram for cascade81_100 data set obtained by MODLEM.

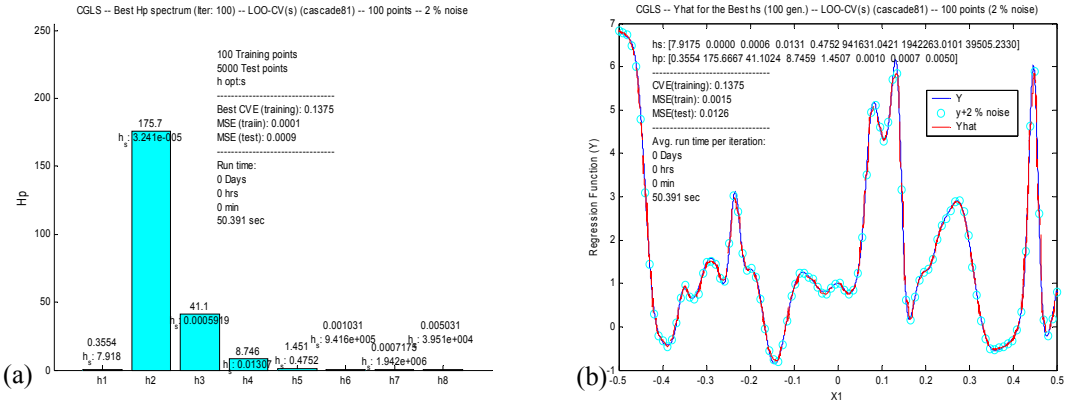



Figure 6.34 (a) Smoothing parameter spectrum for the best set of bandwidth parameters obtained for the cascade81_100 data set by ModelM and CGLS. (b) Plot of the smoothing parameters, LOO-CVE, MSE (training) and MSE (test) for cascade81 obtained by CGLS.

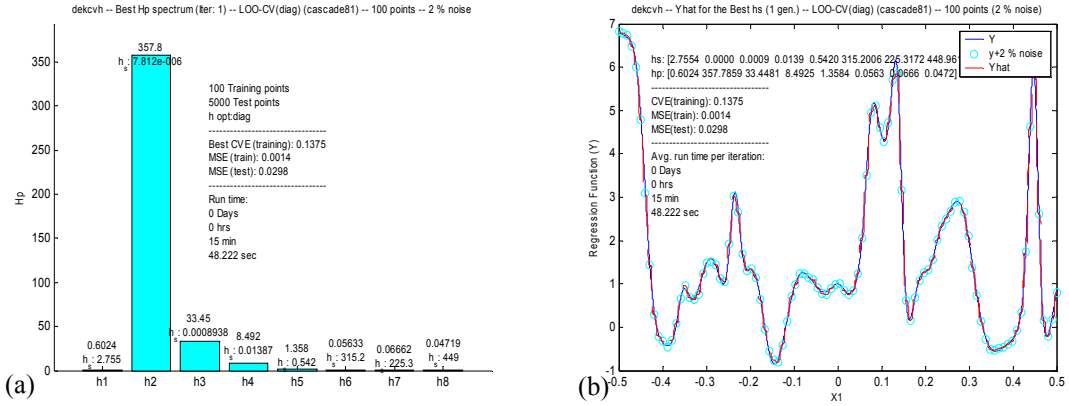



Figure 6.35 DEALKR results for Cascade81_100: (a) Bandwidth spectrum. (b) Function approximation.

Table 6.6 Summary of results for Cascade81_100: 4 relevant and 4 irrelevant input variables.

| **Method** | **Data Set:** | **Cascade81_100** |
|---|---|---|
| **Global** | $h \cdot eye(8,8)$ | $0.1351 \cdot eye(8,8)$ |
| | *LOO-CVE:* | 1.2933 |
| | *MSE(Train):* | 0.3554 |
| | *MSE(Test):* | 0.7695 |
| **CGLS 'best'** | $\begin{bmatrix} h_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h_{2,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_{3,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{4,4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{5,5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_{6,6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_{7,7} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{8,8} \end{bmatrix}$ | $\begin{bmatrix} 7.918 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.24e^{-5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0059 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0131 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4752 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9.4e^{5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.94e^{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.95e^{4} \end{bmatrix}$ |
| | *LOO-CVE:* | 0.1375 |
| | *MSE(Train):* | 0.0015 |
| | *MSE(Test):* | 0.0126 |
| **DEALKR$_D$** | $\begin{bmatrix} h_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h_{2,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_{3,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{4,4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{5,5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_{6,6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_{7,7} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{8,8} \end{bmatrix}$ | $\begin{bmatrix} 2.755 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.8e^{-6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0009 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0139 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.542 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 315.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 225.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 449 \end{bmatrix}$ |
| | *LOO-CVE:* | 0.1375 |
| | *MSE(Train):* | 0.0014 |
| | *MSE(Test):* | 0.0298 |

### 6.5.3 S4G1U10_100: Two Relevant and Eight Irrelevant Variables

Data set S4G1u10_100 has ten input dimensions (2 relevant + 8 irrelevant). It was created by taking the original S4G1_100 data set (see Figure 6.36) and adding 8 noisy input dimensions sampled from a uniform random distribution, $x_3 - x_{10} \sim U([0,1])$.

#### 6.5.3.1 *Global h*

The global bandwidth parameter was selected by minimizing the LOO-CVE. A plot of the LOO-CVE curve is presented in Figure 6.37 (a). The minimum LOO-CVE was 0.2205 for $h = 2.1215$ with a training MSE = 0.1264 and test MSE = 0.2518. Clearly use of a global bandwidth is useless for input selection as the function approximation presented in Figure 6.37 (B).

#### 6.5.3.2 *ModelM and CGLS Results*

ModelM was used with a 10x10 grid of starting points to "map" the LOO-CVE surface and look for local minima. Figure 6.38 (a) shows the smoothing parameter histogram results for ModelM. Notice the spread in the bandwidth spectra and the large standard deviations, both indicators of multiple minima. The bandwidth spectrum for the best set of bandwidths obtained by ModelM is presented in Figure 6.38 (b). CGLS correctly identified variables 1 and 2 as significant but incorrectly included variables 4, 9, and 10. It also incorrectly specified as significant variables 3, 5, 6, 7, and 8. Notice the detrimental affect on the function approximation presented in Figure 6.38 (c), which had a LOO-CVE = 0.1541, a training MSE = 0.0452 and a test MSE = 0.2717.

#### 6.5.3.3 *DEALKR$_D$*

The bandwidth spectrum obtained by DEALKR$_D$ is shown in Figure 6.39 (a). DEALKR$_D$ correctly identified inputs 1 and 2 as relevant and variables 3-10 as irrelevant. The resulting function approximation with a LOO-CVE = 0.0632, a training MSE = 0.0029 and test MSE of 0.0696, appeared to be unscathed by the high level of input noise as indicated by the results in Figure 6.39 (b).

A summary of the results for S4G1u10_100 is provided in Table 6.7.

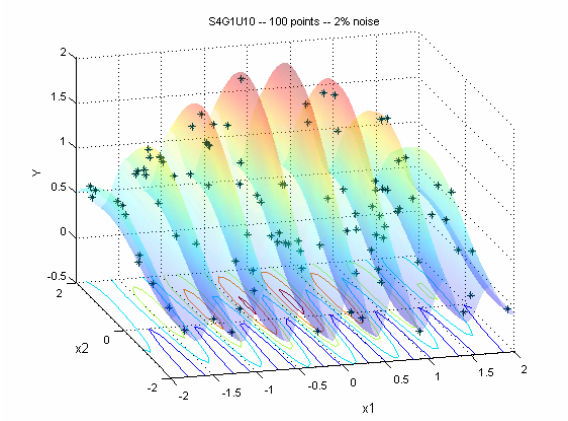



Figure 6.36 Correct response of the two relevant input variables for SG41u10_100 and S4G1u_20_100 data sets.

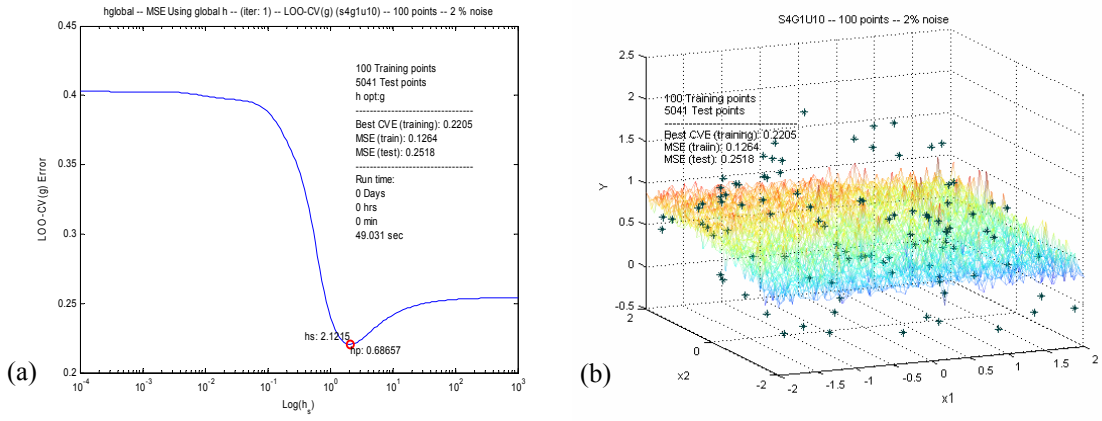
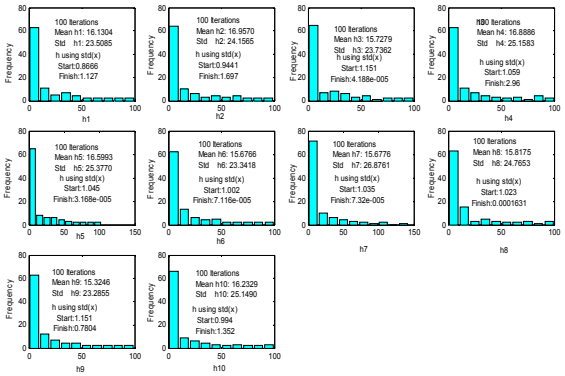
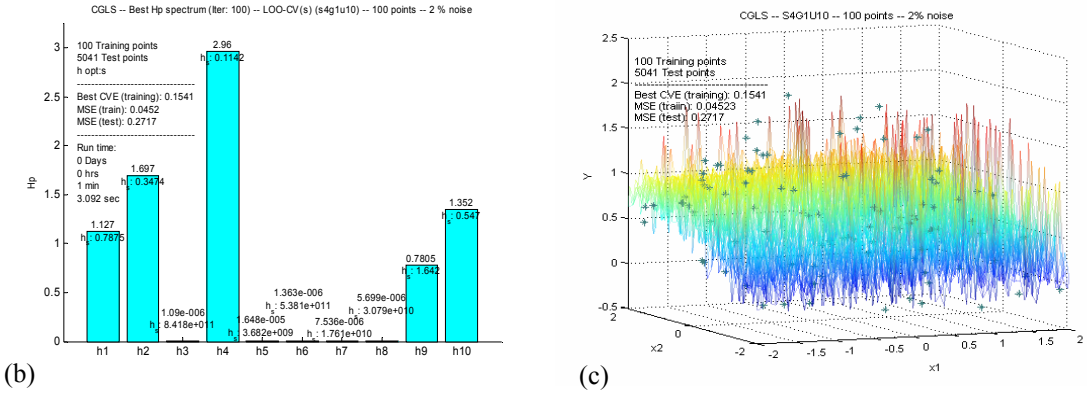



Figure 6.37 Global bandwidth results for S4G1u10_100: (a) LOO-CVE plot. (b) Function approximation contaminated by noisy input variables.

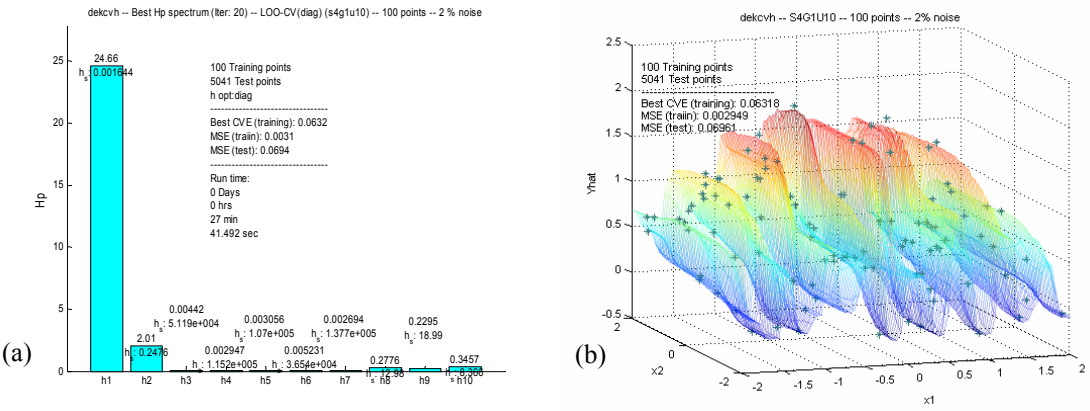


(a)




(b) (c)

Figure 6.38 ModelM results for S4G1u10_100: (a) Bandwidth Histogram. (b) Bandwidths spectrum for best set of bandwidths. (c) Function approximation contaminated by noise from irrelevant inputs.




(a) (b)

Figure 6.39 DEALKR$_D$ results for S4G1u10_100: (a) Bandwidth spectrum correctly identifies 1 and 2 as relevant and 3-10 as irrelevant. (b) Resulting function approximation.

Table 6.7 Summary of results for S4G1U10_100: 2 relevant and 8 irrelevant input variables.

| **Method** | **Data Set:** | **S4G1U10_100** |
|---|---|---|
| **Global** | $h \cdot eye(10,10)$ | $2.1215 \cdot eye(10,10)$ |
| | *LOO-CVE:* | 0.2205 |
| | *MSE(Train):* | 0.1264 |
| | *MSE(Test):* | 0.2518 |
| **CGLS 'best'** | $\begin{bmatrix} h_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h_{2,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_{3,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{4,4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{5,5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_{6,6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_{7,7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{8,8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{9,9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{10,10} \end{bmatrix}$ | $\begin{bmatrix} 0.7875 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3474 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8.4e^{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1142 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.7e^{9} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.4e^{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.8e^{10} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.1e^{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.642 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.547 \end{bmatrix}$ |
| | *LOO-CVE:* | 0.1541 |
| | *MSE(Train):* | 0.0452 |
| | *MSE(Test):* | 0.2717 |
| **DEALKR$_D$** | $\begin{bmatrix} h_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h_{2,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_{3,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{4,4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{5,5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_{6,6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_{7,7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{8,8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{9,9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{10,10} \end{bmatrix}$ | $\begin{bmatrix} 0.0016 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2476 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.1e^{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.2e^{5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.1e^{5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.7e^{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.4e^{5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12.98 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18.99 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8.366 \end{bmatrix}$ |
| | *LOO-CVE:* | 0.06318 |
| | *MSE(Train):* | 0.0029 |
| | *MSE(Test):* | 0.0696 |

### 6.5.4 S4G1U20: Two Relevant and Eighteen Irrelevant Variables

Data set S4G1u20_100 represents and even greater challenge with twenty input dimensions (2 relevant + 18 irrelevant). It was created by taking the original S4G1_100 data set (see Figure 6.36) and adding 18 noisy input dimensions sampled from a uniform random distribution, $x_3 - x_{10} \sim U([0,1])$.

#### 6.5.4.1 *Global h*

The global bandwidth parameter was selected by minimizing the LOO-CVE. A plot of the LOO-CVE curve is presented Figure 6.40 (a). The minimum LOO-CVE was 0.2378 for $h = 3.449$. The function approximation had a LOO-CVE = 0.2378, a training MSE = 0.04517 and test MSE = 0.2691. The affects of the noisy inputs is clearly seen in the plot of the function approximation in Figure 6.40 (b).

#### 6.5.4.2 *ModelM and CGLS Results*

ModelM was used with a 10x10 grid of starting points to "map" the LOO-CVE surface and look for local minima. Figure 6.41 shows the smoothing parameter histogram results for ModelM. Again notice the spread in the bandwidth spectra and the large standard deviations. The bandwidth spectrum for the best set of bandwidths obtained by ModelM is presented in Figure 6.42 (a). In addition to correctly identifying variables 1 and 2 as significant incorrectly included variables 3, 6, 7, 9, 10, 11, 12, 14, 16, 18 and 20. It correctly specified variables 4, 5, 8, 13, 15, 17, and 19 as irrelevant. The impact on the function approximation is even more devastating than in the S4G1u10_100 case. The function approximation shown in Figure 6.42 (a), had a LOO-CVE = 0.2103, a training MSE = 0.0001 and a test MSE = 0.4576.

#### 6.5.4.3 *DEALKR$_D$*

The bandwidth spectrum obtained by DEALKR$_D$ is shown in Figure 6.43 (a). DEALKR$_D$ again correctly identified inputs 1 and 2 as relevant and variables 3-20 as irrelevant. The resulting function approximation had a LOO-CVE = 0.0835, a training MSE = 0.0001 and test MSE = 0.0948 and appeared unaffected by the high level of noisy inputs as indicated by the results in Figure 6.43 (b)

A summary of the results for S4G1u20_100 is provided in Table 6.8

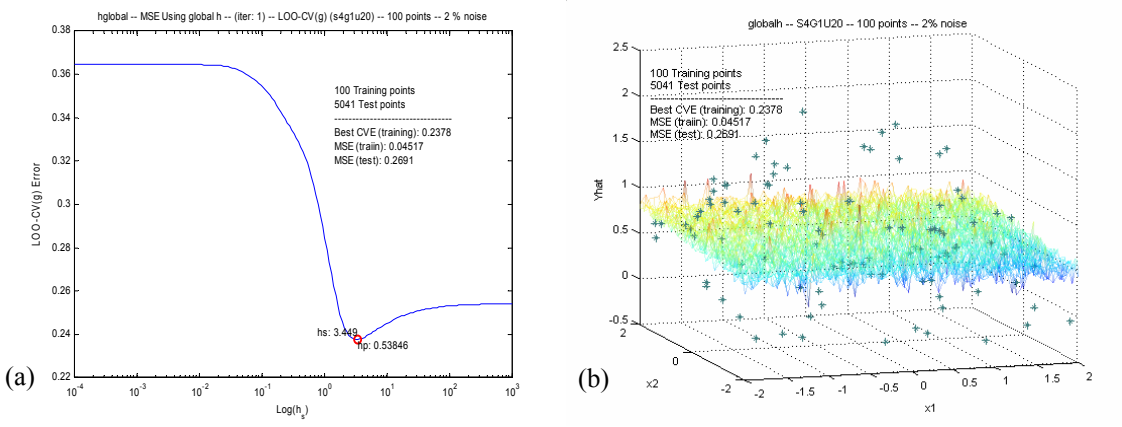



Figure 6.40 Global bandwidth results for S4G1u20_100: (a) LOO-CVE plot. (b) Function approximation contaminated by noisy input variables.

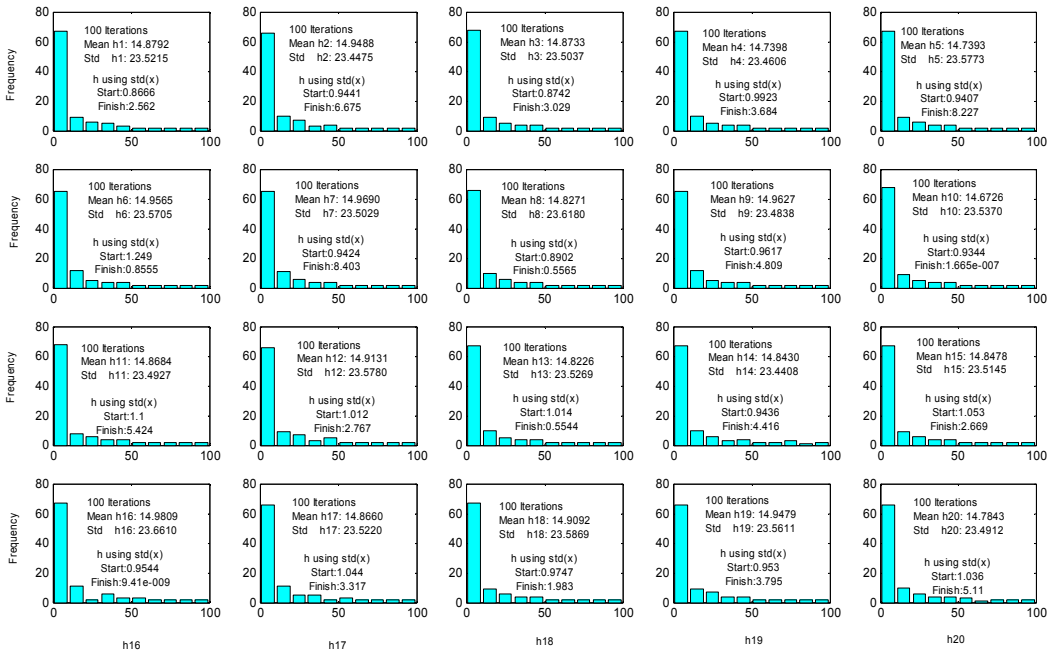
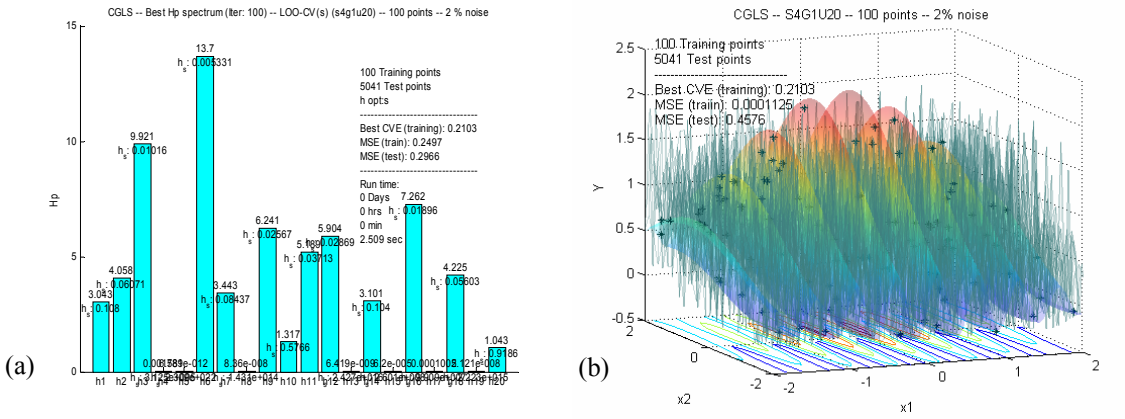
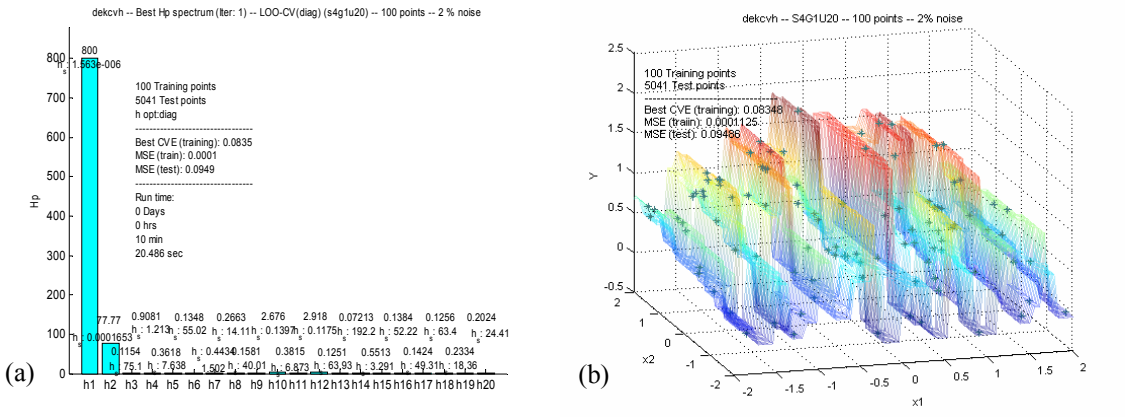



Figure 6.41 Smoothing parameter histograms for S4G1u20 obtained from ModelM showing mean and standard deviations for each smoothing parameter.




(a) (b)

Figure 6.42 ModelM results for S4G1u20_100: (a) Bandwidth spectrum for best set of bandwidths obtained by ModelM. (b) Function approximation contaminated by noise from irrelevant inputs.




(a) (b)

Figure 6.43 DEALKR$_D$ results for S4G1u20_100: (a) Bandwidth spectrum correctly identifies 1 and 2 as relevant and 3-20 as irrelevant. (b) Resulting function approximation.

Table 6.8 Summary of results for S4G1U20_100: 2 relevant and 18 irrelevant input variables.

| Method | Data Set | S4G1U20_100 |
|---|---|---|
| **Global** | *Irrelevant:[3-20]:* | - |
| | *Relevant: [1,2]:* | 1-20 |
| | *LOO-CVE:* | 0.2378 |
| | *MSE(Train):* | 0.04517 |
| | *MSE(Test):* | 0.2691 |
| **CGLS 'best'** | *Irrelevant:[3-20]:* | 4,5,8,13,15,17,19 |
| | *Relevant: [1,2]:* | 3, 6, 7, 9, 10, 11, 12, 14, 16, 18 20 |
| | *LOO-CVE:* | 0.2103 |
| | *MSE(Train):* | 0.0001 |
| | *MSE(Test):* | 0.4576 |
| **DEALKR$_D$** | *Irrelevant:[3-20]:* | 3-20 |
| | *Relevant: [1,2]:* | 1,2 |
| | *LOO-CVE:* | 0.0835 |
| | *MSE(Train):* | 0.0001 |
| | *MSE(Test):* | 0.0948 |

## 6.6 Real World Data

We will now present the performance results on real world data sets. This section starts out describing the superior performance of LRR compared to OLS, and ORR for a prediction task. Specifically it is shown that LRR selectively damps or passes components based on their relative predictive capacity. The remaining sections demonstrate the an inferential sensing capabilities of DEALRR and DEALKR.

### 6.6.1 Automobile Data: Predicting Acceleration

The automobile data set used in this example can be found at the University of California Irvine, Repository of Machine Learning Database [11]. The dataset was first used in the 1983 American Statistical Association Exposition and later used by Quinlan [175] to predict automobile gas mileage. It contains information on 392 automobiles. The seven variables used are listed in Table 6.9. Our goal is to use the first six variables, Figure 6.44 (a), to predict the seventh variable: the car's acceleration, Figure 6.44 (b).

#### 6.6.1.1 *Principle Component Analysis*

Performing a Principal Components Analysis (PCA) on the standardized data results in the amounts of variation incorporated in each the six Principal Components (PC) which are listed along with the singular values in Table 6.10.

Table 6.9: Automobile Data Set.

| | Variable | Type |
|---|---|---|
| 1: | MPG | continuous |
| 2: | Cylinders | multi-valued discrete |
| 3: | Displacement | continuous |
| 4: | Horsepower | continuous |
| 5: | Weight | continuous |
| 6: | Year | multi-valued discrete |
| 7: | Acceleration | continuous |

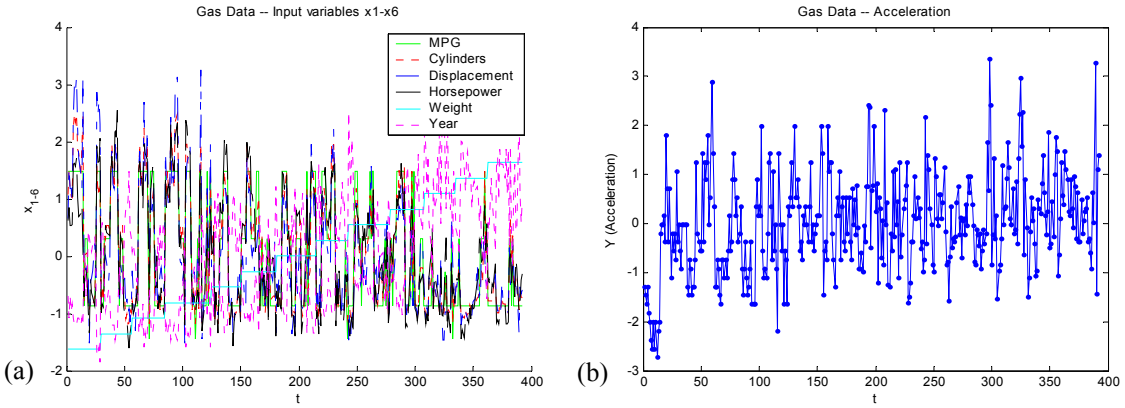



Figure 6.44 (a) Six input variables of Automobile data set. (b) Output, acceleration, for Automobile data set.

Table 6.10: Principal Component Analysis.

| PC | Singular Values | % Variation |
|---|---|---|
| 1: | 42.67 | 77.6 |
| 2: | 18.40 | 14.4 |
| 3: | 9.03 | 3.5 |
| 4: | 7.86 | 2.6 |
| 5: | 5.41 | 1.2 |
| 6" | 3.75 | 0.6 |

Examination of the principal components reveals:

- PC#1 is a weighted average of cylinders, displacement, power, and weight and negatively with MPG.
- PC#2 is weighted towards the year.
- PC#3 is weighted towards MPG.
- PC#4 is a measure of the difference between the two variables cylinders and power.
- PC#5 is weighted towards weight.
- PC#6 is probably mostly noise.

The condition number of X'X, which is the matrix that is inverted when calculating the OLS solution, is slightly ill-conditioned with a condition number of 130.

Table 6.11 presents the results of a correlation analysis of the principal components and the response variable: acceleration. In this table we see that the first component is highly correlated with acceleration and that components 3, 4, and 5 have slight correlations with acceleration. This is to be expected since component two is year, and there were old and new cars with high and low accelerations. We may expect that ridge regression will pass the first five components and that by passing component two, the predictive performance will be slightly degraded.

#### 6.6.1.2 *DEALRR Results*

We will now evaluate various models using Mallows' *CL [DEALRR(CL)], GCV [DEALRR(GCV)], and ICOMPRPS [DEALRR(ICOMPRPS)]* and truncating versions of each *DEALRR(TCL)*, *DEALRR(TGCV)*, and *DEALRR(TICOMPRPS)* as estimators of the predictive error equation (4.2). Specifically, we will evaluate these models against OLS, Principal Component Regression (PCR) with all and various combinations of PCs, and Ridge regression with the regularization component optimized to be 0.8286. This regularization parameter is optimal in the sense that it gives the minimum *CL* value.

Table 6.11 Correlation Analysis.

| Principal Component | Absolute Correlation |
|---|---|
| 1: | 0.5510 |
| 2: | 0.0013 |
| 3: | 0.3625 |
| 4: | 0.3006 |
| 5: | 0.3014 |
| 6: | 0.0598 |

The MSE and MPE results in Table 6.12 show that the model with the minimum MPE value (other than DEALRR) is PCR with components 1, 3, 4, and 5. This agrees with the results expected from the correlation analysis, which expect components 2 and 6 to be removed.

Similar results occur when these predictive models are used with a validation set. In that case, the odd observations are used for training and the even observations are used for calculating the validation error. The predictive error corresponds to the estimates given by DEALRR.

Note that the regularization coefficient of 0.8286 is significantly smaller than each of the singular values, and will therefore pass all of the components. This regularization parameter reduces the condition number from 130 to 120.

Table 6.13 lists the LRR parameters obtained via DEALRR(CL) optimization and their corresponding filter factors. We see that the 2nd component (Year) is properly damped out, and that the last component is partially damped.

Figure 6.45 is a plot of ridge filter factors, localized ridge filter factors, and correlation coefficients. Note the very low correlation of the 2nd component (year) with the response variable. Standard ridge regression allows that component to pass (filter factor near 1) while localized ridge effectively damps it out of the solution (filter factor near 0). Referring to Table 6.12, we see that the DEALRR(CL) solution gives the best *CL* value, which is an estimate of predictive error. Therefore, the DEALRR(CL) outperformed all other linear prediction models for the automobile example.

Table 6.12: Prediction Results.

| **Method** | **MSE** | **MPE (CL)** |
|---|---|---|
| OLS: | *2.8863* | 2.9746 |
| All PCs [1 2 3 4 5 6]: | *2.8863* | 2.9746 |
| One PC [1]: | 5.2868 | 5.3138 |
| [1 3]: | 4.2891 | 4.3329 |
| [1 3 5]: | 3.5995 | 3.6546 |
| [1 3 4 5]: | 2.9134 | 2.9729 |
| Ridge (alpha 0.8286): | 2.8869 | 2.9739 |
| DEALRR(CL): | 2.8885 | *2.9577* |
| DEALRR(GCV): | 2.8886 | 2.9590 |
| DEALRR(ICOMPRPS): | 2.8954 | 3.0009 |
| DEALRR(TCL): | *2.8863* | 2.9599 |
| DEALRR(TGCV): | 2.8874 | 2.9762 |
| DEALRR(TICOMPRPS): | *2.8863* | 2.9891 |

Table 6.13: Localized ridge parameters and filter factors.

| PC | Singular Values | Localized Ridge Parameters | Localized Ridge Filter Factors |
|---|---|---|---|
| 1: | 42.67 | 2.416 | 0.9968 |
| 2: | 18.40 | 8899.3 | 0.0000 |
| 3: | 9.03 | 0.779 | 0.9926 |
| 4: | 7.86 | 0.819 | 0.9893 |
| 5: | 5.41 | 0.562 | 0.9893 |
| 6: | 3.75 | 2.286 | 0.7286 |

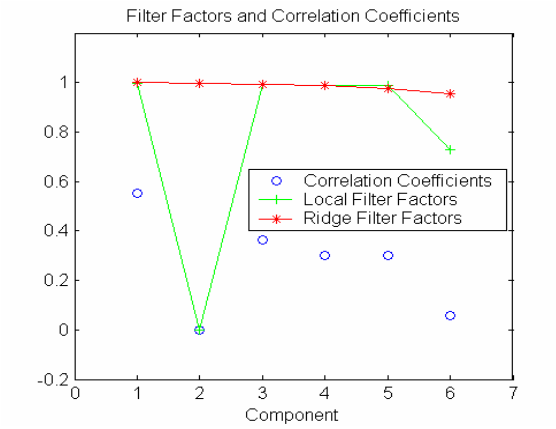



Figure 6.45 Filter Factors produced by DEALRR(CL) and Correlation Coefficients.

### 6.6.2 Inferential Sensing

In this example we report on the ability of DEALRR and DEALKR to perform inferential sensing, which is the prediction of a sensor value through the use of correlated plant variables. To demonstrate this we use eight two variables recorded at a Tennessee Valley Authority (TVA) power plant, arranged in a data matrix **X** (1,000 x 82) as predictor variables to infer a response variable **Y** (1,000 x 1) which the sensor under surveillance. The data set is designated as TVA82Z2.

Two hundred samples were used for training and the entire set was used for testing. The training data is shown in Figure 6.46 (a) and the test data is plotted in Figure 6.46 (b). The data ranges from a high of 1.9814e+003 to a low of -13.0056. Prediction accuracy was estimated for the following: ordinary least squares (OLS), localized ridge regression models: DEALRR(CL), DEALRR(GCV), DEALRR(ICOMPRPS), truncated versions of LRR that completely pass or dump a variable:

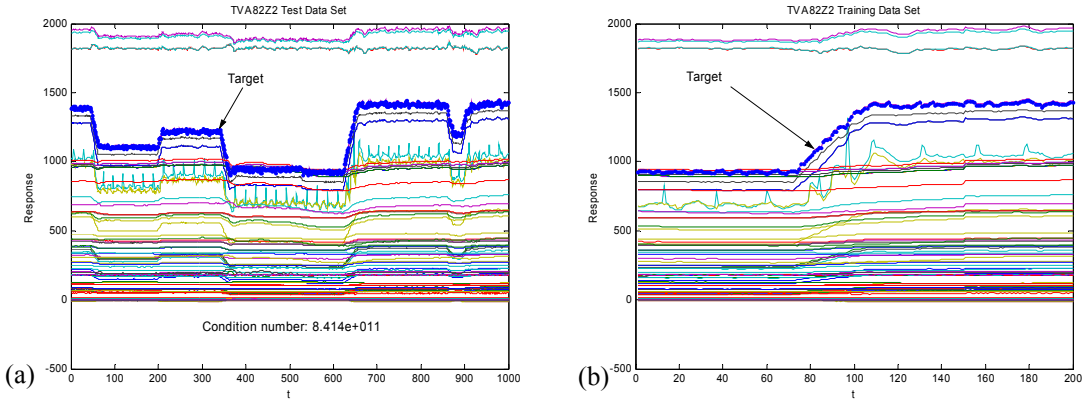



Figure 6.46 TVA82Z2 data set: (a) Test set (1,000 points). (b) Training set (200 points).

DEALRR(TCL),DEALRR(TGCV), DEALRR(TICOMPRPS), localized kernel regression models: global for $\mathbf{H} \in S$, and CGLS and DEALKR for $\mathbf{H} \in D$. TVA82Z2 contains 1,000 samples of 83 sensors, 200 samples were used for training and the full 1,000 for testing. One of the sensors is treated as the target variable, *y*, and the remaining 82 are used as predictor variables, *x*.

The data set is highly collinear as indicated by the large condition numbers: $8.4 \times 10^{11}$ for TVA82Z2(test) and $2.3 \times 10^{12}$ for TVA82Z2(train).

6.6.2.1 *Results*

The results for the truncating and non-truncating case are plotted separately to prevent the plots form being too cluttered. First let us look at the truncating case. The training results are shown in Figure 6.47 (a). All methods appear to have fit the training data well with MSE results ranging from a high of 5.975 for LKR using a global bandwidth to a low of 0.1226 for DEALKR. It is reasonable that DEALKR would have the smallest MSE since the objective function used for training was LOO-CV. The prediction results for the test data are shown in Figure 6.47 (b). The best results were obtained by DEALRR(TICOMPRPS) with a MSE = 18.90 and the worst were for LKR using a global *h* with a MSE=686.3. A summary of the training and test MSE results and a ranking from lowest to highest is contained in Table 6.14

Next let us look at the non truncating versions of the LRR objective functions. The training results are shown in Figure 6.48 (a). Again methods appear to have fit the training data well with MSE results ranging from a high of 5.975 for LKR using a global bandwidth to a low of 0.1226 for DEALKR. The prediction results for the test data are shown in Figure 6.48 (b).

It is worth noticing on the test data, DEALRR(ICOMPRPS), which is a linear technique, outperformed

DEALKR, which is a non-linear technique, with MSE=2.543, and MSE=32.30 respectively. When we examine the training results we see that DEALKR was the best performer with a MSE= 0.1226 and that only DEALKR using a global *h* performed worse than DEALRR(ICOMPRPS) with a MSE=0.9085. This can be attributed to two factors: 1) LOO-CVE does not penalize LKR enough (so even though it is non-linear and should outperform a linear method it ended up fitting the training data too well) and 2) ICOMPRPS provides the highest degree of penalization of the objective functions evaluated, which leads to poorer training results but produces a model that generalizes better. ICOMPRS' greater penalization produces a function that is smoother than those produced by any of the other objective functions, which is why the training MSE is higher. The poorer performance of the other objective functions on the test data indicates that they need to be penalized more (they are under-regularized). This suggests that application of ICOMP to DEALKR would produce very good results for both linear and nonlinear data.

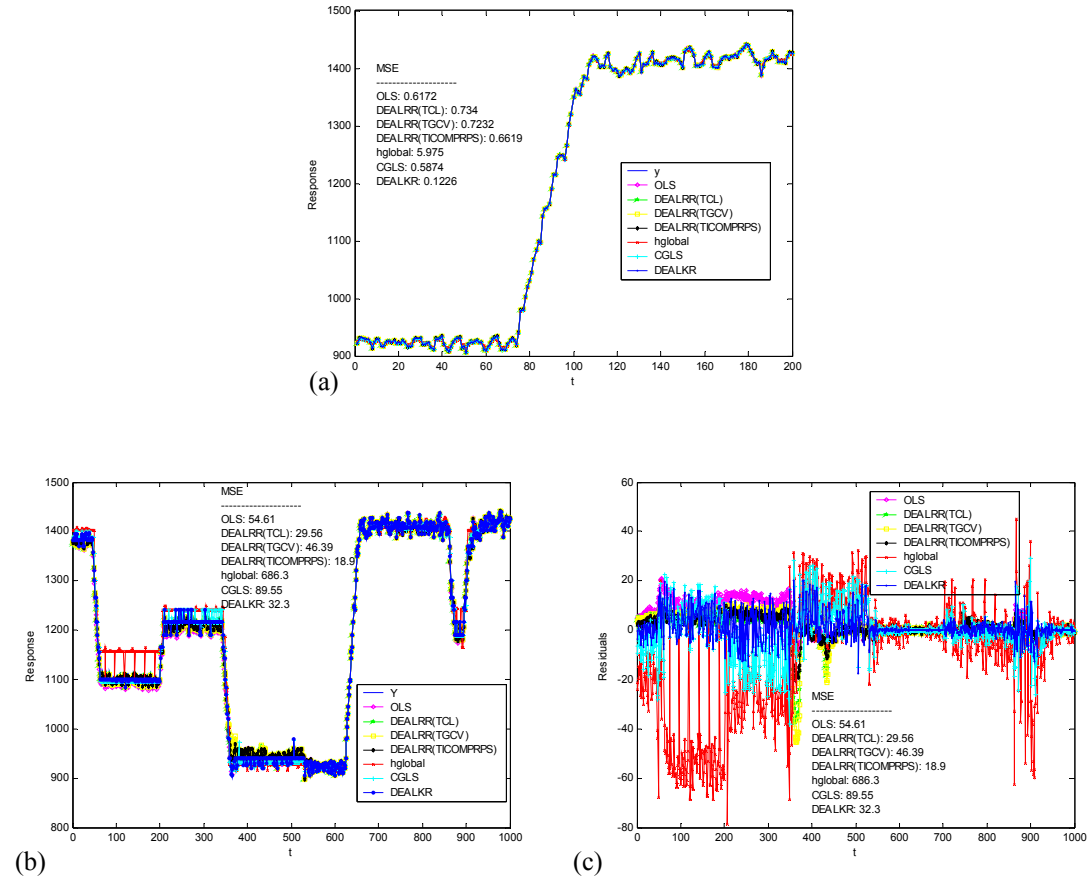



Figure 6.47 TVA82Z2 results for truncating DEALRR objective functions: (a) Training data (200 points). (b) Test data (1,000 points). (c) Test residuals.

Table 6.14: TVA82Z2 Prediction Results Ranking (from 1st to 10th).

| | Training | | Test | |
|---|---|---|---|---|
| **Method** | *Ranking* | *MSE* | *Ranking* | *MSE* |
| OLS | 3rd | 0.6172 | 8th | 54.61 |
| DEALRR(CL): | 6th | 0.7251 | 7th | 47.52 |
| DEALRR(GCV): | 8th | 0.7920 | 4th | 30.41 |
| DEALRR(ICOMPRPS): | 9th | 0.9085 | 1st | 2.543 |
| DEALRR(TCL): | 7th | 0.7340 | 3rd | 29.56 |
| DEALRR(TGCV): | 5th | 0.7232 | 6th | 46.39 |
| DEALRR(TICOMPRPS): | 4th | 0.6619 | 2nd | 18.90 |
| Global *h* | 10th | 5.9750 | 10th | 686.30 |
| CGLS | 2nd | 0.5874 | 9th | 89.55 |
| DEALKR | 1st | 0.1226 | 5th | 32.30 |

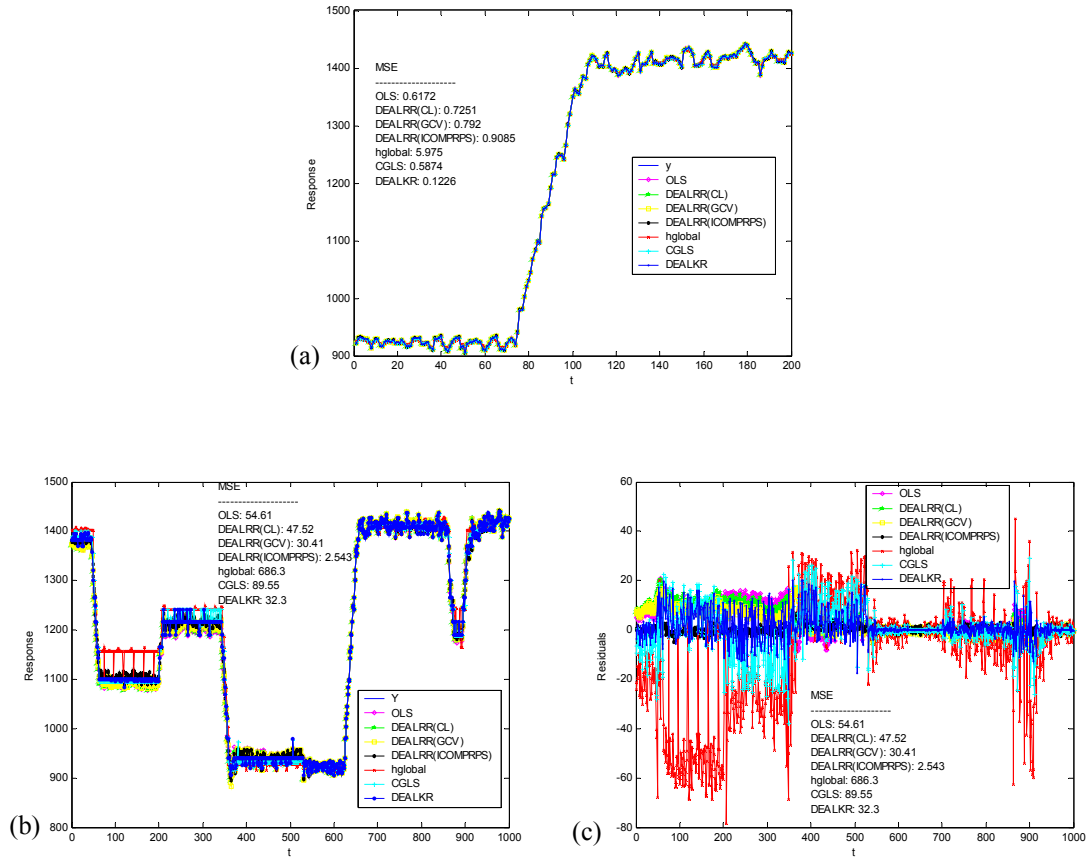



Figure 6.48 TVA82Z2 results for continuous DEALRR objective functions: (a) Training data (200 points). (b) Test data (1,000 points). (c) Test residuals.

Chapter 7

# Conclusion

## 7.1 Summary

This dissertation described the successful application of differential evolution to solving five learning problems: i) the localized ridge regression problem, ii) the localized kernel regression problem, iii) the full bandwidth matrix optimization problem, iv) the relevant predictor variable selection problem, and v) the local minima problem. The effectiveness of the solutions were demonstrated on both artificial and real world data sets.

DEALRR: Differential Evolution automated selection of localized ridge regression parameters was shown to solve the localized ridge regression problem. DEALRR was shown to out perform OLS, ORR, PCR on two real world data sets.

$DEALKR_D$ and $DEALKR_F$: **D**ifferential **E**volution **a**utomated selection of bandwidth parameters for **l**ocalized **k**ernel **r**egression methods (diagonal and full matrix) were demonstrated to solve the localized kernel regression and the optimization of the full bandwidth matrix problems. They were shown to outperform LKR with a global bandwidth or a diagonal bandwidth matrix selected using CGLS for both artificial and real world data.

$DEALKR_D$ was demonstrated as a solution to the relevant predictor variable selection problem. It successfully identified relevant and irrelevant inputs for artificial data set with only 2 relevant inputs and as many as 18 irrelevant inputs.

ModelM: Method of Detecting Local Minima was shown to be successful at identifying the presence of local minima for multi-dimensional data. This technique could be used to not only determine that local minima are present but also provide an estimate of how many there are.

## 7.2 Future work

### 7.2.1 Method of Synchronized Differential Evolution for Multi-parameter Optimization

Synchronized Differential Evolution for Multi-parameter Optimization (SDEMO) would build upon the ideas used in ModelM.: i) a population of seeds values or starting points is created, ii) CGLS then converges from each seed location to a local minimum, ii) the converged minima become the targets of mutation and recombination instead of the initial seed points. By analogy, image attempting to create a contour map of a mountain range. Classical DE would be like air dropping a population of "scouts" that just stand where they land and report their present "elevation". Consequently DE ignores valuable information about the gradient or slope of the surface upon which a "scout" stands. SDEMO will use "scouts" that follow the gradient of the surrounding error surface until they converge to a minimum and then use the "scout" at this elevation (value of the objective function) to create the next generation.

### 7.2.2 Using DE to Optimize its Own Parameters for Truncated Variable Selection

An interesting topic would be the use of DE to select its own parameter settings. One scenario could be as follows:

A truncating version of the DEALRR or DEALKR$_D$ totally passes of dumps an input variable acting as a quick relevance indicator to jump start the more comprehensive optimization (filter factors). It would be interesting to create various sizes of binary sequences (since that is in essence what DE is optimizing in the case of the truncating objective function.). DE could be used to alter the population strategy and values of CR, F, and K, to find the optimal settings, based on the size of the data set and objective vectors being optimized.

### 7.2.3 Apply DEALKR to Localized Polynomial Regression

A promising area of future research would be to extend the modeling capabilities of DEALKR to include localized polynomial regression (LPR), DEALPR. DE would be able to optimize not only the bandwidths of the polynomials but also the optimal polynomial order.

### 7.2.4 Acceleration and Increased Performance

#### 7.2.4.1 *Vectorization*

An immediate improvement in performance could be achieved by rewriting the code to take advantage of vectorization. Preliminary results for a vectorized version of LKR using GCV as an objective function indicate nearly 100-fold increase in speed over an implementation using for loops.

#### 7.2.4.2 *More Efficient Distance Calculation*

Initialize the distance query matrix and then index it for subsequent calls instead of re-computing the distance. This is especially important for larges data sets. It is worth noting that the DE based techniques described in this work are very capable of optimizing higher dimension multivariate data sets, but their speed decreases rapidly as the size of the data set increases. In the present implementation the speed seems to be directly related to continual recalculation of the distance measure performed for every "query" point and for every individual member of the population. This wasteful re-computation is the highest for LOO-CV, and is recalculated for every population member and its associated trial vector repetitively generation after generation. Since it is not the distance among neighbors and "query points" that is changing but the "boundaries: of the neighborhood as the bandwidth parameters are optimized it makes since to compute an initial distance matrix and then index only the "archived" distance data that lie in the neighbor hood defined by the changing kernel bandwidth parameters. One would expect that the increase in performance of this approach while significant for large data sets might be barely detectable for small data sets.

#### 7.2.4.3 *Parallelization*

Parallelization of these techniques is another viable option to increase overall performance and more easily handle larger data sets. DE is well suited to parallel implementation . Parallelization would be especially compelling when computational time is dominated by execution of the objective function. This would enable the execution of multiple simultaneous instance of and would conceivably benefit greatly by use of parallel computing platforms. While there are many possibilities the approach presently being considered is the use of a network cluster.. For the author the move in that direction is motivated by the attractive possibility of continuing the development of this suite of techniques within a native MATLAB environment.

### 7.2.5 Use of other objective functions for DEALKR

It is well know that LOO-CV is computationally intensive and requires large amounts of memory especially if this technique is to be applied to real world industrial data sets. One way to attack the problem would be to incorporate other objective functions such as Mallows' CL, ICOMP and GCV.

#### 7.2.5.1 *ICOMP for LKR and Localized Polynomial Regression*

The real-world data results, which were discussed in section 6.6.2.1, demonstrated the superior performance of ICOMP over the other objective function evaluated and suggest that one of the most promising areas of future research would be the development and application of an ICOMP objective function for DEALKR and DEALPR. The added level of penalization and the fact that DEALPR would be able to perform input variable selection and accurately model both linear and non-linear data.

#### 7.2.5.2 *Multi-objective Optimization*

Another possibility would be to pursue multiple objective criteria or objective functions simultaneously. DE has been successfully applied to multi-objective optimization problems. A suite of multiple fast objective functions that complement and compensate for the short comings in others could be optimized in parallel.

# References

# References

**-A-**


1. Hussein A. Abbass. An evolutionary artificial neural networks approach for breast cancer diagnosis. *Artificial Intelligence in Medicine*, 25(3):265-281, July 2002. http://www.sciencedirect.com/science/article/B6WVB-463WWXD-1T4/1/8030d8779c7ca4b472ef8ea0e54e3bee

2. Advanced methods for non-parametric modeling – Datasets used during the course. http://www.inrialpes.fr/is2/people/goutte/datafun.html

3. H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control,* 19(6):716–723, 1974.

4. Christopher G. Atkeson, Andrew M. Moore, and Stefan Schal. *Locally Weighted Learning*. AI Review, Vol. 11, pages 11-73, April 1997.


**-B-**


5. B.V. Babu and S.A. Munawar. Optimal Design of Shell-and-Tube Heat Exchangers by Different Strategies of Differential Evolution, 2000. http://www.bvbabu.50megs.com/about.html

6. B.V. Babu. Evolutionary Computation - At a Glance. *NEXUS, Annual Magazine of Engineering Technology Association*, BITS, Pilani, pages 3-7, 2001.

7. A.D. Back and T.P. Trappenberg. Selecting inputs for modeling using normalized higher order statistics and independent component analysis. *IEEE Transactions on Neural Networks*, 12(3):612-617, May, 2001.

8. T. Bäck, U. Hammel, and H.P Schwefel. Evolutionary Computation: Comments on the History and Current State. *IEEE Transactions on Evolutionary Computation*, 1(1):3-17, April 1997.

9. R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

10. James O. Berger and C. Srinivasan. Generalized Bayes Estimators in Multivariate Problems. *Annals of Statistics*, 6(4):783-801, July 1978.

11. C.L. Blake and C.J. Merz. UCI Repository of Machine Learning Databases. University of California, Irvine, Department of Information and Computer Science, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html

12. P. Blomgren and T. F. Chan. Modular Solvers for Constrained Image Restoration Problems. Technology Report, Mathematics Department, UCLA, Los Angeles, 1999.

13. P.P. Bonissone. Soft computing: The convergence of emerging reasoning technologies. *Soft Computing. A Fusion of Foundations, Methodologies, and Applications* 1(1):6-18, 1997.

14. B. V. Bonnlander and A. S. Weigend. Selecting input variables using mutual information and nonparametric density estimation. *Proceeding of the 1994 International Symposium on Artificial Neural Networks (ISSAN'94)*, Tainan, Taiwan, pages 42-50, 1994.

15. G. Bontempi, M. Biattari, and H. Bersani. *Lazy Learning: A Logical Method for Supervised Learning.* Pages 97-136.

16. K.M. Bossley. Regularization Theory Applied to Neurofuzzy modeling, Technical Report ISIS-TR3, Department of Electronics and Computer Science, University of Southampton, 1996.

17. L. Breiman; D. Freedman. How Many Variables Should be Entered in a Regression Equation? *Journal of the American Statistical Association*, 78(381):131-136, Mar., 1983. http://links.jstor.org/sici?sici=0162-1459%28198303%2978%3A381%3C131%3AHMVSBE%3E2.0.CO%3B2-H

18. Leo Breiman, Jerome H. Friedman. Predicting Multivariate Responses in Multiple Linear Regression, *Journal of the Royal Statistical Society. Series B (Methodological)*, 59(1):3-54, 1997. http://links.jstor.org/sici?sici=0035-9246%281997%2959%3A1%3C3%3APMRIML%3E2.0.CO%3B2-S

19. M.A. Buckner, A. Gribok, A. Urmanov and J.W. Hines. Application of Generalized Ridge Regression for Nuclear Power Plant Sensor Calibration Monitoring. *5th International Conference on Fuzzy Logic and Intelligent Technologies in Nuclear Science (FLINS)*, Gent, Belgium, September 16-18 2002.

20. Andreas Buja, Trevor Hastie, and Robert Tibshirani. Linear Smoothers and Additive Models, *Annals of Statistics*, 17(2): 453-510, June 1989. http://links.jstor.org/sici?sici=0090-5364%28198906%2917%3A2%3C453%3ALSAAM%3E2.0.CO%3B2-Y

**-C-**

21. Colin Campbell. *Kernel Methods: A Survey of Current Techniques*. http://citeseer.nj.nec.com/campbell00kernel.html

22. Gavin C. Cawley and Nicola L. C. Talbot. Improved sparse least-squares support vector machines, *Neurocomputing*, In Press, Corrected Proof, Available online, July 15 2002.

23. Chris Chatfield. Model Uncertainty, Data Mining and Statistical Inference. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 158(3):419-466, 1995.

24. Vladimir Cherkassky and Filip Mulier. Guest editorial: Vapnik-Chervonenkis (vc) learning theory and its applications. *IEEE Transactions on Neural Networks*, 10(5):985- 987, 1999.

25. S. Chen, C.F.N. Cowan, and P.M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2:302-309, 1991.

26. S. Chen, P. M. Grant, and C.F.N. Cowan. Orthogonal least squares algorithm for training multioutput radial basis function networks. *Proc. Inst. Elect. Eng. F*, 139(6):378-384, 1992.

27. S. Chen, E.S. Chng, and K. Alkadhimi. Regularised orthogonal least squares algorithm for constructing radial basis function networks. *Int. J. Contr.*, 64(5):829-837, 1996.

28. S. Chen, Y. Wu, and B.L. Luk. Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks. IEEE Transactions on Neural Networks, 10(5): 1239–1243, September 1999.

29. S. Chen. Local regularization assisted orthogonal least squares regression. submitted to *IEEE Transactions on Neural Networks*, 2001.

30. Chen, S. Kernel-based data modeling using orthogonal least squares selection with local regularization. *7th Annual Chinese Automation and Computer Science Conference in UK*, pages 27-30, September 2001.

31. S. Chen, X. Hong, and C.J. Harris. Sparse kernel regression modeling using combined locally regularized orthogonal least squares and D-optimality experimental design. *IEEE Transactions on Automatic Control* (submitted for publication in June 2002)

32. S. Chen. Locally regularised orthogonal least squares algorithm for the construction of sparse kernel regression models. *Proceedings, 6th Int. Conference on Signal Processing.* Beijing, China, August 26-30 2002.

33. S. Chen and X. Hong and C.J. Harris (2003) Sparse kernel regression modeling using combined locally regularized orthogonal least squares and D-optimality experimental design. *IEEE Transactions on Automatic Control.*

34. V. Cherkassky, Shao Xuhui, F.M. Mulier, and V.N. Vapnik. Model complexity control for regression using vc generalization bounds. *IEEE Transactions on Neural Networks*, 10(5): 1075-1089, 1999.

35. Ji-Pyng Chiou and Feng-Sheng Wang. Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process. *Computers and Chemical Engineering*, 23(9): 1277-1291, November 1 1999.

36. Carlos A. Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems*, 1(3):269-308, August 1999.

37. Carlos A. Coello and Aguirre Arturo Hernández. Evolutionary Multiobjective Design of Combinational Logic Circuits. *Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware, IEEE Computer Society*, Jason Lohn, Adrian Stoica, Didier Keymeulen & Silvano Colombano (editors), Los Alamitos, California, pages 161-170, July 2000.

38. David Corne, Marco Dorigo and Fred Glover. *New Ideas in Optimization.* McGraw-Hill, 1999.

39. P.L. Cornelius and J. Byars. Computing Iterative Weighted Least Squares Estimates of Genetic Variance Components. *Biometrics*, 33(2):375-382, June 1977. http://links.jstor.org/sici?sici=0006-341X%28197706%2933%3A2%3C375%3ACIWLSE%3E2.0.CO%3B2-L

40. Christopher Cox and Guangqin Ma. Asymptotic Confidence Bands for Generalized Nonlinear Regression Models. *Biometrics*, 51(1):142-150, March 1995.

http://links.jstor.org/sici?sici=0006-341X%28199503%2951%3A1%3C142%3AACBFGN%3E2.0.CO%3B2-F

41. D. R. Cox and Nanny Wermuth. An Approximation to Maximum Likelihood Estimates in Reduced Models. *Biometrika*, 77(4):747-761, December 1990. http://links.jstor.org/sici?sici=0006-3444%28199012%2977%3A4%3C747%3AAATMLE%3E2.0.CO%3B2-F

42. Dennis D. Cox and Finbarr O'Sullivan. Asymptotic Analysis of Penalized Likelihood and Related Estimators. *Annals of Statistics*, 18(4):1676-1695, December 1990. http://links.jstor.org/sici?sici=0090-5364%28199012%2918%3A4%3C1676%3AAAOPLA%3E2.0.CO%3B2-V

43. D. Dennis and Isabel Llatas Cox. Maximum Likelihood Type Estimation for Nearly Nonstationary Autoregressive Time Series. *Annals of Statistics*, Vol. 19, No. 3. (Sep., 1991), pp. 1109-1128. http://links.jstor.org/sici?sici=0090-5364%28199109%2919%3A3%3C1109%3AMLTEFN%3E2.0.CO%3B2-%23

**-D-**

44. K.A. De Jong. *An Analysis of the behavior of a class of genetic adaptive systems.* University of Michigan, Ann Arbor. (University Microfilms No. 76-9381), 1975.

45. Arthur P. Dempster, Chandu M. Patel, Murray R. Selwyn and Arthur J. Roth. Statistical and Computational Aspects of Mixed Model Analysis. *Applied Statistics*, 33(2):203-214, 1984. <http://links.jstor.org/sici?sici=0035-9254%281984%2933%3A2%3C203%3ASACAOM%3E2.0.CO%3B2-2>

46. Wen-Shuenn Deng, Chih-Kang Chu and Ming-Yen Cheng. A study of local linear ridge regression estimators. *Journal of Statistical Planning and Inference*, 93(1-2):225-238, February 2001.

47. L. Desbat and D. Girard. The "minimum reconstruction error" choice of regularization parameters: Some more scientific methods and their application to deconvolution problems. *SIAM J. of Scientific Computing*, 16:1387–1403, 1995.

48. David Draper. Assessment and Propagation of Model Uncertainty. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):45-97, 1995.

49. R. Dunia, S.J. Qin, T.F. Edgar and T.J. McAvoy. Sensor fault identification and reconstruction using principal component analysis. *Proc. of IFAC Congress'96*, *Vol. N*, San Francisco, pages 259-264, June 30 - July 5 1996.

**-E-**

50. R.C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micromachine and Human Science,* IEEE service center, Piscataway, NJ, Nagoya, Japan, pages 39-43, 1995.

51. Bradley Efron and Robert Tibshirani. Nonparametric Function Estimation: General Methodology Using Specially Designed Exponential Families for Density Estimation. *Annals of Statistics*, 24(6):2431-2461, December 1996.

52. R.C. Elston. On the Correlation Between Correlations. *Biometrika*, 62(1):133-140, April 1975. <http://links.jstor.org/sici?sici=0006-3444%28197504%2962%3A1%3C133%3AOTCBC%3E2.0.CO%3B2-S>

53. D. A. Elston and M. F. Proe. Smoothing Regression Coefficients in an Overspecified Regression Model with Interrelated Explanatory Variables. Applied Statistics, 44(3):395-406, 1995.

54. H. W. Engl and W. Grever. Using the L-curve for determining optimal regularization parameters, *Numerical Mathematics*, 69:25–31, 1994.

55. Robert F. Engle and Simone Manganelli. CAViaR: Conditional Autoregressive Value at Risk by Regression Quantiles. UCSD Economics Discussion Paper 1999-2000, University of California, San Diego, Department of Economics. October 1999. <ftp://weber.ucsd.edu/pub/econlib/dpapers/ucsd9920.pdf>

56. Evolutionary computing in a nutshell. *European Network of Excellence in Evolutionary Computing, EvoNet*. <http://evonet.dcs.napier.ac.uk/evoweb/resources/ec_nutshell/index.html>

**-F-**

57. Manfred M. Fischer, Martin Reismann and Katerina Hlavackova-Schindler. Parameter estimation in neural spatial interaction modeling by a derivative free global optimization method. *The IV International Conference on GeoComputation,* Mary Washington College, Fredericksburg, VA, USA. 1999.

58. H. E. Fleming. Equivalence of regularization and truncated iteration in the solution of ill-posed image reconstruction problems. *Linear Algebra Applications,* 130:133–150, 1990.

59. L.J. Fogel, A.J. Owens and M.J. Walsh. *Artificial Intelligence through Simulated Evolution.* New York:Wiley, 1966.

60. D.B. Fogel. An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks*, 5:-14, 1994.

61. D.B. Fogel. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, Piscataway, NJ, 1995.

62. Jerome H. Friedman and Werner Stuetzle. Projection Pursuit Regression. *Journal of the American Statistical Association (in Theory and Methods),* 76(376):817-823, December, 1981. <http://links.jstor.org/sici?sici=0162-1459%28198112%2976%3A376%3C817%3APPR%3E2.0.CO%3B2-1>

63. Jerome H. Friedman. Multivariate Adaptive Regression Splines. *Annals of Statistics (Invited Paper),* 19(1): 1-67 March, 1991. <http://links.jstor.org/sici?sici=0090-5364%28199103%2919%3A1%3C1%3AMARS%3E2.0.CO%3B2-D>

**-G-**

64. D. Gamberger and N. Lavrac. Conditions for Occam's Razor Applicability and Noise Elimination. *European Conference on Machine Learning*, 1997.

65. S. Geman, E. Bienenstock and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1-58, 1992.

66. F. Girosi, M. Jones and T. Poggio. Regularization theory and neural-network architectures, *Neural Computation*, 7:219-269, 1995.

67. D.E. Goldberg. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley, 1989.

68. M. Goldstein and A.F.M. Smith. Ridge-type estimators for Regression Analysis. J.R. *Statist. Soc. B,* 36:284-291, 1974.

69. M. Goldstein. Bayesian Analysis of Regression Problems. *Biometrika*, 63(1):51-58, April, 1976. http://links.jstor.org/sici?sici=0006-3444%28197604%2963%3A1%3C51%3ABAORP%3E2.0.CO%3B2-1

70. G.H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge Parameter. *Technometrics,* 21(2):215-223, 1979.

71. I.J. Good and R.A. Gaskins. Nonparametric Roughness Penalties for Probability Densities. *Biometrika*, 58(2):255-277, August, 1971.

72. Cyril Goutte. *Statistical Learning and Regularization for Regression: Application to System Identification and Time Series Modeling*. PhD thesis, Connectionist Group, LIP6, Pierre and Marie Curie University – Paris, Technical report no. 1997/033, 1997. http://www.ei.dtu.dk/staff/goutte/PUBLIS/thesis.html

73. Cyril Goutte and Jan Larsen. *Adaptive metric kernel regression*. *Neural Networks for Signal Processing VIII -- Proceedings of the 1998 IEEE Workshop, VIII*, (in press, Piscataway, New Jersey, 1998.

74. Cyril Goutte and Jan Larsen. Adaptive metric kernel regression, *The Journal of VLSI Signal Processing*, 26(1/2):155-67, 2000. http://www.kap.nl/journalhome.htm/0922-5773

75. Cyril Goutte. Personal Correspondence on July 20, 2002. Dr. Goutte provided a URL for obtaining the AMKREG code: Data and Matlab functions for the course on advanced methods for non-parametric modeling. http://www.inrialpes.fr/is2/people/goutte/datafun.html

76. *16 Attributes of a System for Automatic Programming.* http://www.genetic-programming.com/attributes.html

77. Seven Differences Between Genetic Programming and Other Approaches to Machine Learning and Artificial Intelligence. http://www.genetic-programming.com/sevendiffs.html

78. P.J. Green. Penalized likelihood for general semi-parametric regression models, *Int. Statist. Rev*, 55:245-259, 1987.

79. A.V. Gribok, I. Attieh, J.W. Hines and R.E. Uhrig. Regularization of feedwater flow rate evaluation for venturi meter fouling problems in nuclear power plants. *Nuclear Technology,* 134:3-14, 2001.

80. A.V. Gribok, I. Attieh, J.W. Hines, R.E. and Uhrig. Stochastic Regularization of Feedwater Flow Rate Evaluation for Venturi Meter Fouling Problems in Nuclear Power Plants. *Inverse Problems in Engineering,* 00:1-26, 2001.

81. A.V. Gribok, J.W. Hines, A. Urmanov and R.E. Uhrig. Heuristic, systematic, and informational regularization for process monitoring. [accepted for publication in the special issue of], *International Journal of Intelligent Systems on Intelligent Systems for Process Monitoring,* Wiley Publishers, 2002.

82. W. Groetsch. Theory of Tikhonov Regularization for Fredholm Equations of the First Kind. Pitman, Boston, 1984.

**-H-**

83. J. Hadamard. Sur les problemes aux derivees parielies et leur signification physique. *Bull. Univ. of Princeton*, pages 49-52, 1902.

84. J. Hadamard, Lectures on the Cauchy Problem in Linear Partial Differential Equations. *Yale University Press*, 1923.

85. Peter Hall and D.M. Titterington. Common Structure of Techniques for Choosing Smoothing Parameters in Regression Problems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 49(2):184-198, 1987. http://links.jstor.org/sici?sici=0035-9246%281987%2949%3A2%3C184%3ACSOTFC%3E2.0.CO%3B2-4

86. P.C. Hansen. Regularization, GSVD and Truncated GSVD. *BIT*, 29:491-504, 1989.

87. Per Christian Hansen. Analysis of Discrete Ill-Posed Problems by Means of the L-Curve, *SIAM Review*, 34(4):561-580, December, 1992. http://links.jstor.org/sici?sici=0036-1445%28199212%2934%3A4%3C561%3AAODIPB%3E2.0.CO%3B2-J

88. P. C. Hansen and D. P. O'Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.*, 14:1487–1503, 1993.

89. P.C. Hansen. The L-curve and its use in the numerical treatment of inverse problems. invited paper for P. Johnston (Ed.), *Computational Inverse Problems in Electrocardiology*, WIT Press, Southampton, pages 119-142, 2001.

90. Trevor Hastie and Robert Tibshirani. Generalized Additive Models, *Statistical Science*, 1(3):297-310, August 1986. http://links.jstor.org/sici?sici=0883-4237%28198608%291%3A3%3C297%3AGAM%3E2.0.CO%3B2-Z

91. Trevor Hastie and Robert Tibshirani, Generalized Additive Models: Some Applications. *Journal of the American Statistical Association*, (in Applications), 82(398): 371-386, June 1987.

http://links.jstor.org/sici?sici=0162-1459%28198706%2982%3A398%3C371%3AGAMSA%3E2.0.CO%3B2-N

92. Trevor Hastie, Robert Tibshirani and Andreas Buja. Flexible Discriminant Analysis by Optimal Scoring. *Journal of the American Statistical Association*, (in Theory and Methods), 89(428): 1255-1270, December, 1994. http://links.jstor.org/sici?sici=0162-1459%28199412%2989%3A428%3C1255%3AFDABOS%3E2.0.CO%3B2-H

93. Trevor Hastie, Andreas Buja and Robert Tibshirani. Penalized Discriminant Analysis. *Annals of Statistics*, (in Multivariate Analysis), 23(1):73-102, February, 1995. http://links.jstor.org/sici?sici=0090-5364%28199502%2923%3A1%3C73%3APDA%3E2.0.CO%3B2-Z

94. T. Hastie, R. Tibshirani and J. Froedman. The Elements of Statistical Learning; Data Mining, Inference, and Predictions. Springer-Verlag, New York 2001.

95. Dollena S. Hawkins, David M. Allen and Arnold J. Stromberg. Determining the number of components in mixtures of linear models. *Computational Statistics & Data Analysis,* 38(1): 15-48, November 28 2001. http://www.sciencedirect.com/science/article/B6V8V-44B1W87-2/1/788c9c2839fdef7a21c96746f98ce4ad)

96. Douglas M. Hawkins and Xiangrong Yin. A faster algorithm for ridge regression of reduced rank data. *Computational Statistics and Data Analysis*, In Press, Uncorrected proof available online March 15 2002. http://www.sciencedirect.com/science/article/B6V8V-45C03KG-2/1/fe4d73d6cf25323e2e478447abbd5b53

97. Tim Hendtlass. A Combined Swarm Differential Evolution Algorithm for Optimization Problems. *Lecture Notes in Computer Science*, 2070:11-18, Springer-Verlag. ISSN:0302-9743, 2001.

98. Jörg Heitkötter and David Beasley. *The Hitch-Hiker's Guide to Evolutionary Computation: A list of Frequently Asked Questions (FAQ).* USENET:comp.ai.genetic, 1997. ftp://rtfm.mit.edu:/pub/usenet/news.answers/ai-faq/genetic/

99. J.W Hines and R.E. Uhrig. Matlab Supplement to Fuzzy and Neural Approaches in Engineering. *Adaptive and Learning Systems for Signal Processing, Communications and Control Series*, John Wiley & Sons, 1997.

100. J.W. Hines, A.V. Gribok, I. Attieh, and R.E. Uhrig. Regularization methods for inferential sensing in nuclear power plants. *Fuzzy Systems and Soft Computing in Nuclear Engineering*, Ed. Da Ruan, Springer, 1999.

101. J.W. Hines, A. Gribok, A. Urmanov and M.A. Buckner. Selection of Multiple Regularization Parameters in Local Ridge Regression Using Evolutionary Algorithms and Prediction Risk Optimization. *4th International Conference on Inverse Problems in Engineering*, Rio de Janeiro, Brazil, 2002.

102. J.W. Hines, A. Gribok, A. Urmanov and M. Buckner. Selection of Multiple Regularization Parameters in Local Ridge Regression Using Evolutionary Algorithms and Prediction Risk Optimization. *Inverse Problems in Engineering* (accepted for publication), 2002.

103. R.R. Hocking and M.J. Lynn. A Class of Biased Estimators in Linear Regression. *Technometrics*, 18:425-438, November 1976.

104. A.E. Hoerl and R.W. Kennard. Ridge regression: biased estimation for Nonorthogonal problems, *Technometrics*, 12(1):55-82, 1970.

105. Arthur E. Hoerl, Robert W. Kennard and Roger W. Hoerl. Practical Use of Ridge Regression: A Challenge Met. *Applied Statistics*, 34(2):114-120, 1985. http://links.jstor.org/sici?sici=0035-9254%281985%2934%3A2%3C114%3APUORRA%3E2.0.CO%3B2-J

106. F. Hoffmann, C. Isik and V. Zacharias. Learning and Adaptation in Fuzzy Control: Soft Computing Techniques for the Design of Intelligent Systems. John Wiley & Sons, Inc., New York, 2001.

107. J.H. Holland. *Adaptation in natural and artificial systems.* Ann Arbor: The University of Michigan Press, 1975.

108. X. Hong, S. Chen and P.M. Sharkey. Automatic kernel regression modeling using combined PRESS statistic and regularized orthogonal least squares. *IEEE Proceedings Vision, Image and Signal Processing,* 2003.

109. K. Hornick, M. Stinchcombe and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359-366, 1989.

110. C.R. Houck, J.A. Joines and M.G. Kay. *A genetic algorithm for function optimization: A Matlab implementation.* NCSU-IE Technical Report 95-09, 1995.

111. J. Huang, D. Brennan, L. Sattler, J. Alderman, B. Lane and C. O'Mathuna. A comparison of calibration methods based on calibration data size and robustness. *Chemometrics and Intelligent Laboratory Systems*, 62(1):25-35, April 28 2002. http://www.sciencedirect.com/science/article/B6TFP-44NM16C-2/1/7a8e015140d1e98d3af5ea3798afa36b

112. Hsuan-Jui Huang and Feng-Sheng Wang. Fuzzy decision-making design of chemical plant using mixed-integer hybrid differential evolution. *Computers & Chemical Engineering* (in press, uncorrected proof available online). http://www.sciencedirect.com/science/article/B6TFT-45VCGP2-1/1/810babcc40212f92f93c0bbac0739bcd

**-I-**

**-J-**

113. Joines, J. A. and C.T. Culbreth. Job Sequencing and Inventory Control for a Parallel Machine Problem: A Hybrid-GA Approach. *Proceedings of 1999 IEEE Conference on Evolutionary Computation*, ed. D. Fogel and Z. Michalewicz, pages 579-585, Piscataway, New Jersey: Institute of

Electrical and Electronics Engineers, 1999.
http://www.ie.ncsu.edu/joines/papers/papers.html#IEEE99

114. Lee K. Jones. A Simple Lemma on Greedy Approximation in Hilbert Space and Convergence Rates for Projection Pursuit Regression and Neural Network Training. *Annals of Statistics*, *(in Short Communications),* 20(1): 608-613, March 1992.
http://links.jstor.org/sici?sici=0090-5364%28199203%2920%3A1%3C608%3AASLOGA%3E2.0.CO%3B2-7

**-K-**

115. L. Kaufman and A. Neumaier. Regularization of ill-posed problems by envelope guided conjugate gradients. *J. Comput. Graph. Stat.*, 6:451-463, 1997.

116. Vojislav Kecman. Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models, The MIT Press, Cambridge, MA, 2001.

117. Peter J. Kempthorne. Admissible Variable-Selection Procedures When Fitting Regression Models by Least Squares for Prediction. *Biometrika*, 71(3):593-597, Dec., 1984.

118. J. Kennedy and R.C. Eberhart. Particle Swarm Optimization. *Proc. IEEE International Conference on Neural Networks*, Perth Australia, IEEE Service Centre, Piscataway NJ USA IV, pages 1942-1948, 1995.

119. J. Kennedy and R.C. Eberhart. The Particle Swarm: Social Adaptation in Information-Processing Systems. Chapter 25 in *New Ideas in Optimization.* David Corne, Marco Dorigo and Fred Glover Editors, McGraw-Hill Publishing Company, England, ISBN:007-709506-5, 1999.

120. M. E. Kilmer and D. P. O'Leary. *Choosing regularization parameters in iterative methods for ill-posed problems.* Tech. Report CS-TR-3937, Computer Science Department, University of Maryland, October 1998.
http://citeseer.nj.nec.com/kilmer98choosing.html

121. S. Kirkpatrick, C.D. Gelatt and M.P. Vechhi. Optimization by Simulated Annealing. *Science,* 220(4568):671-680, 1983.

122. John R. Koza. *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems.* Stanford University Computer Science Department technical report STAN-CS-90-1314, June 1990.

123. John R. Koza. The genetic programming paradigm: Genetically breeding populations of computer programs to solve problems. In Soucek, Branko and the IRIS Group (editors). *Dynamic, Genetic, and Chaotic Programming*. New York: John Wiley, pages 203-321, 1992.

124. John R. Koza, William Mydlowec, Guido Lanza, Jessen Yu and Martin A. Keane. *Reverse engineering of metabolic pathways from observed data using genetic programming.* 2000.
http://www.genetic-programming.com/psb2001.ps

125. John R. Koza, Forrest H. Bennett III, David Andre and Martin A. Keane. Genetic Programming: Biologically Inspired Computation that Creatively Solves Non-Trivial Problems. *Evolution as*

*Computation, DIMACS Workshop, Princeton,* Laura F. Landweber and Erik Winfree (Editors), Heidelberg: Springer-Verlag, ISBN: 3-540-66709-1, pages 15– 44, January 1999.

126. K. Krishnakumar. Micro-genetic algorithms for stationary and nonstationary function optimization. *Proc. SPIE Intell. Cont. Adapt. Syst.*, 1196:289-296, 1989.

**-L-**

127. Chia-Jui Lai and C. K. Chu. Triple smoothing estimation of the regression function and its derivatives in nonparametric regression. *Journal of Statistical Planning and Inference*, 98(1-2): 157-175, October 1 2001.
http://www.sciencedirect.com/science/article/B6V0M-43P43YN-D/1/e580c97a3e4b930dfe2436a5b6aea747

128. J. Lampinen and I. Zelinka. On stagnation of the differential evolution algorithm. *Proceedings of MENDEL 2000, 6th International Mendel Conference on Soft Computing*, Brno, Czech Republic, pages 76–83, 2000.

129. J. Lampinen. Solving problems subject to multiple nonlinear constraints by the differential evolution. *Proceedings of MENDEL 2001, 7th International Conference on Soft Computing*, Brno, Czech Republic, pages 50-57, 2001.

130. J. Lampinen. A bibliography of differential evolution algorithm, technical report, Lappeenranta University of Technology, 2001.
http://www.lut.fi/~jlampine/debiblio.htm

131. Peter Lancaster and Kestutis Salkauskas. *Curve and Surface Fitting: An Introduction.* Academic Press Inc., 1986.

132. C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*, Prentice-Hall, pages 183-194, 1994.

133. Michael LeBlanc, Robert Tibshirani. Combining Estimates in Regression and Classification. *Journal of the American Statistical Association, (in Theory and Methods ),* 91(436): 1641-1650, December 1996.
http://links.jstor.org/sici?sici=0162-1459%28199612%2991%3A436%3C1641%3ACEIRAC%3E2.0.CO%3B2-F

134. A.S. Leonov and A.G. Yagola. The L-curve method always introduces a non-removable systematic error. *Moscow University Physics Bulletin*, 52(6):20-23, 1997.

135. Stan Lipovetsky and W. Michael Conklin. Multiobjective regression modifications for collinearity. *Computers & Operations Research*, 28(13):1333-1345, November 2001.
http://www.sciencedirect.com/science/article/B6VC5-43KJMRG-6/1/ecf7a8585f30ac02836cbdc0d537be15

136. C. Loader. *Old Faithful Erupts: Bandwidth Selection Reviewed*. Technical Report 95.9. AT&T Bell Laboratories, Statistics Department. 1995.
http://cm.bell-labs.com/cm/ms/departments/sia/doc/meth.html

**-M-**

137. C.L. Mallows, Some comments on CP. *Technometrics*, 15(4):661-675, 1973.

138. V. Maniezzo. Genetic evolution of the topology and weight distribution of neural networks. *IEEE Transactions on Neural Networks*, 5:39-53, 1994.

139. J. Marron and D. Nolan. Canonical kernels for density estimation. *Statistics & Probability Letters*, 7(3):195-199, 1988.

140. Timothy Masters and Walker Land. A new training algorithm for the general regression neural network. *1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, 3:1990–1994, 1997.

141. James McNames. *Innovations in Local Modeling for Time Series Prediction*. PhD Dissertation. Dept. of Elect. Eng. Stanford University, May 1999.

142. Z. Michalewicz and M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation Journal,* 4:1–32, 1996.

143. W. L. Miller. Measures of Electoral Change Using Aggregate Data. *Journal of the Royal Statistical Society. Series A (General)*, 135(1): 122-142, 1972. http://links.jstor.org/sici?sici=0035-9238%281972%29135%3A1%3C122%3AMOECUA%3E2.0.CO%3B2-7

144. Andrew. M Moore. *Instance-based learning (aka Case-based or Memory-based or non-parametric) Tutorial Slides*. Oct. 25, 2001. http://www-2.cs.cmu.edu/~awm/tutorials/mbl.html

145. D. J. Montana and L. Davis. Training feedforward neural networks using genetic algorithms. *Proc. 11th Int. Joint Conf. Artificial Intell.*, San Mateo, CA, pages 762-767, 1989.

146. V.A. Morozov. On the solution of functional equations by the method of regularization, *Soviet Math. Dokl.,* 7:414-417, 1966.

**-N-**

147. E.A. Nadaraya. Nonparametric Estimation of Probability Densities and Regression Curves. Kluwer Academic Publishers, Dordrecht, 1989.

148. A. Neumaier. Residual Inverse Iteration for the Nonlinear Eigenvalue Problem. *SIAM Journal on Numerical Analysis*, 22(5):914-923, October 1985. http://links.jstor.org/sici?sici=0036-1429%28198510%2922%3A5%3C914%3ARIIFTN%3E2.0.CO%3B2-X

149. A. Neumaier. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM Review,* 40:636-666, 1998.

150. Hans Nyquist. Restricted Estimation of Generalized Linear Models. *Applied Statistics*, 40(1): 133-141, 1991. http://links.jstor.org/sici?sici=0035-9254%281991%2940%3A1%3C133%3AREOGLM%3E2.0.CO%3B2-2

**-O-**

151. R. L. Obenchain. Good and Optimal Ridge Estimators. *Annals of Statistics*, 6(5): 1111-1121, September 1978.
http://links.jstor.org/sici?sici=0090-5364%28197809%296%3A5%3C1111%3AGAORE%3E2.0.CO%3B2-Z

152. Robert L. Obenchain. A Critique of Some Ridge Regression Methods: Comment. *Journal of the American Statistical Association (in Theory and Methods),* 75(369):95-96, March 1980.
http://links.jstor.org/sici?sici=0162-1459%28198003%2975%3A369%3C95%3AACOSRR%3E2.0.CO%3B2-8

153. Robert L. Obenchain. *Shrinkage Regression: ridge, BLUP, Bayes and Stein.* Preliminary draft, 200+ pages, 1997.
http://members.iquest.net/~softrx/rxadobe.htm

154. D. P. O'Leary. *Near-Optimal Parameters for Tikhonov and Other Regularization Methods.* Computer Science Department Report CS-TR-4004 Institute for Advanced Computer Studies Report UMIACS-TR-99-17, University of Maryland, March 1999.
http://citeseer.nj.nec.com/98548.html

155. M.J.L. Orr. Local smoothing of radial basis function networks. *International Symposium on Artificial Neural Networks*, Hsinchu, Taiwan, 1995.
http://citeseer.nj.nec.com/orr95local.html

156. M.J.L. Orr. Regularisation in the Selection of Radial Basis Function Centers. *Neural Computation*, 7(3):606-623, 1995.
http://citeseer.nj.nec.com/orr95regularisation.html

157. M.J.L. Orr. *Introduction to radial basis function networks Technical Report*, Center for Cognitive Science, University of Edinburgh, 1996.
http://www.anc.ed.ac.uk/~mjo/papers/intro.ps.gz

158. M. J. Orr. An EM Algorithm for Regularised RBF Networks. *International Conference on Neural Networks and Brain*, Beijing, China, 1998.
http://www.anc.ed.ac.uk/~mjo/papers/icnnb98.ps.gz

159. M. J. Orr. *Recent advances in Radial Basis Function networks*. Technical report, 1999.
http://www.anc.ed.ac.uk/#mjo/papers/recad.ps.gz

160. M. J. Orr, J. Hallman, K. Takezawa, A. Murray, S. Ninomiya, M. Oide, and T. Leonard. Combining regression trees and radial basis functions. Division of informatics, Edinburgh University, 1999. *Submitted to IJNS*.
http://citeseer.nj.nec.com/orr99combining.html

161. M. J. Orr, J. Hallam, A. Murray, and T. Leonard. Assessing rbf networks using delve. *IJNS*, 2000.

**-P-**

162. I.C. Parmee. Evolutionary and Adaptive Computing in Engineering Design: The Integration of Adaptive Search Exploration and Optimization with Engineering Design Pro. Springer-Verlag New York, Inc., 2000.

163. K.E. Parsopoulos, E.C. Laskari and M.N. Vrahatis. Solving L1 Norm Errors-In-Variables Problems Using Particle Swarm Optimizer. M.H. Hamza (ed.), *Artificial Intelligence and Applications*, IASTED/ACTA Press, Anaheim, CA, USA, ISBN:0-88986-301-6, ISSN:1482-7913, pages 185-190, 2001.

164. K.E. Parsopoulos and M.N. Vrahatis. Particle Swarm Optimizer in Noisy and Continuously Changing Environments. *Artificial Intelligence and Soft Computing*, M.H. Hamza (ed.), IASTED/ACTA Press, Anaheim, CA, USA, ISBN:0-88986-283-4, ISSN:1482-7913, pages 289-294, 2001.

165. K.E. Parsopoulos, V.P. Plagianakos, G.D. Magoulas and M.N. Vrahatis. Stretching Technique for Obtaining Global Minimizers Through Particle Swarm Optimization. *Proceedings of the Particle Swarm Optimization Workshop*, Indianapolis, IN, USA, pages 22-29, 2001.

166. K.E. Parsopoulos, V.P. Plagianakos, G.D. Magoulas and M.N. Vrahatis. Objective Function "Stretching" to Alleviate Convergence to Local Minima. *Nonlinear Analysis, Theory, Methods & Applications*, 47(5):3419-3424, Pergamon, 2001.

167. K.E. Parsopoulos and M.N. Vrahatis. Particle Swarm Optimization Method for Constrained Optimization Problems. *Intelligent Technologies - Theory and Applications: New Trends in Intelligent Technologies*, P. Sincak, J. Vascak, V. Kvasnicka, J. Pospichal (eds.), IOS Press (Frontiers in Artificial Intelligence and Applications series, Vol. 76), ISBN: 1-58603-256-9, pages 214-220, 2001.

168. K.E. Parsopoulos, and M.N. Vrahatis. Particle Swarm Optimization Method in Multiobjective Problems. *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002)*, pages 603-607, ISBN:1-58113-445-2, 2002.

169. Tomaso Poggio and Federico Girosi. A Theory of Networks for Approximation and Learning. *AI Memo 1140/CBIP Paper 31*, Massachusetts Institute of Technology AI Lab, Cambridge, MA, July 1989.
ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-1140.pdf

170. K. Price and R. Storn. Differential Evolution: Numerical Optimization Made Easy. *Dr. Dobb's Journal*, pages 18-24, April 1997.

171. K. Price. An introduction to differential evolution. *New Ideas in Optimization*, McGraw-Hill, London, pages 79–108, 1999.

172. K. Price and R. Storn. Differential Evolution Homepage. April, 2001.
http://www.ICSI.Berkeley.edu/~storn/code.html

173. J. Principe, D. Xu and J. Fisher. *Information Theoretic Learning, in Unsupervised Adaptive Filtering.* Simon Haykin Editor, pages 265-319, Wiley, 1999.

174. Jose C. Principe, Dongxin Xu, Qun Zhao, John Fisher. Learning from examples with information theoretic criteria. *Journal of VLSI Signal Processing-Systems*, 26(1-2):61-77, August 2000.

**-Q-**

175. R. Quinlan. Combining instance-based and model-based learning. *Proceedings on the Tenth International Conference of Machine Learning*, University of Massachusetts, Amherst, Morgan Kaufmann, pages 236-243, 1993.

**-R-**

176. C. Radhakrishna Rao, Estimation of Parameters in a Linear Model. *Annals of Statistics (The 1975 Wald Memorial Lectures),* 4(6):1023-1037, November 1976.

177. C. Rasmussen. Evaluation of Gaussian processes and other methods for non-linear regression, *PhD thesis*, University of Toronto, pages 121-7, 1996.

178. I. Rechenberg. Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart, 1973.

179. K. Riley and C.R. Vogel. Two-level preconditioners for ill-conditioned linear systems with semidefinite regularization. S*ubmitted to Journal of Computational and Applied Mathematics*, 1999. http://www.mcs.sdsmt.edu/%7Ekriley/papers/preprint.ps.gz

180. B. W. Rust, *Truncating the Singular Value Decomposition for Ill-Posed Problems*, Tech. Report NISTIR 6131, Mathematical and Computational Sciences Division, National Institute of Standards and Technology, Gaithersburg, MD, 1998. ftp://gams.nist.gov/pub/rust/TruncSVD/MS-TruncSVD.ps

181. R.A. Rutenbar. Simulated annealing algorithms: An overview. *IEEE Circuits & Devices Magazine*, 100:19-26, 1989.

**-S-**

182. Safety Evaluation Report: Safety Evaluation by the Office of Nuclear Reactor Regulation Application of On-Line Performance Monitoring to Extend Calibration Intervals of Instrument Channel Calibrations Required by the Technical Specifications. EPRI Topical Report #104965, June 22 2000.

183. Michael G. Schimek (editor). Smoothing and Regression: Approaches, Computation, and Application. Wiley-Interscience, 2000.

184. Gregor P.J. Schmitz. *Combinatorial Evolution of Feedforward Neural Network Models for Chemical Processes*. Ph.D. Thesis, University of Stellenbosch, June 1999. http://www.lut.fi/~jlampine/schmitz.zip

185. Gregor P.J. Schmitz and Chris Aldrich. Combinatorial evolution of regression nodes in feedforward neural networks. *Neural Networks* 12(1):175-189, 1999.

186. B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1999.

187. D. Schuurmans and F. Southey. An adaptive regularization criterion for supervised learning. *Proceedings of International Conference on Machine Learning (ICML-2000)*, pages 847-854, 2000.

188. H.P. Schwefel, *Numerical Optimization of Computer Models,* John Wiley & Sons, U.K, 1981.

189. Burkhardt Seifert and Theo Gasser. Finite-Sample Variance of Local Polynomials: Analysis and Solutions. *Journal of the American Statistical Association (in Theory and Methods),* 91(433): 267-275, March 1996. http://links.jstor.org/sici?sici=0162-1459%28199603%2991%3A433%3C267%3AFVOLPA%3E2.0.CO%3B2-5

190. Self-Organizing Migrating Algorithm (SOMA) Web site. http://www.ft.utb.cz/people/zelinka/soma/

191. A.J. Smola and B. Schölkopf. *A tutorial on support vector regression*. NeuroCOLT2 Technical Report NC2-TR-1998-030, 1998.

192. R. Storn. Differential Evolution Design of an IIR-Filter with Requirements for Magnitude and Group Delay. *IEEE International Conference on Evolutionary Computation ICEC 96*, Technical Report TR-95-026, ICSI, pages 268 - 273, May 1995.

193. R. Storn and K. Price. *Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces*. Technical Report TR-95-012, ICSI, March, 1995. http://www.icsi.berkeley.edu/techreports/1995.abstracts/tr-95-012.html

194. R. Storn and K. Price. Minimizing the real functions of the ICEC'96 contest by Differential Evolution. *IEEE Conference on Evolutionary Computation*, Nagoya, pages 842 – 844, 1996. http://http.icsi.berkeley.edu/~storn/contest4.ps.gz

195. R. Storn. *System Design by Constraint Adaptation and Differential Evolution*. Technical Report TR-96-039, ICSI, November 1996. http://http.icsi.berkeley.edu/~storn/tr-96-039.ps.gz

196. R. Storn. Echo Cancellation Techniques for Multimedia Applications - A Survey. Technical Report TR-96-046, ICSI, November 1996.

197. R. Storn. On the Usage of Differential Evolution for Function Optimization. *NAFIPS 1996*, Berkeley, pages 519 – 523, 1996. http://www.icsi.berkeley.edu/~storn/bisc1.ps.gz

198. R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, Kluwer Academic Publishers, 11(4):341–359, 1997.

199. R. Storn. System Design by Constraint Adaptation and Differential Evolution. *IEEE Transactions on Evolutionary Computation*, 3(1):22 – 34, 1999.

200. William E. Strawderman. Minimax Adaptive Generalized Ridge Regression Estimators. *Journal of the American Statistical Association (in Theory and Methods)*, 73(363):623-627, September 1978. http://links.jstor.org/sici?sici=0162-1459%28197809%2973%3A363%3C623%3AMAGRRE%3E2.0.CO%3B2-N

201. M. Sugiyama and H. Ogawa. A new information criterion for the selection of subspace models. *Proceedings of the 8th European Symposium on Artificial Neural Networks (ESANN2000)*, Bruges, Belgium, pages 69-74, April 26-28 2000.

202. M. Sugiyama and H. Ogawa. Another look at $C_P$ and network information criterion as approximation of subspace information criterion. *Technical Report TR00-0012*, Department of Computer Science, Tokyo Institute of Technology, Japan, 2000.

203. M. Sugiyama and H. Ogawa. Subspace information criterion --- Unbiased generalization error estimator for linear regression. *NIPS 2000 Workshop on Cross-Validation, Bootstrap and Model Selection*, Breckenridge, USA, December 1 2000.

204. M. Sugiyama and H. Ogawa. Subspace information criterion for model selection. *Neural Computation,* 13(8):1863–1889, 2001.

205. M. Sugiyama and K.-R. Müller. Subspace information criterion for infinite dimensional hypothesis spaces. *Technical Report*, TR01-0016, Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan, 2001.

206. M. Sugiyama, D. Imaizumi and H. Ogawa. Subspace information criterion for image restoration --- Optimizing parameters in linear filters. *IEICE Transactions on Information and Systems*, E84-D(9):1249-1256, September, 2001.

207. M. Sugiyama and H. Ogawa. Model selection with small samples. *Artificial Neural Nets and Genetic Algorithms, Proceedings of the International Conference in Prague, Czech Republic, 2001*, V. Kurkova, N. C. Steele, R. Neruda, and M. Karny (eds.), Springer-Verlag, pages 418-421, 2001. Presented at *5th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA 2001)*, Prague, Czech Republic, April 22-25 2001.

208. M. Sugiyama and H. Ogawa. Optimal design of regularization parameter in linear regression. *Presented at Highdimensional Nonlinear Statistical Modeling,* Wulkow, Germany, September 15-19, 2001.

209. M. Sugiyama and H. Ogawa. Theoretical and experimental evaluation of subspace information criterion. *Special Issue on New Methods for Model Selection and Model Combination Machine Learning,* 48(1–3):25–50, 2002.
http://ogawa-www.cs.titech.ac.jp/~sugi/publications/2002/sic-eval.pdf

210. Masashi Sugiyama and Hidemitsu Ogawa. Optimal design of regularization term and regularization parameter by subspace information criterion. *Neural Networks*, 15(3): 349-361, April 2002.
http://www.sciencedirect.com/science/article/B6T08-454FD3S-1/1/d07ec50468fd3d4ae1d194cf264ff9bc

211. M. Sugiyama and K.-R. Müller. Selecting ridge parameters in infinite dimensional hypothesis spaces. *Proceedings of International Conference on Artificial Neural Networks (ICANN2002)*, Madrid, Spain, August 27-30, 2002.

212. M. Sugiyama and H. Ogawa. Release from active learning/model selection dilemma: Optimizing sample points and models at the same time. *Proceedings of International Joint Conference on Neural Networks (IJCNN2002)*, 3:2917-2922, Honolulu, Hawaii, USA, May 12-17, 2002.

213. M. Sugiyama and K.-R. Müller. Ridge parameter determination in infinite dimensional hypothesis spaces. *IEICE Technical Report, NC2001-135*, pages 21-28, 2002.

214. Rolf Sundberg. Continuum Regression and Regression, *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(3): 653-659, 1993. http://links.jstor.org/sici?sici=0035-9246%281993%2955%3A3%3C653%3ACRARR%3E2.0.CO%3B2-3

**-T-**

215. P.F. Thall, K.T. Russell and R.M. Simon. Variable selection in regression via repeated data splitting. *Journal of Computational and Graphical Statistics*, 6:416-434, 1997.

216. Robert Tibshirani. Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267-288, 1996. http://links.jstor.org/sici?sici=0035-9246%281996%2958%3A1%3C267%3ARSASVT%3E2.0.CO%3B2-G

217. A.N. Tikhonov. Solution of incorrectly formulated problems and regularization method. *Soviet math. Dokl*. 4:1053-1038, 1963.

218. A.N. Tikhonov and V.Y. Arsenin. *Solution of Ill-posed Problems*, Washington DC: W. H. Winston, 1977.

219. M.E. Tipping. The Relevance Vector Machine**.** *Advances in Neural Information Processing Systems 12*, S.A. Solla, T.K. Leen and K.-R. Müller (Eds.), MIT Press, pages 652–658, 2000.

220. Philip H.S. Torr. Model Selection for Two View Geometry: A Review. *Shape, Contour and Grouping in Computer Vision*, pages 277-301, 1999. http://citeseer.nj.nec.com/303971.html

221. L.H. Tsoukalas and R.E. Uhrig. *Fuzzy and Neural Approaches in Engineering.* Wiley, John & Sons, Inc., 1996.

222. Josef Tvrdík and Ivan Krivý. Evolutionary Heuristics in Nonlinear Regression. *Proceedings of MENDEL'99, 5th International Mendel Conference on Soft Computing*, Ošmera, Pavel (ed.), June 9–12, 1999, Brno, Czech Republic. Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automation and Computer Science, Brno, Czech Republic, ISBN 80-214-1131-7, pages 59–64, 1999.

223. Josef Tvrdík and Ivan Krivý. Simple Evolutionary Heuristics for Global Optimization. *Computational Statistics & Data Analysis*, 30(3):345–352, May 28, 1999.

**-U-**

224. A.M. Urmanov, H. Bozdogan, A.V. Gribok, J.W. Hines, and R.E. Uhrig. Information Complexity-Based Regularization Parameter Selection for Solution of Ill-Conditioned Inverse Problems. *Inverse Problems*, 18(3), April 2002.

225. A.M. Urmanov. An Information Approach to Regularization Parameter Selection for the Solution of Ill-Posed Inverse Problems Under Model Misspecification. PhD Dissertation. Dept. of Nuclear Eng.

University of Tennessee, Knoxville, August 2002.
http://etd.utk.edu/2002/UrmanovAleksey.pdf

226. A.M. Urmanov granted permission to include the MATLAB code of the objective functions used with DEALRR. *Personal correspondance*. Jan. 9, 2003.

**-V-**

227. S. Valle, W. Li and S.J. Qin. Selection of the Number of Principal Components: The Variance of the Reconstruction Error Criterion with a Comparison to Other Methods. *Ind. Eng. Chem. Res.*, 38:4389-4401, 1999.

228. M.H. van Emden, An Analysis of Complexity. *Mathematical Centre Tracts*, 35, Amsterdam, 1971.

229. M. Vannucci. Matlab Code for Bayesian Variable Selection. *The ISBA Bulletin*, 7(3), September 2000, Texas A&M University, USA, 2000.
http://stat.tamu.edu/~mvannucci/webpages/matlab/isbanews.pdf

230. V.N. Vapnik. An overview of statistical learning theory, *IEEE Transactions on Neural Networks*, 10(5): 988- 999, 1999.

231. A. Vehtari, and J. Lampinen. Bayesian input variable selection using cross-validation predictive densities and reversible jump MCMC. *Technical Report B28*, Laboratory of Computational Engineering, Helsinki University of Technology, 2001.
http://www.lce.hut.fi/publications/pdf/VehtariLampinen_B28.pdf.bak

232. Miguel Villalobos, Grace Wahba. Inequality-Constrained Multivariate Smoothing Splines with Application to the Estimation of Posterior Probabilities. *Journal of the American Statistical Association (in Theory and Methods)*, 82(397):239-248, March 1987.
http://links.jstor.org/sici?sici=0162-1459%28198703%2982%3A397%3C239%3AIMSSWA%3E2.0.CO%3B2-K

233. C.R. Vogel. Non-convergence of the l-curve regularization parameter selection method, *Inverse Problems*, 12:535-547, 1996.

**-W-**

234. Gerd Wagner. *Foundations of Knowledge Systems with Applications to Databases and Agents. University of Leipzig*. Germany The Kluwer International Series on Advances in Database Systems. Volume 13, July 1998.
http://www.inf.fu-berlin.de/~wagnerg/FoundKS.pdf

235. Grace Wahba. Data-Based Optimal Smoothing of Orthogonal Series Density Estimates. *Annals of Statistics*, 9(1):146-156, January 1981.
http://links.jstor.org/sici?sici=0090-5364%28198101%299%3A1%3C146%3ADOSOOS%3E2.0.CO%3B2-J

236. Grace Wahba. A Comparison of GCV and GML for Choosing the Smoothing Parameter in the Generalized Spline Smoothing Problem. *Annals of Statistics*, 13(4):1378-1402, December 1985.
http://links.jstor.org/sici?sici=0090-5364%28198512%2913%3A4%3C1378%3AACOGAG%3E2.0.CO%3B2-M

237. Grace Wahba. Spline Models for Observational Data. *SIAM*, 1990.

238. Alan T. K. Wan. On generalized ridge regression estimators under collinearity and balanced loss. *Applied Mathematics and Computation*, 129(2-3):455-467, July 10, 2002. http://www.sciencedirect.com/science/article/B6TY8-45TSJ30-J/1/8cd206fc65fcd678691a8320685c2159

239. M.P. Wand and M.C. Jones. Kernel Smoothing. Monographs on Statistics and Applied Probability 60.Chapman and Hall, 1995.

240. Feng-Sheng Wang and Jyh-Woei Sheu. Multiobjective parameter estimation problems of fermentation processes using a high ethanol tolerance yeast. *Chemical Engineering Science*, 55(18):3685-3695, September 15. 2000. http://www.sciencedirect.com/science/article/B6TFK-40GJ2MB-9/1/10b780c123662124bb655897a1a62a26

241. L. X. Wang, J. M. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks*, 3:807-814, 1992.

242. S.N. Wood. Minimising model fitting objectives that contain spurious local minima by bootstrap restarting. *Biometrics*, 57:240-224, March 2001.

**-X-**

**-Y-**

243. L. Yao and W. A. Sethares. Nonlinear parameter estimation via the genetic algorithm. *IEEE Trans. Signal Processing*, 42:927-935, 1994.

**-Z-**

244. L.A. Zadeh. Foreword. Applied Soft Computing Applied Soft Computing, 2:1–2, 2001.

245. N. Zavaljeski and K.C. Gross. Uncertainty Analysis for Multivariate State Estimation in Safety-Critical and Mission-Critical Maintenance Applications. *Proceedings of the Maintenance and Reliability Conference (MARCON 2000)*, Knoxville, TN, May 7-10, 2000.

246. Scott L. Zeger and M. Rezaul Karim. Generalized Linear Models With Random Effects; A Gibbs Sampling Approach. *Journal of the American Statistical Association*, 86(413):79-86. March 1991.

247. I. Zelinka, J. Lampinen, and L. Nolle. SOMA - Self-Organizing Migrating Algorithm. *Proceedings of the 13th International Conference on Process Control '01*, Strbske Pleso, High Tatras, SK, pages 100-105, June 11-14, 2001.

248. Hongyuan Zha and Per Christian Hansen. Regularization and the General Gauss-Markov Linear Model, *Mathematics of Computation*, 55(192):613-624, October 1990.

249. E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257-271,

November 1999.
ftp://ftp.tik.ee.ethz.ch/pub/people/zitzler/ZT1999.ps

# Appendix

## A.1 MATLAB Code for ICOMPRPS Objective Function

```
function [ff,b_reg,c]=ga_ff_icomprps(y,U,s,V,d)
% GA_FF_ICOMRPS  A FITNESS FUNCTION FOR GA OPTIMIZATION.
%             IT SCORES ICOMPRPS FOR A GIVEN VECTOR
%             OF REGULARIZATION PARAMETERS (EACH
%             ELEMENT OF THE VECTOR CORRESPONDING
%             TO A COMPONENT IN SVD DECOMPOSITION)
%             THE LINEAR REGRESSION PROBLEM Y=X*B+E
%             IS SOLVED BY GENERALIZED RIDGE THAT
%             CAN DUMP PARTICULAR COMPONENTS.
%
% USAGE:
%         [FF]=GA_FF_ICOMPRPS(Y,X,D);
%         OR
%         [FF,B_REG,C]=GA_FF_ICOMPRPS(Y,U,S,V,D)
% WHERE
%         FF    - ICOMPRPS VALUE
%         B_REG - REGULARIZED REGRESSION COEFFICIENTS
%         C     - COEFFICIENTS THAT SHOW HOW MUCH EACH
%                 COMPONENT CONTRIBUTES TO THE SOLUTION
%                 B_REG = C(1)*V1 + C(2)*V2 + ... + C(M)*VM
%         Y     - AN (Nx1) RESPONSE VECTOR
%         X     - AN (NxM) MATRIX OF PREDICTORS
%         U,S,V - SVD DECOMPOSITION OF X ST X=U*DIAG(S)*V'.
%         D     - AN (Mx1) VECTOR OF REGULARIZATION PARAMETERS
%                 D(i)>=0.
%
% WRITTEN BY ALEKSEY URMANOV
% APR. 6, 2002 E-MAIL: URMANOV@UTK.EDU

if nargin<5, d=s;[U,s,V]=csvd(U);end
[n,m]=size(U);
if size(d,1)<m, d=d';end

% OLS noise estimate:
b_ols=V*diag(1./s)*U'*y;
r_ols=(y-(U*diag(s)*V')*b_ols);
s2=r_ols'*r_ols/n;
```

```
b_reg=V*diag(s./(s.^2+d.^2))*U'*y;
r_reg=(y-(U*diag(s)*V')*b_reg);
rho=r_reg'*r_reg;
tr=trace(U*diag((s.^2)./(s.^2+d.^2))*U');
v=1./(s.^2+d.^2);
c1 = length(v)/2*(log(mean(v))-mean(log(v)));

ff=rho/n + 2*s2/n*(tr+c1);
c=diag(s./(s.^2+d.^2))*(U'*y);

return
```

## A.2 MATLAB Code for Mallows' CL Objective Function

```
function [cl,b_reg,c]=ga_ff_cl(y,U,s,V,d)
% GA_FF_CL  A FITNESS FUNCTION FOR GA OPTIMIZATION.
%         IT SCORES CL VALUES FOR A GIVEN VECTOR
%         OF REGULARIZATION PARAMETERS (EACH
%         ELEMENT OF THE VECTOR CORRESPONDING
%         TO A COMPONENT IN SVD DECOMPOSITION)
%         THE LINEAR REGRESSION PROBLEM Y=X*B+E
%         IS SOLVED BY LOCALIZED RIDGE THAT
%         CAN DUMP PARTICULAR COMPONENTS.
%
% USAGE:
%         [CL]=GA_FF_CL(Y,X,D);
%         OR
%         [CL,B_REG,C]=GA_FF_CL(Y,U,S,V,D)
% WHERE
%         CL    - MALLOW'S CL PREDICTIVE ERROR ESTIMATOR)
%         B_REG - REGULARIZED REGRESSION COEFFICIENTS
%         C     - COEFFICIENTS THAT SHOW HOW MUCH EACH
%                 COMPONENT CONTRIBUTES TO THE SOLUTION
%                 B_REG = C(1)*V1 + C(2)*V2 + ... + C(M)*VM
%         Y     - AN (Nx1) RESPONSE VECTOR
%         X     - AN (NxM) MATRIX OF PREDICTORS
%         U,S,V - SVD DECOMPOSITION OF X ST X=U*DIAG(S)*V'.
%         D     - AN (Mx1) VECTOR OF REGULARIZATION PARAMETERS.
%
% WRITTEN BY ALEKSEY URMANOV
% NOV. 16, 2001 E-MAIL: URMANOV@UTK.EDU

if nargin<5, d=s;[U,s,V]=csvd(U);end
[n,m]=size(U);
if size(d,1)<m, d=d';end

b_ols=V*diag(1./s)*U'*y;
r_ols=(y-(U*diag(s)*V')*b_ols);
s2=r_ols'*r_ols/n;

b_reg=V*diag(s./(s.^2+d.^2))*U'*y;
```

```
r_reg=(y-(U*diag(s)*V')*b_reg);
rho=r_reg'*r_reg;
%tr=trace(U*diag((s.^2)./(s.^2+d.^2))*U');
tr=sum((s.^2)./(s.^2+d.^2));

cl=rho/n + 2*s2/n*tr;
c=diag(s./(s.^2+d.^2))*(U'*y);

return
```

## A.3 MATLAB Code for GCV Objective Function

```
function [gcv,b_reg,c]=ga_ff_gcv(y,U,s,V,d)
% GA_FF_GCV  A FITNESS FUNCTION FOR GA OPTIMIZATION.
%          IT SCORES GCV VALUES FOR A GIVEN VECTOR
%          OF REGULARIZATION PARAMETERS (EACH
%          ELEMENT OF THE VECTOR CORRESPONDING
%          TO A COMPONENT IN SVD DECOMPOSITION)
%          THE LINEAR REGRESSION PROBLEM Y=X*B+E
%          IS SOLVED BY LOCALIZED RIDGE THAT
%          CAN DUMP PARTICULAR COMPONENTS.
%
% USAGE:
%          [GCV]=GA_FF_GCV(Y,X,D);
%          OR
%          [GCV,B_REG,C]=GA_FF_GCV(Y,U,S,V,D)
% WHERE
%          GCV   - GCV
%          B_REG - REGULARIZED REGRESSION COEFFICIENTS
%          C     - COEFFICIENTS THAT SHOW HOW MUCH EACH
%                  COMPONENT CONTRIBUTES TO THE SOLUTION
%                  B_REG = C(1)*V1 + C(2)*V2 + ... + C(M)*VM
%          Y     - AN (Nx1) RESPONSE VECTOR
%          X     - AN (NxM) MATRIX OF PREDICTORS
%          U,S,V - SVD DECOMPOSITION OF X ST X=U*DIAG(S)*V'.
%          D     - AN (Mx1) VECTOR OF REGULARIZATION PARAMETERS.
%
% WRITTEN BY ALEKSEY URMANOV
% APR. 6, 2002 E-MAIL: URMANOV@UTK.EDU

if nargin<5, d=s;[U,s,V]=csvd(U);end
[n,m]=size(U);
if size(d,1)<m, d=d';end

b_reg=V*diag(s./(s.^2+d.^2))*U'*y;
r_reg=(y-(U*diag(s)*V')*b_reg);
rho=r_reg'*r_reg;
tr=sum((s.^2)./(s.^2+d.^2));
```

```
gcv=(rho/n)/(1-tr/n)^2;
c=diag(s./(s.^2+d.^2))*(U'*y);

return
```

# Vita

Mark Alan Buckner was born in Kingsport, Tennessee on June 22, 1963. He attended elementary and secondary schools in Kingsport, Tennessee, where he graduated from Dobyns-Bennet High School in 1981. He entered Carson-Newman College in Jefferson City, Tennessee in the fall of 1981, majoring in Physics, and Psychology with Minors/Concentrations in Math, Computer Science and Health Physics. He obtained his Bachelor of Arts degree in the spring 1986.

He has been an employee at Oak Ridge National Laboratory since March, 1987 and he is presently a member of the RF and Microwave Systems Group in the Engineering Science and Technology Division.

He began attending the University of Tennessee in 1988, working towards a Master of Science degree in Nuclear Engineering while continuing to work full-time at ORNL. He obtained his Master of Science degree in Nuclear Engineering at the University of Tennessee in 1991.

He re-entered the University of Tennessee and began working toward a Doctor of Philosophy degree in the fall of 1996. He received his Ph.D. in Nuclear Engineering in the spring of 2003.

While at ORNL he has served as the Principle Investigator and Program Manager (PI/PM) for a Cooperative Research and Development Agreement between Lockheed Martin Energy Research Corporation and a major oil service provider to develop a High-Temperature Smart Transducer Interface Node and Telemetry System. He has also been PI on the NASA SFINX project, developing a Scalable Fault-tolerant Intelligent Network for Xducers. He has served as PI/PM of the Advanced RADiation Sensors (ARADS) development group which provides R&D support for the Navy RADIAC instrument program (1995-1998). He designed and developed the first prototype of a plastic scintillator fiber optic based sensor used in the Continuous Automated Vault Inventory Monitoring System (CAVIS) being field tested at Y-12. He developed the Gunite Tank Isotope Mapping Probe (GIMP), a CsI(Tl)+Si diode based gamma camera, being used to perform isotopic imaging of the interior walls of the Gunite waste tanks at X-10. He developed a non-destructive assay system for real-time monitoring of the reactor gas removal system at the Molten Salt Reactor Experiment at ORNL using CdZnTe detectors and an adaptive neural-fuzzy inference engine. He received an R&D-100 award as co-developer of the Harshaw Model 8800 Thermoluminescent Dosimetry System. He developed the passive Combination Area Neutron Spectrometer, utilizing a combination of TLD and bubble detector technologies under DOE EH-1 FWP EHHA050.

He has authored or co-authored over 25 publications in the open literature. He is an active Member of the IEEE, and has served as Secretary of IEEE IMS TC-9, and the vice-chair of the IEEE P1451.3 working group

developing "A Smart Transducer Interface for Sensors and Actuators". He is presently serving on International Committee for Information Technology Standards (INCITS) T20 Real Time Locating System Working group. He has served as a member of the Virtual Socket Interface Alliance, System Level Design working group, was a Lead assessor for the Department of Energy Laboratory Accreditation Program (1993-1996), and served on the Program Committee for the 1999 IRIS Sensors and Data Fusion Specialty Group Meeting.

Mark has 15 years of combined experience in electronic hardware and software systems design, Heath Physics and radiation instrumentation, including novel radiation sensors, smart sensor networks, radios platforms for mobile ad hoc networking and passive RFID applications for tagging and tracking of hazardous materials.

Mark, his wife, Lisa and their three children, Sydney, Luke and Madeline, currently live in Oak Ridge, Tennessee. They are active in the community, volunteering in the local public school their daughter attends. And, they are active members of Covenant Presbyterian Church, PCA, where Mark serves as an elder, and Sunday school teacher.