RESPONSE SURFACE OPTIMIZATION FOR HIGH DIMENSIONAL SYSTEMS WITH MULTIPLE RESPONSES

Direttore della Scuola: Ch.ma Prof.ssa ALESSANDRA SALVAN
Supervisore: Ch.ma Prof.ssa IRENE POLI
Co-supervisori: Ch.ma Prof.ssa KATE SMITH-MILES
                  Ch.mo Prof. ROB J HYNDMAN

Dottoranda: LAURA VILLANOVA

31 Luglio 2010
Contents

Abstract iii

Riassunto vii

Acknowledgement xi

1 Introduction 1

1.1 Overview ......................................................... 1

1.2 Main contributions of the thesis ................................. 4

2 The motivating problem 9

2.1 Slide functionalization for biomedical applications .......... 10

2.2 The multiresponse high-dimensional chemical system: design of experiments and optimization ..................... 16

3 Nonparametric and semiparametric response surface methodology 21

3.1 Experimental designs ........................................... 26

3.2 Response surface modeling .................................... 30

3.3 Optimization methods .......................................... 48

3.4 Towards a customized approach ................................ 57

4 The evolutionary model-based multiresponse experimental design 59

4.1 The Evolutionary Model-based Multiresponse Approach (EMMA) .......... 60

4.2 EMMA: the efficacy in the optimization process ............ 68

4.3 EMMA: the success in the chemical problem ............... 75

4.4 EMMA: towards the multiresponse model selection ........ 79
5 Multiresponse regression

5.1 The multivariate regression problem ................. 84
5.2 Separate (uniresponse) multiple regression ......... 86
5.3 Multivariate multiple regression .................... 99
5.4 Towards problem-based model selection ............ 120

6 A meta-learning framework for the model selection 

6.1 The problem space .................................. 125
6.2 The algorithm and performance spaces ............ 151
6.3 The feature space .................................. 152
6.4 Concluding remarks: ongoing and future research . 161

Conclusions ............................................. 163
Abstract

This thesis is about the optimization of physical systems (or processes) characterized by a high number of input variables (e.g., operations, machines, methods, people, and materials) and multiple responses (output characteristics). These systems are of interest because they are common scenarios in real-world studies and they present many challenges for practitioners in a wide range of applicative fields (e.g., science, engineering). The first objective of the study was to develop a model-based approach to support the practitioners in planning the experiments and optimizing the system responses. Of interest was the creation of a methodology capable of providing a feedback to the practitioner while taking into account his/her point of view. The second objective was to identify a procedure to select the most promising model, to be combined with the model-based approach, on the basis of the features of the applicative problem of interest.

To cope with the first objective, experimental design, modeling and optimization techniques have been combined in a sequential procedure that interacts with the practitioner at each stage. The developed approach has roots in nonparametric and semiparametric response surface methodology (NPRSM), design and analysis of computer experiments (DACE), multi-objective optimization and swarm intelligence computation. It consists of augmenting an initial experimental design (set of experiments) by sequentially identifying additional design points (experiments) with expected improved performance. The identification of new experimental points is guided by a particle swarm optimization (PSO) algorithm that minimizes a distance-based function. In particular, the distance between the measured response values and a target is minimized. The target is composed of ideal values of the responses and is selected using a multivariate adaptive regression splines (MARS) model, which is updated as soon as new experiments are implemented.
and the corresponding response values are measured. The developed approach resulted in a sequential procedure named Evolutionary Model-based Multiresponse Approach (EMMA). When tested on a set of benchmark functions, EMMA was shown to overcome the potential problem of premature convergence to a local optimum and to correctly identify the true global optimum. Furthermore, EMMA is distribution-free and it allows the automatic selection of the target, in contrast to the trial-and-error procedures usually employed for this purpose. Finally, EMMA was applied to a real-world chemical problem devoted to the functionalization of a substrate for possible biomedical studies. With respect to the method typically employed by the scientists, improvements of the responses of up to 380% were detected. The proposed approach was thus shown to hold much promise for the optimization of multiresponse high dimensional systems. Moreover, EMMA turned out to be a valuable methodology for industrial research. Indeed, by means of a preliminary simulation study, it gave an initial estimate of the number of experiments and time necessary to achieve a specific goal, thus providing an indication of the budget required for the research.

To deal with the second objective of the research, a meta-learning approach for model selection was adopted. Interest in model selection strategies arose from questions such as ‘Is MARS the best model we could have used?’ and ‘Given an applicative problem, how can we select the most promising modeling technique to be combined with EMMA?’ Indeed, it is now generally accepted that no single model can outperform some other models over all possible regression problems. Furthermore, the model performance ‘... may depend on the detailed nature of the problem at hand in terms of the number of observations, the number of response variables, their correlation structure, signal-to-noise ratio, collinearity of the predictor variables, etc.’ (Breiman & Friedman 1997). The meta-learning approach was adopted to select the most promising model on the basis of measurable characteristics of the investigated problem. The basic idea was to study a set of multiresponse regression models and evaluate their performance on a broad class of problems, that were characterized by various degrees of complexity. By matching the problem characteristics and the models’ performance, the aim was to discover the conditions under which a model outperforms others as well as to acquire some rules to be used as a guidance when faced with a new application. The procedures to simulate the datasets were developed, the metrics to
measure the problems characteristics were identified, and the R code to evaluate the models’ performances was generated. The foundations for a large computational study was therefore established. Implementation of such study is part of ongoing research, and future works will aim to examine the obtained empirical rules from a theoretical perspective with a view to confirm their validity, as well as generating insights into each model’s behaviour.
Riassunto

La tesi riguarda l’ottimizzazione di sistemi (o processi) fisici caratterizzati da un elevato numero di variabili in ingresso (operazioni, macchine, metodi, persone, materiali) e da più variabili risposta, impiegate per misurare le proprietà del prodotto finale. Questa tipologia di sistemi è molto frequente in un ampio spettro di campi applicativi, che spaziano dalla scienza all’ingegneria, e pone lo sperimentatore di fronte a delle problematiche di non sempre facile risoluzione. Il primo obiettivo di questo studio era di sviluppare un approccio, basato su un modello statistico, che fosse in grado di supportare lo sperimentatore nella pianificazione degli esperimenti e nell’ottimizzazione delle risposte del sistema. Fondamentale era lo sviluppo di una procedura capace di tenere in considerazione il punto di vista dello sperimentatore e fornirgli continuamente un feedback. Il secondo obiettivo della ricerca era l’identificazione di un metodo volto a selezionare il miglior modello statistico, da integrare all’approccio proposto, sulla base delle caratteristiche del problema applicativo investigato.

Il primo obiettivo ha portato allo sviluppo di una procedura sequenziale che impiega tecniche di disegno sperimentale, modellazione e ottimizzazione, e che interagisce, ad ogni passo, con lo sperimentatore. La metodologia proposta è stata denominata EMMA e coinvolge varie aree di ricerca scientifica e computazionale, quali superfici di risposta nonparametriche e semiparametriche, disegno e analisi di esperimenti a computer, ottimizzazione multiobiettivo e computazione ispirata al comportamento degli sciami in natura. EMMA prevede l’identificazione di un disegno sperimentale (insieme di esperimenti) che viene successivamente integrato con dei punti sperimentali (esperimenti), identificati in modo sequenziale. Il processo di identificazione dei nuovi punti sperimentali è guidato da un algoritmo di ottimizzazione particle swarm, che minimizza la distanza fra i valori...
di risposta osservati e un target. Il target è un insieme di valori ottimali, uno per ogni risposta, che vengono selezionati usando un modello di regressione multivariata basato su spline (MARS). Tale target viene aggiornato non appena i nuovi esperimenti vengono implementati e le corrispondenti risposte vengono misurate. Quando testato su un insieme di funzioni standard, EMMA ha dimostrato di poter superare il potenziale problema di convergenza prematura verso un ottimo locale e di poter identificare correttamente il vero ottimo globale. Inoltre, EMMA non richiede nessuna assunzione sulla distribuzione dei dati e, diversamente da altre procedure, permette di selezionare automaticamente il target. Infine, EMMA è stata applicata ad un problema chimico volto alla funzionalizzazione di un substrato per possibili applicazioni biomediche. Rispetto al metodo generalmente usato dagli scienziati, EMMA ha permesso di migliorare le risposte del sistema di vari punti percentuali, e incrementi fino al 380% sono stati osservati. L’approccio proposto costituisce pertanto un metodologia con elevate potenzialità per l’ottimizzazione di sistemi multirisposta ad alta dimensionalità. Inoltre, grazie a degli studi di simulazione, EMMA permette di ottenere una stima iniziale del numero di esperimenti e del tempo necessario per raggiungere il miglioramento desiderato. Di conseguenza, potendo fornire un’indicazione del budget richiesto per lo studio di interesse, la metodologia risulta essere di interesse specialmente nel settore della ricerca industriale.

Il secondo obiettivo ha portato allo sviluppo di un approccio di meta-apprendimento per la selezione del modello. L’interesse nella selezione del modello deriva da domande quali ‘E MARS il miglior modello che avremmo potuto usare?’ e ‘Dato un problema applicativo, come possiamo selezionare la tecnica di modellazione più promettente da combinare con EMMA?’. Infatti, è ormai riconosciuto che non esiste un modello le cui performance sono migliori, rispetto ad altre tecniche di modellazione, per tutti i possibili problemi di regressione. Inoltre, le performance di un modello ‘... possono dipendere dalla natura del problema investigato in termini di numero di osservazioni, numero di variabili risposta, struttura di correlazione delle variabili, rapporto segnale-rumore, grado di collinearity dei predittori, etc.’ (Breiman & Friedman 1997). L’approccio di meta-apprendimento è stato adottato per identificare il modello statistico più promettente, sulla base delle caratteristiche del problema investigato. L’idea consisteva nello studiare un insieme di modelli di regressione multirisposta e valutare la loro performance su un’ampia classe di problemi caratterizzati
da diversi gradi di complessità. Studiando la relazione fra le caratteristiche del problema e la performance dei modelli, lo scopo è di scoprire sotto quali condizioni un modello è migliore di altri e simultaneamente acquisire alcune regole da poter usare come linee guida nello studio di nuove applicazioni. A tale scopo sono state sviluppate le procedure per simulare i dati, le metriche per misurare le caratteristiche dei problemi, e il codice R necessario per la valutazione delle performance dei modelli. Questo ha permesso di gettare le fondamenta di un ampio studio di simulazione, la cui implementazione fa parte della ricerca attualmente in corso. Lo scopo della ricerca futura è di esaminare, da un punto di vista teorico, le regole empiriche ottenute in modo da poterne confermare la validità, oltre che favorire una migliore comprensione del comportamento delle tecniche di modellazione investigate.
Acknowledgement

This thesis is the result of a three year adventure, which allowed me to meet people that I hold in high esteem, and to grow both as a student and as a person. Prof. Irene Poli made this adventure possible. She inspired my enthusiasm about statistics and has always encouraged me since my early studies at the university. I’m grateful to her also for suggesting the research topic and for providing me with the possibility to collaborate with CIVEN and ECLT, where my research started. I’m most grateful to Prof. Kate Smith-Miles for welcoming me at Monash University, for the trust she displayed, for her professionalism, for always being available despite her commitments, and for assisting me with the research time after time. I’m also most grateful to Prof. Rob Hyndman for his great supervision, for his availability and professionalism, for welcoming me in his research group, and for his ability to transmit his enthusiasm for the research. Also, I would like to thank the PhD course in statistics, with special thanks to Prof. Alessandra Salvan for the advice and availability she continuously demonstrated, during both my studies in Padova and abroad. I would like to thank Giuliana for her friendship and sensibility, Alessandra for her help and professionalism, and all the staff of the Department of Statistical Sciences in Padova. I’m grateful also to Padova University, ECLT, CIVEN, Monash University, and University Ca’ Foscari of Venezia for enabling my studies and research.

A huge, endless thanks to my lovable husband Paolo for encouraging and supporting me every single day, for holding my hands during the most arduous moments and celebrating with me during the most triumphant. Thank you also to my parents Cesira and Gianni for their support, for accommodating my studies, and for having been able to understand my choices and needs. Thanks to my sister Barbara and her husband Nicola, for their
kindness and for their words of encouragement. Also, I would like to thank Fiorenza, Renzo, Stefano and Raffaella for their thoughtfulness and enheartening advice. Many thanks to my dear friends Lisa, Gianni, Dario and Nicholae, Ermanno, Serena and Andrea, Simone, Giorgia, Mara, Michele, Mauro, Michele, Alessandro. Thanks also to Alessandro and Michela, Paolo and Stefania, Piero and Roberta, Vanya, Paolo, Diego, Stefano and Valentina, and Simone and Anhaid. A special thanks to my Australian colleague Mark, for his friendship and kindness, and for his helpful suggestions. Thanks also to Kareem for his sensibility and to Ying-Oon, Olivia, Amanda, Gertrude, Doris, Rosemary, Linda, and Melissa.

Thank you guys!
Chapter 1

Introduction

1.1 Overview

This thesis focuses on Nonparametric and Semiparametric Response Surface Methods (NPRSMs) for the optimization of high dimensional systems. The term ‘system’ refers to any physical process such as biological and chemical systems, linear systems in control, and non-linear dynamic systems (Bates et al. 1996). According to Montgomery (2009), a system can usually be visualized as a combination of resources (e.g., operations, machines, methods, and people) that transforms some input (often a material) into an output. The characteristics of the output are measured by one or more observable response variables and are influenced by the inputs (factors or design variables). In real-world applications, systems characterized by a large number of inputs are common. These processes are usually referred to as high dimensional or large systems (Bates et al. 1996). Moreover, if the output characteristics are measured by more than one variable, then the system has multiple responses. When the main purpose is the optimization of a multiresponse high dimensional system, compromise solutions are expected because of the trade-off between the output variables. Furthermore, a global perspective is adopted to provide an overview of the whole system and to avoid the identification of local optima. As a result, wide ranges of the input variables are investigated and complex nonlinear relationships between inputs and outputs are likely to arise. Furthermore, real-world problems are often characterized by the presence of infeasible experiments,
which make it necessary to avoid the investigation of certain areas of the search space through the definition of suitable constraints on the input variables. Finally, restrictions on the quantity of the available resources (e.g., budget, materials, time) typically limit the number of experiments that can be implemented to investigate and optimize a system. Nonparametric and semiparametric response surface methodology (NPRSM) is suitable for the optimization of high dimensional systems, as its goal is to explore and present relationships between the process response (output variable) and the design variables (inputs) with a view to identifying the settings of the inputs that provide an optimum response (Myers 1999). As in the standard response surface methodology (RSM) (Box & Wilson 1951, Box & Draper 1987, Khuri & Cornell 1996, Myers & Montgomery 2002), data from an experimental design is used to build a model approximating the true functional relationship between the output and input variables, then optimization is carried out on the obtained model (Montgomery 2009). However, the standard RSM designs (e.g., the factorial, the fractional factorial, and the central composite designs) and models (first- and second-order models) are often unsuitable for the high dimensional systems (Bates et al. 1996). As a result, the response surface designs and polynomial models give way to space-filling designs, which are capable of spreading the points uniformly in the investigated area, and highly adaptive models, which are more suitable to the complex non-linear responses and the high dimensionality of the input space (Bates et al. 1996, Myers et al. 2004). Furthermore, the local optimization procedures (e.g., steepest ascent) are replaced by heuristics capable of dealing with nonlinear surfaces characterized by multiple optima.

The work in this thesis is motivated by an applicative problem related to the functionalization of a substrate for possible biomedical applications. This functionalization is achieved using a chemical process to cover the substrate. The aim of the experimentation is to select the chemical composition which results in the optimal coating quality. The chemical composition is a combination of six factor levels (possible values of the input variables), where the factors are the chemical components employed. The coated slide is the output and its quality is measured by four response variables. A fifth response is employed to measure the stability of the chemical composition which is used to obtain the
coating. Finally, practical considerations give rise to two constraints on the input variables and limitations on the number of experiments to be carried out.

The thesis is organized as follows. Chapter 2 describes the applicative problem which motivated this work. Details of the technology investigated and of the chemistry employed are provided in order to give an overview of the practical constraints and challenges of the problem. Finally, the optimization problem is formulated. Chapter 3 presents a literature review on Nonparametric and Semiparametric Response Surface Methods (NPRSMs). In the first part of the chapter, standard Response Surface Methods (RSMs) and Design and Analysis of Computer Experiments (DACE) are described briefly in order to provide a self-contained overview of the research field of interest. The purpose of the chapter is to give an account of the experimental design, modeling and optimization techniques typically employed in NPRSM. Differences between NPRSM and the standard RSM techniques are highlighted and methods developed in RSM to cope with multiple responses are introduced. This chapter also provides the theoretical background to the mixed model-based penalized splines regression. This is a methodology which is particularly suited to the NPRSM framework; however, to the best of our knowledge, it has not yet been employed in the context of NPRSM. Concluding remarks about the limitations of the existing techniques are given and the foundation for the development of a new approach is provided. Chapter 4 describes the proposed approach and presents the results achieved on a standard set of test functions and on the applicative problem which motivated this work. At the end of the chapter, a discussion of the open questions arising from the study leads to the remainder of the thesis. From now on, the focus is on regression modeling techniques which are suitable for multiresponse problems. Two classes of methods for modeling multiresponse problems are investigated. They are the multivariate multiple regression procedure and the separate (uniresponse) multiple regression procedure (Breiman & Friedman 1997). Our main interest is in the selection of the modeling approach which is most appropriate to the problem at hand. Indeed, according to the No Free Lunch (NFL) theorems introduced by Wolpert & Macready (1997), there is no universally best algorithm (or modeling technique) for tackling a broad problem domain. As was stated by Wolpert & Macready (1997), for any algorithm (or model), any improved performance over one class of problems is offset by a diminished performance over another class.
In such a context, an understanding of the problem characteristics is crucial in order to select the most appropriate technique, which will ideally take into consideration the known structural properties of the problem or instance (Smith-Miles 2008). Of particular relevance are the questions raised by Breiman & Friedman (1997), who stated that ‘An important issue is whether any of the multivariate multiple-regression procedures offer sufficient improvement over separate (uniresponse) multiple regressions to justify their consideration as viable alternatives. And, among those that do, which provide the best trade-off between accuracy improvement and increased complexity, both in terms of implementation and computations?’. The authors pointed out that ‘The answers to these questions may depend on the detailed nature of the problem at hand in terms of the number of observations, the number of response variables, their correlation structure, signal-to-noise ratio, collinearity of the predictor variables, etc.’ The background theory relating to separate and multivariate multiple-regression techniques is provided in chapter 5. Chapter 6 describes a meta-learning architecture which is adopted as a research methodological framework to generate a knowledge system for model selection purposes (Smith-Miles 2008). The meta-learning framework consists of four elements: a set of simulated problems (instances), a set of problem characteristics (features), a set of multiresponse models (methods), and one or more performance measures. The final goal is to model the relationship between method performances and problem features in order to provide some guidance in the selection of the regression technique which is most suitable to the problem at hand. The chapter provides a detailed description of various procedures, which have been developed to simulate the data, and metrics, which have been identified to capture the features of the simulated data. Finally, the conclusions of the thesis are drawn by reviewing the research process, pointing out the encountered problems, and describing the achieved results as well as the ongoing and future research.

1.2  Main contributions of the thesis

The main contribution of this thesis is the development of a model-based approach for the optimization of high dimensional systems with multiple responses. In contrast to the majority of the existing NPRSMs, the proposed approach does not rely strongly
on the fitted model. Indeed, NPRSMs assume that if the fitted model is an adequate approximation of the true response function, then the analysis will be approximately equivalent to the analysis of the actual system (Montgomery 2009). However, if the initial sample is sparse, the fitted model can give a highly misleading view of the approximated function. Because the misleading view is subsequently taken as correct, the optimization procedure may either stop prematurely or become excessively local (Jones 2001). This drawback is a result of the two-stage nature of most of the existing methodologies. Jones (2001) proposed a one-stage approach in which, instead of optimizing the fitted model, a target value or goal for the objective function is assumed. According to Jones (2001) and Gutmann (2001), this methodology can handle extremely deceptive problems. However, when the goal is unknown or unclear, several target values must be used in a trial-and-error approach, thus increasing the computational burden of the procedure. The approach developed in this thesis is a one-stage distribution-free method that combines a multiresponse Multivariate Adaptive Regression Spline (MARS) model (Friedman 1991) and a Particle Swarm Optimization (PSO) algorithm (Kennedy & Eberhart 1995). When applied to the chemical problem that motivated this work, the proposed approach is shown to successfully handle multiple responses and constraints on the input variables, while satisfying the practical limitations of time and resources. Significant improvements of up to 380% for the response variables are achieved. The proposed approach represents a promising methodology which is suitable for product improvement purposes in a wide range of applications extending from science to the field of engineering. Also because of the clear scientific improvements achieved on the chemical problem, scientists are showing interest in the developed procedure.

A parallel contribution is the meta-learning architecture which has been developed for model selection in multiresponse settings. The framework aims to develop a theoretical contribution to aid in moving away from a black-box approach to model selection. The purpose is not just to identify the most promising modeling technique for the problem at hand; the main aim is also the development of knowledge, from the analysis of several tasks, of the reason why a particular model performs well in certain situations (Brazdil Pavel 1998, Vilalta & Drissi 2002). The regularities encountered during the investigation of multiple problems are employed to discover the conditions under which a
given learning algorithm (or model) outperforms others, and to define some rules relating model performance and domain characteristics (Vilalta & Drissi 2002). The software R is employed to develop the meta-learning architecture, comprising instances (simulated data), features (instance characteristics), multivariate and separate multiple regression models, and performance measures. The modeling techniques taken into consideration include reduced rank regression (Izenman 2008), curds and whey regression (Breiman & Friedman 1997), partial least squares, multivariate adaptive regression splines (Friedman 1991), neural networks, linear regression, ridge regression and mixed model-based penalized spline regression (Ruppert et al. 2003). To compare the performances of the models investigated, two classes of problems are generated. The first class of problems includes datasets with characteristics similar to those of the chemical problem which motivated this work (e.g., sample size, dimensionality of the input and output spaces, constraints on the input variables, error structure). The second class of problems is obtained using a tool which was developed specifically to generate multiresponse problems with a high flexibility and increasing degrees of complexity. A set of metrics is developed to measure the characteristics of the problems under investigation and to obtain insights into the relationship between the problem characteristics and the model performance, using both direct and indirect techniques (landmarking). Performance evaluation is part of the ongoing research, and future results are expected to guide the selection of models according to the features of the application under investigation.

In summary, the contributions of this thesis are:

- the development of a one-stage distribution-free approach which is capable of:
  - designing experiments and identifying improved solutions in high dimensional settings with multiple responses;
  - overcoming the drawbacks of the existing two-stage approaches;
  - automating the target selection step in a one-stage methodology;
  - facing the challenging aspects and constraints of some real-world problems such as a high dimensional variable space, constraints on the input variables, multiple responses, expensive or time-consuming experimental trials, and unknown functional relationships between response and design variables; and
– attracting the attention of practitioners;

• the proposal of a meta-learning architecture as a research methodological framework for generating a knowledge system for the selection of multivariate response models based on the characteristics of the data; and

• the development of computer code for the future contribution of R packages (R Development Core Team 2009) for:

  – implementing the one-stage distribution-free approach;

  – simulating multivariate problems with high levels of flexibility and different degrees of complexity; and

  – implementing the meta-learning framework for the selection of multivariate models based on the features of the problem at hand.
Chapter 2

The motivating problem

The microarray is a diagnostic technology which was discovered by Schena and co-workers in the early 1990s (Schena et al. 1995). It is a very promising screening method which is useful for identifying a wide range of popular diseases (Schena 2003), including cancer (Schulze & Downward 2001) and psychiatric disorders (Bunney et al. 2003). Several microarray platforms are available nowadays, and provide different degrees of flexibility, accuracy, efficiency and cost (Chittur 2004). Various customized devices based on the microarray technology can be fabricated in properly equipped laboratories. The Coordinamento Interuniversitario Veneto per le Nanotecnologie (Associazione CIVEN, Via delle Industrie 5, Torre Hammon, 30175 Venezia-Marghera, http://www.civen.org/it/) is an association, located in Venice (Italy), which is devoted to research and experimentation in the field of nano- and bio-technologies. Microarrays are regularly manufactured and analyzed in its laboratories. Because microarray screening is still an issue, CIVEN is working to optimize its protocol. Due to the importance of surfaces in bio-screening, an extensive study has been performed in collaboration with the European Center for Living Technology (Ca’ Minich, Calle del Clero, S. Marco 2940, 30124 Venezia, Italy, http://www.ecltech.org/ecltech_j/), an international research centre which is dedicated to the study and creation of technologies inspired by life properties such as self-organization, adaptability, and the capacity to evolve and react to environmental stimuli. The chapter is organized as follows. Section 2.1 describes the functionalization problem providing details of the biomedical technology involved (the microarray technology) and
of the chemical process employed (the sol-gel chemistry). In section 2.2 the chemical system is recast into a general framework to describe the process components, state the purpose of the design of the experiments and introduce the optimization problem.

2.1 Slide functionalization for biomedical applications

2.1.1 The microarray technology

Microarrays are high capacity systems which can monitor the expression of tens of thousands of genes in parallel (Schena 2003). Each microarray consists of an ordered array of microscopic elements on a planar substrate that allows the specific bonding of genes or gene products. Gene expression studies, genotype research and DNA re-sequencing are the most diffuse applications of this modern technology. Microarrays have become crucial tools in drug discovery and life science research (Venkatasubbarao 2004). They hold also much promise for the analysis of a wide range of diseases because, if an illness is either the cause or the effect of gene modification, microarray technology can theoretically detect it (Schena 2003). The use of microarrays has been especially intensive in the case of cancer (Schulze & Downward 2001) and psychiatric disorders (Bunney et al. 2003). Various different microarray devices are available nowadays and high-throughput analysis of cDNA, oligonucleotides, proteins or tissues can be performed. Several technologies are currently available for microarray analysis. They differ in terms of the physical support employed (substrate), the length and nature of immobilized biomolecules, labeling and detection methods, fabrication, and immobilization techniques (Chittur 2004). Spotted microarray devices are our focus in this study. They are produced by depositing biological material onto a substrate (typically a glass slides) in arrayed positions (spotting process). A specific gene or gene product is associated with each position. The spotted biological material (probe) can be virtually any type of molecule of interest, including antibodies, enzymes, lipids, cell extracts, and inorganic compounds (Schena 2003). However, the most commonly used probes are PCR products and oligonucleotides. PCR products are obtained via the virtually, known as Polymerase Chain Reaction (PCR), which is an amplification method used to generate thousands to millions of copies of DNA segments (Schochetman et al. 1988). Oligonucleotides are short chains of single-stranded DNA or
RNA (Schena 2003). To bind probes efficiently, the substrate needs to be functionalized, that is, covered with a thin coating of poly-lysine, amino silanes or other reactive silanes (Schena et al. 1996). Subsequently, fluorescent-labeled cDNAs (targets) are deposited over the spotted array and zipper up (hybridize) to the complementary strands of the probes. Unhybridized cDNA molecules are washed off and, then the silica slide is scanned with a laser scanner. This therefore detects hybridized dye molecules, and digitalized images are recorded. The scanned images are composed of an arrayed grid of fluorescent circles, known as spots. The position of a spot on the grid identifies the corresponding PCR product, or gene, while the intensity of the spot (fluorescence intensity) identifies the level of expression of the associated gene. Finally, the digitalized images undergo an analysis process for the statistical selection of significantly expressed genes. The analysis process is a key issue, because spot intensities can be affected by many sources of variation, causing bias and uncertainty in the measurements (Kafadar & Phang 2003). It consists of four main steps: image analysis, normalization, detection of differentially expressed genes, and data mining (Wilson et al. 2003). Image analysis can have a potentially large impact on the subsequent steps of the analysis, such as the clustering or identification of differentially expressed genes (Yang et al. 2002). Its aim is to extract measures of the transcript abundance from microarray scanned images. This is achieved by obtaining fluorescence intensity measures for each spot. A spot’s measured fluorescence intensity includes a certain amount which is not specifically due to the hybridization of the labeled cDNA samples to the spotted biological material, e.g., fluorescence of the substrate (Yang et al. 2002). This nonspecific contribution is measured by the background intensity, which is used to correct the spot intensity. A spot’s foreground and background intensities are typically obtained by drawing a circle over the spot in the digital image. The foreground intensity signal is the fluorescence signal measured inside the spot, whereas the background signal is the fluorescence measured in the area surrounding the spot. An irregular spot shape, low and inhomogeneous fluorescence inside the spot, and high background fluorescence outside the spot, make this task hard and the results inaccurate (Yang et al. 2002). Indeed, the main source of inherent variability in the expression data is related to the spot quality (Wang et al. 2001). Ideal spots are perfectly circular and identical, have uniform signals both inside (intensity signal) and outside (background or
noise signal) the spot, and are characterized by a high signal-to-noise ratio (Wang et al. 2001, Venkatasubbarao 2004, Giannakeas & Fotiadis 2009). The microarray surface is crucial in obtaining good quality spots. Indeed, as was stated by Schena (2003), in many respects, a microarray experiment is only as good as the surface that is used to create it. The goodness of a surface depends strictly on the goodness of the coating used to functionalize it. Coatings for microarray applications are typically fabricated by sol-gel and plasma depositions (Conzone & Pantano 2004, Donggeun et al. 2006), both of which provide a variety of chemical functionalities by means of organo-functional alkoxysilane coatings. Amino and epoxy functionalities are the most widely used within the microarray industry (Conzone & Pantano 2004). The current study is devoted to the investigation of spotted microarray devices which employ a silica slide as the substrate, oligonucleotides as immobilized biomolecules (probes), and fluorescent molecules for labeling and detection. Biomolecules immobilization is achieved using amino-functionalized coatings, fabricated using the sol-gel chemistry.

2.1.2 The sol-gel chemistry

Sol-gel is defined as the synthesis of ceramic materials by the preparation of a sol, gelation of a sol, and removal of the solvent (Brinker 1990). This technique has already been applied to the production of microarray coatings because of its many attractive features, including a fine control of the chemical composition, low post thermal treatment, low equipment cost, limited steric hindrance problems associated with covalent immobilization methods, high density of reactive groups on the surface, preserved activity of immobilized biomolecules, and well-performing hybridization kinetics (Brinker 1990, Venkatasubbarao 2004). Furthermore, some studies have pointed out the possibility of patterning sol-gel amino functionalized coatings (Lee & Hajela 1996, Martnez et al. 2010, Na et al. 2008), and defined microarray technology as a possible application (Falcaro et al. 2009). Indeed, patterning technology may overcome some of the problems related to the shape of the spot, because the reactive functionalities can be located in well-defined regions (Falcaro & Innocenzi 2010). However, if the coating quality is good enough to give ideal spots without involving patterning processes, the substrate manufacturing process becomes easier and cheaper. Various different functionalized coatings can be prepared
via the sol-gel route: aldehyde, amine, epoxy and active ester groups are currently used to functionalize silica slide in order to efficiently bind biomolecules. However, amino- and epoxy-silane are the most common silanes involved in sol-gel processing for printable microarray technology (Rupcich et al. 2003). Even if different chemical reactions are involved, both functionalities promote the immobilization of DNA probes in a solid support. Aminosilane coatings in particular are characterized by a high concentration of primary amino groups on the coated surface, which become positively charged when placed in contact with a near-neutral aqueous solution. This positively charged surface is well suited to the ionic attraction of negatively charged DNA probes (Schena 2003, Wilson et al. 2003, Conzone & Pantano 2004). The objective of this study is to obtain sol-gel functionalized coatings which result in a film giving ideal spots. Amino-methylsilane coatings are obtained by using a mixture of aminopropyltriethoxysilane (APTES), methyltriethoxysilane (MTES) and tetraethylorthosilicate (TEOS); other compositional variables such as water (H₂O), hydrochloric acid (HCl) and ethanol (EtOH) have also been considered. Pre-cleaned silica slides are dipped into the chemical solution which is produced by mixing APTES, MTES, TEOS, HCl, EtOH, and H₂O (dip-coating deposition), so that the slide is covered with a thin film. A well-know problem related with sol-gel systems involving amino-silanes is the possible transition from sol to gel. Indeed, the additional basic catalyst effect of the aminosilane drastically increases the cross-linking rate of siloxane units, and this results in a transition from sol to gel, that can occur in less than a couple of minutes. This rapid transition affects the film properties (i.e. homogeneity, grafting capacity, thickness), therefore, the same chemical composition will hardly produce coatings with similar features. Due to the preliminary nature of this study, a dye molecule, known as fluorescein isothiocyanate (FITC) is employed. The use of FITC molecules has the double advantage of reducing the experimental cost and simplifying the laboratory protocol. In particular, due to its chemical affinity with the amino group, FITC can simulate the behavior of probe molecules, and constitutes a cheap substitute for the oligonucleotides. Furthermore, a cheaper, simpler and faster laboratory protocol is obtained, because the fluorescent signal emitted by the FITC molecules can be used as a measure of the intensity. The hybridization between probes and fluorescent-labelled target molecules is therefore no longer required (Falcaro et al. 2009).
Materials

Tetraethyl orthosilicate (TEOS), 3-aminopropyltriethoxysilane (APTES), methyltriethoxysilane (MTES), chloridric acid (HCl 1M), ethanol (EtOH) and fluorescein isothiocyanate (FITC) were purchased from Aldrich and used as-received. P-type/boron doped, (100) oriented, 400 µm thick silicon wafers (Si-Mat) and silica slides (76 mm × 26 mm, 1.2 mm thick from knittel Glaser) were used as substrates. The silicon wafers were rinsed with EtOH and acetone. Each silica slide was cleaned using a solution obtained by mixing H$_2$O$_2$ (25 mL) and NH$_4$OH (25 mL). After 5 minutes under sonication, the slides were rinsed with water, EtOH, and acetone, and then dried under a nitrogen gas flux prior to use. The water was purified by a Milli-Q Synthesis A10 system. The process used to synthesize the functional coatings consists of a standard sol-gel synthesis: TEOS, MTES, APTES, a proper solvent (EtOH), deionised water (H$_2$O) and hydrochloric acid (HCl 1M) were mixed under stirring conditions. Aminopropyltriethoxysilane (APTES), methyltriethoxysilane (MTES) and tetratethoxysilane (TEOS) were used as the silane mixture. Two constrains have been imposed: the total amount of moles of the three precursor are kept equal to a value in the range $[4.5 \cdot 10^{-3}, 5 \cdot 10^{-3}]$, while the number of TEOS moles is required to be more than $2.5 \cdot 10^{-3}$. The components are added one at a time in the following order: EtOH, TEOS, MTES, H$_2$O, HCl 1M. This solution is then stirred for 15 minutes to partially hydrolize the silane precursors, after which APTES is added dropwise and kept under vigorous stirring conditions for 20 minutes (aging time, $t_a$). If no gelation phenomenon occurs during $t_a$, then the solution is used for the dip-coating deposition process (2.5 mm $s^{-1}$ withdrawal speed, 20% relative humidity). The films are then dried for 30 minutes at 100°C.

Measurements

Amino functionalized silica slides are spotted using a FTA 1000 contact angle tester in order to control a 50 µL drop deposition, to simulate the circular shapes obtained using microarray technology. Three drops are deposited for each slide (see Figure 2.1). Both the distance between the substrate and the needle, and the pumping rate used to produce the drop have been kept constant for all of the spotting processes (respectively 3 mm and 5 µL $s^{-1}$). Triton X has been added to a sodium phosphate buffer solution to simulate the
Amino functionalized slides are spotted using a contact angle tester, and three drops are deposited for each slide. The coating with the 3 drops is left in a closed vessel (70% relative humidity and 40°C) for 12 hours, then the excess dye is removed in an optimized two-step process: first, the slide is dipped in a 50 mL closed vessel with 40 mL of methanol for 10 minutes; second, the slide is dipped in a 50 mL closed vessel with 40 mL of water for 10 minutes. The coating is then dried under nitrogen.

The performances (spot circularity, foreground intensity, background intensity, and spot homogeneity) were measured as described further below. A circle with the largest possible area is inscribed on each spot, allowing us to define a centre C and a minimum radius.
(min$_R$) (Figures 2.2c and 2.2d). A specific point along the perimeter of the spot shape $P$ is then chosen in order to define the maximum distance $CP$ (max$_R$). The $min_R/max_R$ ratio measures the spot circularity (Figures 2.2c and 2.2d) which is equal to one in the case of perfect circularity (Figure 2.2d). The spot intensity is calculated by averaging the fluorescence intensity values inside the yellow dotted circular area (Figures 2.2c and 2.2d). The background intensity is obtained by averaging 60 intensity measurements along the blue dashed circumference (about 4/3 of the maximum dimensions of the spot, see Figures 2.2c and 2.2d). The three intensities with the maximum values (yellow pointers in Figures 2.2e and 2.2f) are averaged to give the value of the foreground intensity. The three pixels with the minimum values (red pointers in Figures 2.2e and 2.2f) are averaged to calculate the background intensity. The minimum to maximum intensity ratio within the circumference defines the homogeneity of the intensity signal inside the spot.

2.2 The multiresponse high-dimensional chemical system: design of experiments and optimization

A process can usually be visualized as a combination of resources (e.g., operations, machines, methods, and people) that transforms some input (often a material) into an output (Montgomery 2009). The characteristics of the output are typically measured by one or more observable response variables which are influenced by the input variables. When the number of the inputs is large and if multiple responses are of interest, then the process is referred to as multiresponse high dimensional system. Either screening experiments or the practitioner’s knowledge can be used to reduce the system dimensionality and select the critical inputs, which are the variables that are most influential on the responses (Montgomery 2009). The objective here is process optimization, which consists of determining the region, in the important inputs that lead to the best possible responses. To locate the optimum, it is necessary to perform experiments that vary all of the inputs simultaneously (Montgomery 2009). In the design of the experiments, the input variables are referred to as factors and the values of the factors are referred to as levels. The levels over which the response is investigated define the experimental region. A given point in that region corresponds to a specific combination of factor levels and is referred to as
Figure 2.2: Spot circularity, intensity, background and homogeneity are measured, and information about the coating quality is obtained.

design point. Once a set of design points has been selected, an experimental design is specified and the responses can be measured in relations to the identified points (Santner et al. 2003). Let $X = (X_1, \ldots, X_p)^T$ be a vector of $p$ factors and let $Y = (Y_1, \ldots, Y_q)^T$ be a vector of $q$ responses. Denote by $E$ the experimental region and by $D = (x_1, \ldots, x_S)$ an experimental design with $S$ design points at which the responses $Y_j, j = 1, \ldots, q$ will be observed. Here, $x_s = (x_{s1}, \ldots, x_{sp})$, for $s = 1, \ldots, S$. Finally, the observed responses are denoted by $Y = (y_1, \ldots, y_S)$, where $y_s = (y_{s1}, \ldots, y_{sq})$, $s = 1, \ldots, S$. Because the responses of interest are influenced by the factors, we can write

$$Y = f(X) + \epsilon,$$  \hspace{1cm} (2.2.1)
where $\epsilon$ represents the noise or error observed in the response $Y$. The purpose is to determine the vector $x^*$ that minimizes (or maximizes) a given set of $q$ objective functions $f(x^*) = \{f_1(x^*), \ldots, f_q(x^*)\}$, subject to bounds on the input variables and constraints of the form $g(x^*) = b$ and/or $h(x^*) \neq c$, which restrict the solution space. In many real-life problems, the objectives under consideration conflict with each other. Hence, optimizing $f(X)$ with respect to a single objective often results in unacceptable solutions with respect to the other objectives. Therefore, it is almost impossible to obtain a perfect multi-objective solution that simultaneously optimizes each objective function (Konak et al. 2006).

In the chemistry process investigated, the resources are the material scientists, the laboratory instruments, the sol-gel chemistry, and so on. The inputs are compositional variables (EtOH, H$_2$O, HCl, APTES, MTES, and TEOS) and process variables (withdrawal speed, relative humidity, heating temperature, heating time, and so on). Because process variables are less likely to affect the coating functionality and because their effect on the coating features is known, process variables are kept constant, and only compositional inputs are investigated. A preliminary study of 40 experiments is conducted to estimate the time and cost of each single experimental trial and to define the ranges and levels of the compositional variables. It has been estimated that the whole process required to generate a single coated slide costs about 30 hours and 300 euros. The chosen ranges and levels of the design variables are reported in Table 2.1. They are determined with the purpose of considering a wide range of compositions. Simultaneously, practical limitations, such as costs and the precision of the laboratory instruments, are taken into consideration. Also taken into account are experience-based aspects: only levels of the input variables which are likely to generate detectable changes in the sol-gel mixture are used. To keep the ratios between the chemical components of the mixture almost constant, two constraints on the number of moles of some compositional variables are defined:

$$\begin{align*}
\text{APTES} + \text{MTES} + \text{TEOS} & \in [4.5 \cdot 10^{-3}, 5 \cdot 10^{-3}] \\
\text{TEOS} & > 2.5 \cdot 10^{-3}.
\end{align*}$$

(2.2.2) (2.2.3)

The system output is the coated substrate, and its characteristics are measured by four response variables. They are spot circularity (Cir), foreground intensity (Int), background
Table 2.1: Factor levels (number of moles (M) and micromoles (mM) investigated to generate the sol-gel recipes)

<table>
<thead>
<tr>
<th></th>
<th>EtOH (M)</th>
<th>H₂O (M)</th>
<th>HCl (M)</th>
<th>TEOS (mM)</th>
<th>MTES (mM)</th>
<th>APTES (mM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.170</td>
<td>0.014</td>
<td>0.010</td>
<td>2.19</td>
<td>0.25</td>
<td>0.21</td>
<td></td>
</tr>
<tr>
<td>0.306</td>
<td>0.039</td>
<td>0.028</td>
<td>2.63</td>
<td>0.70</td>
<td>0.58</td>
<td></td>
</tr>
<tr>
<td>0.442</td>
<td>0.064</td>
<td>0.046</td>
<td>3.07</td>
<td>1.15</td>
<td>0.97</td>
<td></td>
</tr>
<tr>
<td>0.578</td>
<td>0.089</td>
<td>0.064</td>
<td>3.51</td>
<td>1.60</td>
<td>1.34</td>
<td></td>
</tr>
<tr>
<td>0.714</td>
<td>0.114</td>
<td>0.082</td>
<td>3.95</td>
<td>2.05</td>
<td>1.72</td>
<td></td>
</tr>
<tr>
<td>0.850</td>
<td>0.139</td>
<td>0.100</td>
<td>4.39</td>
<td>2.50</td>
<td>2.10</td>
<td></td>
</tr>
</tbody>
</table>

intensity (Back), and homogeneity of the intensity signal inside the spot (Hom). A fifth response variable, Stab, is also recorded to measure the stability of the sol-gel mixture. It is a dichotomous variable which assumes a value of one if the sol-gel mixture does not change into gel within a predefined period of time, and zero otherwise. The four responses measuring the coating quality are not considered to be equally relevant, so a set of normalized weights is assigned to them. Higher weights are associated with response variables of higher relevance. Specifically, a weight of 0.4 is assigned to Cir and Int, a weight of 0.12 is assigned to Back, and a weight of 0.08 is assigned to Hom. The response variables Stab, Cir, Int, and Hom should be maximized, whereas Back should be minimized. The solution of the multi-objective optimization problem needs to take into consideration the conflicting nature of the responses, and sub-optimal quality values are therefore expected. Let \( X = (X_1, \ldots, X_6) \) be the 6-dimensional vector of factors where \( X_1 = \text{EtOH}, X_2 = \text{H}_2\text{O}, X_3 = \text{HCl}, X_4 = \text{APTES}, X_5 = \text{TEOS}, \) and \( X_6 = \text{MTES}, \) and let \( Y = (Y_1, \ldots, Y_5) \) be the 5-dimensional vector of responses with \( Y_1 = \text{Stab}, Y_2 = \text{Cir}, Y_3 = \text{Int}, Y_4 = \text{Back}, \) and \( Y_5 = \text{Hom} \). The goal is to determine the vector \( x^* = (x_1^*, \ldots, x_p^*) \) of factor levels that maximizes the set of \( q = 5 \) objective functions \( f(x^*) = \{y_1(x^*), y_2(x^*), y_3(x^*), 1 - y_4(x^*), y_5(x^*)\} \) subject to the constraints in equations (2.2.2) and (2.2.3). The optimization is carried out within the experimental region defined by the ranges reported in Table 2.1. The investigated optimization problem poses challenges due to the following features: wide operability region, irregular design region due to the constraints on the compositional variables, multiple responses, unknown functional relationship between the factors and
the response variables, and time consuming and expensive experiments. Furthermore, the functional relationship between the factors and the response variables is expected to be non-linear.
Chapter 3

Nonparametric and semiparametric response surface methodology

This chapter provides a literature review on the Nonparametric and Semiparametric Response Surface Methodology (NPRSM), a technique which belong to the experimental design field, with roots in the Response Surface Methodology (RSM) and the Design and Analysis of Computer Experiments (DACE) (Myers et al. 2004, Myers 1999, Santner et al. 2003, Fang et al. 2006). In what follows, the three methodologies (RSM, DACE, and NPRSM) are briefly described, to provide a self-contained overview of the research field of interest. More details are given by Box & Wilson (1951), Box & Draper (1987), Myers et al. (1989), Myers (1999), Khuri & Cornell (1996), Myers et al. (2004), Montgomery (2009), Sacks, Welch, Mitchell & Wynn (1989), Sacks, Schiller & Welch (1989), Santner et al. (2003), and Fang et al. (2006). Subsequently, NPRSM experimental design, modeling and optimization techniques are reviewed. This chapter is organized as follows. Section 3.1 introduces the designs employed in NPRSM and provides a comparison with traditional designs as well. Section 3.2 describes the most popular modeling strategies in NPRSM, together with the background theory of mixed model-based penalized spline models. To the best of our knowledge, this class of models has not previously been used for response surface approximation, and is described here because of the great promise it shows. Finally, section 3.3 describes optimization in NPRSM. Developments in the multiresponse field are pointed out at the end of the section on design and optimization techniques. A detailed
description of modeling techniques for systems with multiple responses is provided in Chapter 5.

Let’s first introduce the formal notation that will be used throughout the chapter. Let $X = (X_1, \ldots, X_p)^T$ be a $p$-dimensional vector of inputs and let $Y$ be the real-valued response. Assume a set of training data $(x_1, y_1), \ldots, (x_N, y_N)$ is given. Then, denote by $X$ the $N \times p$ matrix of observed inputs with each row given by $x_n^T = (x_{n1}, \ldots, x_{np})$, $n = 1, \ldots, N$, and column vectors given by $x_1^T, \ldots, x_N^T$. Similarly, denote by $y = (y_1, \ldots, y_N)^T$ the $N$-dimensional vector of measured responses.

Response Surface Methodology (RSM) is a collection of statistical design and numerical optimization techniques which are used to optimize processes and product designs (Myers et al. 2004). The method dates back to the 1950s, when the work by Box & Wilson (1951) was published. According to Mead & Pike (1975), many of the fundamental ideas had been used and discussed since the 1930s. However, there is no question that the paper by Box & Wilson (1951) gave rise to an entire field of research which is still evolving today (Myers et al. 2004). The work by Box & Wilson (1951) was motivated by the ancient problem of planning and analyzing experiments in search of desirable conditions on a set of design variables (Myers et al. 1989). RSM is concerned with estimating a surface from what is typically a small set of observations, with the aim of determining what are the levels of the input variables that maximize the response (Balkin & Lin 2000). The experimental strategy and analysis in RSM revolves around the assumption that a response $Y$ is a function of a set of design variables $X_1, X_2, \ldots, X_p$, and that the function can be approximated in some region of the inputs by a polynomial model (Myers et al. 1989). The most popular approximating functions are the first-order model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon \quad (3.0.1)$$

and the second-order model

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \sum_{i=1}^{p} \beta_{ii} X_i^2 + \sum_{i<j} \beta_{ij} X_i X_j + \epsilon. \quad (3.0.2)$$
(Montgomery 2009). Generally, the assumption of a common error variance $\sigma^2$ is made, and coefficients are estimated by ordinary least squares (Myers et al. 1989). The model parameters can be estimated most effectively if proper experimental designs are used to collect the data (Montgomery 2009). The designs for fitting response surfaces are known as response surface designs. The most popular are the orthogonal design, which includes the $2^k$ factorial and fractional factorial designs, the central composite design (CCD), the small composite design, and the Box-Behnken design (Box & Behnken 1960). Because the interest is in a restricted area of the search space, polynomial models usually provide a reasonable approximation of the true functional relationship in that sub-area (Montgomery 2009). The basic hypothesis is that if the fitted surface is an adequate approximation of the true response function, its analysis will be approximately equivalent to the analysis of the actual system (Montgomery 2009). RSM is a sequential procedure that aims to lead the experimenter rapidly and efficiently to the general vicinity of the optimum. Typically, a first-order model (equation (3.0.1)) is fitted at the initial stages of the procedure, since the region under investigation is likely to be remote from the optimum, and as a consequence, little curvature is detected in the system. The method of steepest ascent is subsequently used to guide the search towards the most promising area of the operability region. The direction of the steepest ascent is normal to the fitted response surface, and corresponds to the direction in which the fitted response increases most rapidly. The steps along the path are proportional to the estimated regression coefficients. The procedure is iterated until the lack of fit of a first-order model indicates the vicinity of the optimum condition (Montgomery 2009). Once the region of the optimum has been found, a second-order model (equation (3.0.2)) is fitted and an analysis is performed to locate the optima (see Figure 3.1). During this whole procedure, the information obtained in one stage is used to plan the next stage. Additional runs might also be necessary to gain more information about the response in certain parts of the region that are not covered adequately by the initial design (Myers et al. 1989). Sequential design and additional experimental runs belong to the concept of design augmentation (Myers et al. 1989).

The Design and Analysis of Computer Experiments (DACE) is a widely studied statistical approach introduced by Sacks, Welch, Mitchell & Wynn (1989). It has been employed extensively in the engineering field for expensive design optimization problems such as
Figure 3.1: RSM is sequential in nature. Initially, a first-order model is fitted (current operating conditions). Then, the method of steepest ascent is used to guide the search towards the most promising area of the operability region. Once the region of the optimum has been found, a second-order model is fitted and an analysis is performed to locate the optima (Montgomery 2009).

computational fluid dynamics, the study of electronic circuits, and finite element analysis of the stress on a mechanical product (Bates et al. 1996, Grierson & Pak 1993, Lee & Hajela 1996, Büche et al. 2004, Pierrot 1999, Jin, Olhofer & Sendhoff 2001, Jin et al. 2002). When processes are so complex that physical experimentation is too time consuming, too expensive, or utterly impossible to carry out (as in the case of weather modeling), mathematical models are employed to simulate the system (Sacks, Welch, Mitchell & Wynn 1989). Despite the advances in computer capacity, complex engineering simulations might still be impractical because of the high computational cost. In DACE, a metamodel (model of the model) or surrogate is built using the data from an experimental design, and then optimized. As a result, the physical process is modeled by a large simulator, and experiments are conducted on the computer code directly (Bates et al. 1996). Both stochastic and deterministic models are used to approximate the system behavior. Experimentation with a stochastic simulation model is much like experimentation with a physical system (Myers et al. 2004). In contrast, experimentation with a deterministic computer model differs from physical experimentation because no random error is associated with the output and, as a consequence, the same response values will be obtained.
when running the code with the same inputs (Sacks, Welch, Mitchell & Wynn 1989). The use of deterministic models was introduced and pursued by Sacks, Welch, Mitchell & Wynn (1989), who proposed the use of a statistical model that treats the response as if it were a realization of a stochastic process. The statistical model adopted by Sacks, Welch, Mitchell & Wynn (1989) is referred to as the Gaussian stochastic process (GP) model, and is known as kriging in the geostatistics and other spatial statistics literature. The GP model is discussed in section 3.2. It has the double benefit of providing a statistical basis for computing an efficient predictor of the response at untried inputs and allowing estimates of predictor uncertainty. Various design criteria and sequential algorithms with adaptations to the data are proposed by Sacks, Welch, Mitchell & Wynn (1989). The idea is to choose a design that predicts the response well at untried inputs of the experimental region and augment it with additional design points. Design augmentation aims to put some emphasis on sampling where the predictor uncertainty is high, so that attention can be paid to parts of the space that are relatively unexplored (Jones 2001). In contrast to the standard RSM, computer experiments seek to find a model that approximates the true response surface over a much wider range of the design variables, sometimes extending over the entire region of operability. As a consequence, models much more complex than first- and second-order polynomials are required (Montgomery 2009).

NPRSM is suitable for situations where the response is not accommodated by polynomial models in the operability region (Myers et al. 2004, Myers 1999). This can potentially occur when one wishes to fit a model that covers a portion of the factor space which is greater than the portion that is typically investigated in standard RSM applications (Myers et al. 2004). The term ‘nonparametric RSM’ implies that there is no specific model and that the primary focus of the analysis is on prediction, which leads to the use of kernel-based regression techniques. Semiparametric RSM signifies that a model is used, but that it is not a standard polynomial model (Myers et al. 2004). As was pointed out by Myers (1999), there are several scenarios which suggest the use of nonparametric and semiparametric RSM. They are:

1. optimum conditions are required;
2. there is less interest in an interpretative function and more in the appearance of the response surface;

3. the function is highly nonlinear, but not necessarily well behaved;

4. there is no need for the designs constructed to honor ‘model form’; rather the designs must be from a space-filling grid.

Whereas RSM and NPRSM differ in many aspects (design, modeling and optimization techniques), the distinction between DACE and NPRSM is less clear. The main difference lies in their ‘native’ applicative sector. DACE was developed as a modeling technique for coping with expensive simulations of complex engineering systems (Sacks, Welch, Mitchell & Wynn 1989), whereas NPRSM has primarily been applied in the biopharmaceutical and biotechnological fields (Myers et al. 2004). Deterministic Gaussian stochastic process models and sequential sampling were originally employed in DACE, while other models, such as neural networks and thin plate splines, have subsequently been introduced. In what follows, DACE will be considered as belonging to the NPRSM class. A parallel body of research, with roots in DACE and NPRSM, has been developed in the field of automatic algorithm parameter optimization. Efficient Global Optimization (Jones et al. 1998) and Sequential Parameter Optimization (Bartz-Beielstein et al. 2005) are the main techniques which have been developed in this field. Their purpose is to quantify the importance of each parameter, as well as the interactions between parameters, and — in the context of multiple instances — between parameters and instance characteristics. The interpolation of performances between parameter settings and the extrapolation to previously unseen regions of the parameter space are also supported. These techniques can aid substantially in the understanding or improvement of an algorithm (Hutter et al. 2009, 2010).

### 3.1 Experimental designs

NPRSM applications often use standard RSM designs such as factorial and central composite designs (Myers et al. 2004). However, standard response surface designs are usually the best choice if the experimenter is considering a polynomial model and if the design region is regular. A regular design region can be represented as a space which is bounded
by the $p$-dimensional hypercube or hypersphere, with any point on or inside the cube or sphere being a candidate design point. Practical problems of the feasibility, operability, and availability of the system being studied might make it necessary to chop corners off the cubes, take slices off the spheres, or dig holes in the design region (Kennard & Stone 1969). Furthermore, in the case of expensive or time-consuming experiments, the number of trials that can be realized is limited. This number is typically small relative to the number of experiments that are conceptually possible. When both the model and the number of realizable trials are fixed, almost all RSM designs lead to an excessively high number of replicates required (Kennard & Stone 1969). If nonstandard situations occur, such as an irregular experimental region or unusual sample size requirements, computer-generated designs are preferable (Montgomery 2009). Computer-generated designs are constructed using computer programs. They are optimal with respect to some criterion, and are therefore referred to as optimal designs. Common optimality criteria are D-optimality, A-optimality and G-optimality (Montgomery 2009). A- and D-optimality are concerned with the estimation of the regression coefficients (Montgomery 2009), and involve minimizing a function of the covariance matrix of the least squares estimates of the parameters, namely the determinant (D-optimality or generalized variance) and the trace (A-optimality or average variance) (Santner et al. 2003). G-optimal designs are concerned with the prediction of the response, and aim to minimize the maximum scaled prediction variance over the design region (Montgomery 2009). The usual approach in computer-generated designs is to specify a model (typically a polynomial), determine the region of interest, select the number of runs to make, specify the optimality criterion, and then choose the design points from a set of candidate points that the experimenter would consider using. The candidate points are typically a grid of points spaced over the feasible design region (Montgomery 2009). With the advent of DACE, other criteria-based designs have also been developed. They are more difficult to implement than the aforementioned designs. Indeed, a Gaussian stochastic process model (see section 3.2) is usually assumed, and the design criteria are functions of its unknown parameters (Santner et al. 2003). Furthermore, analytical results are difficult to obtain, and have been found in only a few special cases (Santner et al. 2003). For a fixed number of runs and for a specific correlation structure, these design criteria aim to choose a design which will predict the response
well at untried inputs in the experimental region $E$ (Sacks, Welch, Mitchell & Wynn 1989). Functions of the mean squared prediction error (MSPE) are considered, such as the Integrated Mean Squared Error (IMSPE) and the Maximum Mean Squared Error (MMSPE) (Sacks, Welch, Mitchell & Wynn 1989). Given an experimental design $D = (x_1, \ldots, x_S)$ in the experimental region $E$, and the data $y = (f(x_1), \ldots, f(x_S))$, let $\hat{y}(x)$ be the predictor of $y(x)$ at an untried $x$. Then, the IMSPE criterion chooses the design $D$ to minimize

$$\int_E \text{MSPE}[\hat{y}(x)] \phi(x) dx$$

(3.1.1)

for a given weight function $\phi(x)$, whereas the MMSPE criterion chooses the design $D$ to minimize

$$\max_{x \in E} \text{MSPE}[\hat{y}(x)].$$

(3.1.2)

The IMSPE and MMSPE design criteria are generalizations of the classical A-optimality and G-optimality (Sannter et al. 2003). A third criterion is the maximum entropy. It aims to maximize the information $I$ about a parameter $\theta$ in a statistical model. The parameter is believed to describe a measured response, and in the Gaussian process models (see section 3.2) it corresponds to the vector of correlation parameters (Sannter et al. 2003). Knowledge about $\theta$ is specified by a probability density or mass function $[\theta]$, which could be either a prior distribution or a posterior distribution. When the design space is discrete, the maximum entropy criterion is as follows. Let $[\theta|D]$ denote the posterior distribution of $\theta$, given the data obtained using the design $D$. The amount of information in $[\theta]$ about $\theta$ before an experiment is

$$I = \int [\theta] \ln(\theta) d\theta,$$

(3.1.3)

and the amount of information about $\theta$ after the experiment using design $D$ is

$$I_D = \int [\theta|D] \ln(\theta|D) d\theta.$$  

(3.1.4)

To evaluate a design, the expected change in information, $I - I_D$, is considered. Because entropy is the opposite of information, the design $D$ is chosen to maximize the entropy of the observed responses at the points in the design (Sannter et al. 2003). Along with criteria-based designs, designs which are capable of filling up the factor space in a uniform
fashion have also been developed (Kennard & Stone 1969, Bates et al. 1996). Because prediction accuracy is the primary concern of NPRSM (Santner et al. 2003), space-filling designs are of particular interest. Indeed, designs that concentrate points on the boundary of the experimental region (such as factorial designs, CCD and D-optimal design), can yield predictors that perform quite poorly in portions of the experimental region that are sparsely observed (Santner et al. 2003). On the other hand, space-filling designs can be a systematic way of scientifically discovering surprising behaviors, because they exercise the code over the entire range of each input variable (Iman & Helton 1988). The simplest strategy for filling an experimental region \( E \) is to select a simple random sample of points from it. When the sample size is small and the experimental region is high-dimensional, Latin Hypercube (LH) designs are preferable (Santner et al. 2003). A LH design with \( S \) runs and \( p \) input variables is an \( S \times p \) matrix in which each column is a random permutation of \( \{1, 2, \ldots, S\} \) (Fang et al. 2006). The procedure for obtaining a LH sample of size \( S \) from \( X \) when the input variables are independently distributed is as follows. The domain of each input variable is divided into \( n \) intervals. The set of all possible Cartesian products of these intervals constitutes a partitioning of the \( p \)-dimensional sample space into \( S^p \) ‘cells’. A set of \( S \) cells is chosen from the population of \( S^p \) cells in such a way that the projections of the centers of each cells onto each axis are spread uniformly across the axis; then a point is chosen at random in each selected cell. For more information about LH designs, see McKay et al. (1979), Santner et al. (2003), Fang et al. (2006). Along with sampling methods, space-filling designs based on measures of distance have also been developed. Their purpose is to avoid points in the design to being too close together. One example is the maximin distance design, which employs the \( p \)th order distance between two points \( w \) and \( x \) in the experimental region \( E \subset \mathbb{R}^p \). For \( d \geq 1 \), an important distance measure is defined by

\[
\rho_d(w, x) = \left[ \sum_{j=1}^{p} |w_j - x_j|^d \right]^{1/d},
\]

(3.1.5)

which corresponds to the Rectangular distance for \( d = 1 \) and the Euclidean distance for \( d = 2 \) (Santner et al. 2003). A third approach for filling an experimental region is comparing the distribution of the design points to the uniform distribution. This results in the well-known uniform designs. To measure the extent to which the design \( D \) differs
from a completely uniform distribution of points, the notion of the discrepancy of a design $D$ is introduced. Let $F_n$ be the empirical distribution function of the points in $D$, then the famous Kolmogorov-Smirnov statistic

$$\sup_{x \in E} |F_n(x) - F(x)|$$

is used as a measure of the discrepancy, and the design $D$ that minimize equation (3.1.6) is chosen (Santner et al. 2003). Various other space-filling designs have also been developed, including sphere-packing (Johnson et al. 1990) and lattice (Bates et al. 1996).

### 3.1.1 Multiresponse experimental design

The design problem is more complex than in the presence of multiple responses than in the case of a single response. As was stated by Khuri & Cornell (1996), a design which is efficient for one response may not be efficient for the other responses, and the choice of a design should be based on a criterion which incorporates measures of efficiency pertaining to all of the responses. The results in this area are restricted to criteria for parameter estimation, design robustness, and multivariate lack of fit, when linear multiresponse models are assumed (Khuri & Cornell 1996). The single-response D-optimal design, has been generalized to obtain multi-response D-optimal designs but the computational cost of this procedure limits its applicability (Khuri & Cornell 1996). To overcome this drawback, Federov (1972) developed a sequential procedure which is capable of dealing with both known and unknown variance-covariance matrices. Draper & Hunter (1966) proposed a Bayesian approach to obtain a design criterion for the selection of $s$ additional experimental runs after $S$ runs have already been performed (Khuri & Cornell 1996).

### 3.2 Response surface modeling

Although they are convenient and generally useful, polynomial models are frequently unappealing in NPRSM. This is due primarily to the more global nature of the fit and the necessary extrapolation in the corners of the design (O’Connell & Wolfinger 1997). Indeed, when a more global representation of the system is desired, the problem complexity and nonlinearity arise. Furthermore, the behavior of the fit of polynomials to the data tends to
be erratic near the boundaries, and extrapolation can be dangerous (Hastie et al. 2009). As a consequence, first- and second-order linear models give way to highly adaptive models which are more suitable to the complex non-linear responses and the high dimensionality of the input space (Bates et al. 1996). The most popular methods used for response modeling in NPRSM are Gaussian stochastic process models (GP), thin plate splines (TPS), and neural networks (NN) (Myers et al. 2004). A less diffuse but still relevant approach is the Multivariate Adaptive Regression Splines (MARS) procedure introduced by Friedman (1991). Another worthwhile methodology that has not yet been employed in the field of NPRSM, to the best of our knowledge, is the mixed model-based penalized spline regression (MMPS). The listed methods are described and discussed further below.

Before proceeding, it is worth noting that all of the aforementioned modeling techniques move beyond linearity, augmenting or replacing the vector of inputs $X$ with additional variables, which are transformations of $X$, and then use linear models in this new space of derived input features. Because the models are linear in the derived input features, the approach has the great advantage that the fitting proceeds as in the simple linear case.

Let $h_m(X) : \mathbb{R}^p \rightarrow \mathbb{R}$ be the $m$th transformation of $X$, $m = 1, \ldots, M$. Then

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X). \tag{3.2.1}$$

Increased flexibility is obtained using piecewise-polynomials and splines, which allow for local polynomial representations (Hastie et al. 2009). Let us consider the 1-dimensional case ($p = 1$). A piecewise liner fit is obtained using basis functions of the form

$$h_1(X) = 1, \quad h_2(X) = X, \quad h_3(X) = (X - \xi_1)_+, \quad h_4(X) = (X - \xi_2)_+, \tag{3.2.2}$$

where $t_+$ denotes the positive part, and $\xi_1, \xi_2$ are referred to as knots. When used in equation (3.2.1), functions of the form of equation (3.2.2) give rise to a piecewise linear fit which is continuous at the two knots. Continuity is obtained by imposing linear constraints on the parameters (e.g., $\beta_1 + \xi_1 \beta_4 = \beta_2 + \xi_1 \beta_5$) derived from equalities of the form

$$f(\xi_1^-) = f(\xi_1^+). \tag{3.2.3}$$
Figure 3.2: Examples of piecewise basis functions. A piecewise-linear basis function \((X - \xi_1)_+\) with the knot \(\xi_1\) is depicted on the left, whereas a piecewise-polynomial basis function \((X - \xi_1)_+^3\) with the knot at \(\xi_1\) is shown on the right.

The basis functions in equation (3.2.2) constitute a way of directly incorporating such constraints on the fit. By increasing the order of the local polynomial, smoother functions are obtained. An example of a piecewise-linear and a piecewise-polynomial basis function is depicted in Figure 3.2. When the piecewise-polynomial function has continuous first and second derivatives at the knots, it is referred to as a cubic spline (Hastie et al. 2009). For example, a cubic spline with knots at \(\xi_1\) and \(\xi_2\) can be obtained using the following basis

\[
\begin{align*}
  h_1(X) &= 1, \\
  h_2(X) &= X, \\
  h_3(X) &= X^2, \\
  h_4(X) &= X^3, \\
  h_5(X) &= (X - \xi_1)_+^3, \\
  h_6(X) &= (X - \xi_2)_+^3.
\end{align*}
\]

If additional constraints are added in order to make the function linear beyond the boundary knots, a natural cubic spline is obtained. The general form for the truncated-power basis set is given by

\[
\begin{align*}
  h_j(X) &= X^{j-1}, & j &= 1, \ldots, M, \\
  h_{M+l}(X) &= (X - \xi_l)_+^{M-1}, & l &= 1, \ldots, K.
\end{align*}
\]

An important class of basis functions, radial basis functions, is obtained when \(M\) is odd and \(h_{M+l}(X) = |X - \xi_l|_+^{M-1}\) is used in equation (3.2.5). Note that \(|X - \xi_l|_+^{M-1}\) can also be
Radial Basis Function

Figure 3.3: Example of a radial basis function $|X - \xi_1|_+$ with a knot at $\xi_1$.

written as

$$|X - \xi_l|^{M-1} = r(|X - \xi_l|),$$  \hspace{1cm} (3.2.6)

where $r(u) = u^{M-1}$ (Ruppert et al. 2003). Radial basis functions depend only on the distance $|X - \xi_l|$ and the univariate function $r$, and because of this property they can easily be extended to higher-dimensional predictor variables. In particular, if $X \in \mathbb{R}^p$ and $\xi_l$ are knots in $\mathbb{R}^p$ with $l = 1, \ldots, L$, then

$$h_{M+1}(X) = r(||X - \xi_l||),$$  \hspace{1cm} (3.2.7)

where $||v|| = \sqrt{v^Tv}$ (Ruppert et al. 2003). An example of a radial basis function is depicted in Figure 3.3. As we will see further below, GP models and TPS use radial basis functions, whereas MARS models use piecewise linear basis functions. Instead, NNs use a basis expansion of the original inputs $X$, with the enhancement that the parameters of the basis functions are learned from the data (Hastie et al. 2009). Truncated-power basis, natural cubic splines and thin plate splines belong to the set of basis functions that can be used in MMPS.

Typically, a very large number of basis functions can be produced, and very complex models of the form of equation (3.2.1) are generally obtained. To control this complexity, different approaches have been developed. An initial approach belongs to the group of
restriction methods and consists of limiting the class of functions used to build the model (such as when additivity is assumed). A second approach belongs to the group of selection methods, on the basis of which only the basis functions \( h_m \) which contribute significantly to the fit of the model are included. A third approach belongs to the group of regularization methods, and consists of restricting the coefficient values (like in ridge regression). We will see that MARS uses a selection method to control the model complexity, whereas TPS, NN and MMPS employ a regularization approach. Smoothing splines and penalized splines are two regularization methods which have been developed to overcome the knot selection problem. This requirement is due to the fact that, to use fixed-knot splines in equations (3.2.4)–(3.2.6), it is necessary to select the order \( M \) of the spline, the number \( l \) of knots (or degrees of freedom) and their placement. Smoothing splines use a maximal set of knots and control the complexity of the fit by regularization. For \( p = 1 \), the problem typically consists of finding the function, among all of the functions \( f(X) \) with two continuous derivatives, that minimizes the penalized residual sum of squares

\[
\text{RSS}(f, \lambda) = \sum_{n=1}^{N} (y_n - f(x_n))^2 + \lambda \int (f''(t))^2 \, dt,
\]

(3.2.8)

where \( \lambda \) is a fixed smoothing parameter and \( f'' \) indicates the second derivative of \( f \). The first term measures the closeness to the data, while the second term penalizes curvature in the function, and \( \lambda \) establishes a tradeoff between the two. If \( \lambda = 0 \), \( f \) can be any function that interpolates the data, whereas if \( \lambda = \infty \), \( f \) is the simple least squares line fit, because no second derivative can be tolerated. For \( \lambda \in (0, \infty) \), the optimal function varies from very rough to very smooth. It can be shown that equation (3.2.8) has an explicit, finite-dimensional, unique minimizer which is a natural cubic spline, with knots at the unique values of the \( x_n \), \( n = 1, \ldots, N \). The penalty term translates to a penalty on the spline coefficients, which are shrunk some of the way toward a linear fit, thus avoiding over-parametrization (Hastie et al. 2009). Spline coefficients can be shrunk toward the linear fit in a more direct manner using the \( L_2 \) penalty (Fang et al. 2006). This approach is referred to as penalized splines. Penalized splines are a generalization of smoothing splines that allow more flexible choices of the spline model, the basis functions for that model, and the penalty (Ruppert et al. 2003). Let \( H(X) = [h_1(X), \ldots, h_{M+l}(X)]^T \) be the
vector of spline basis functions and let \( H(x_n)^T \) be the \( n \)th row of \( H(X) \). Then, a penalized spline is defined as
\[
\hat{f}(X) = \hat{\beta}^T H(X),
\] (3.2.9)
where \( \hat{\beta} \) is the minimizer of
\[
\sum_{n=1}^{N} \{y_n \beta^T H(x_n)\}^2 + \lambda \beta^T D \beta
\] (3.2.10)
for some symmetric positive semidefinite matrix \( D \) and scalar \( \lambda > 0 \) (Ruppert et al. 2003). The explicit form for the fitted coefficients is
\[
\hat{\beta} = (H(X)^T H(X) + \lambda D)^{-1} H(X)^T,
\] (3.2.11)
where \( D = \text{diag}(0_{p+1}, 1_k) \), and \( \lambda \) is a smoothing parameter controlling the smoothness of the fit (Ruppert et al. 2003). Note that if \( D \) is the second moment matrix of \( H(X) \), namely
\[
D = \int_a^b H'(X)[H'(X)]^T dX,
\] (3.2.12)
then the penalty in equation (3.2.8) is obtained. Because model (3.2.9) is linear in \( H(X) \), the penalized spline model can be expressed as
\[
\overline{y} = S_\lambda y,
\] (3.2.13)
where
\[
S_\lambda = B(X)(B(X)^T B(X) + \lambda D)^{-1} B(X)^T.
\] (3.2.14)
The matrix in equation (3.2.14) is the generalization of the hat matrix of a linear regression model, and is usually referred to as the smoother matrix. The smoother matrix \( S_\lambda \) is employed in the selection of the amount of smoothing. This is achieved by means of classical model selection ideas, including cross-validation (CV) and generalized cross-validation (GCV). The CV score is defined as
\[
CV(\lambda) = \sum_{n=1}^{N} \left( \frac{y_n - \overline{y}_n}{1 - S_{\lambda,nn}} \right),
\] (3.2.15)
where $S_{\lambda_{nn}}$ is the $(n,n)$ entry of $S_{\lambda}$. By computing the CV score over a set of values for $\lambda$, the optimal value of the smoothing parameter is selected according to

$$\tilde{\lambda} = \min_{\lambda} CV(\lambda). \quad (3.2.16)$$

The selection of $\tilde{\lambda}$ according to the GCV criteria is achieved by

$$\text{GCV}(\lambda) = \frac{\text{RSS}(\lambda)}{[1 - N^{-1}\text{tr}(S_{\lambda})]^2}, \quad (3.2.17)$$

for the CV score in equation (3.2.16). Here, $\text{RSS}(\lambda) = \sum_{n=1}^{N} (y_n - \tilde{f}(x_n;\lambda))^2$ and $\text{tr}(S_{\lambda})$ is regarded as a degree of freedom (or effective number of parameters) in the model fitting. Indeed, in parametric regression, the trace of the hat matrix identifies the number of fitted parameters and corresponds to the degrees of freedom (Ruppert et al. 2003). A similar interpretation is given to $\text{tr}(S_{\lambda})$ in nonparametric regression (Ruppert et al. 2003, Fang et al. 2006).

**Gaussian stochastic process models**

The use of Gaussian stochastic process models was introduced in the RSM field by Sacks, Welch, Mitchell & Wynn (1989). The authors considered the response to be modeled as a systematic departure from an assumed linear model. Consider the stochastic process $Y(X)$. Then, the GP model

$$y(X) = \sum_{i=1}^{p} \beta_i f_p(X) + Z(X), \quad (3.2.18)$$

is given by the sum of two terms, where the first term $\sum_{i=1}^{p} \beta_i f_p(X)$ is the assumed linear model and the second term $Z(X)$ is the systematic departure. Sometimes it might also be useful to eliminate the linear model and regard the entire response as a realization of the sole stochastic process (Sacks, Schiller & Welch 1989). Model (3.2.18) is deterministic. It is typically employed when computer experiments are carried out, namely when no measurement error occurs. In the absence of measurement error ($\sigma^2 = 0$), the estimated response surface interpolates the observations because the predictor $\tilde{y}(x_n) = y(x_n)$ at an experimental point $x_n$ has a MSPE of 0 (Sacks, Schiller & Welch 1989). $Z(X)$ in equation (3.2.18) is treated as a realization of a stochastic process (random function) in which the covariance structure of $Z$ is related to the smoothness of the response (Sacks, Schiller &
The covariance structure of the stochastic process \( Z \) is given by

\[
\text{cov}(Z(t), Z(u)) = \sigma_Z^2 R(t, u),
\]

where \( t \) and \( u \) are \( p \)-dimensional inputs \( t = (t_1, \ldots, t_p) \) and \( u = (u_1, \ldots, u_p) \), \( \sigma_Z^2 \) is a scale factor, and \( R \) is the correlation function. For a smooth response, a covariance function with some derivatives would be preferred, whereas an irregular response might call for a function with no derivatives (Sacks, Welch, Mitchell & Wynn 1989). Sacks, Welch, Mitchell & Wynn (1989) restrict their attention to the stationary family \( R(t, u) = R(t - u) \), and in particular to the correlations \( R(t, u) = \prod R_i(t_i - u_i) \), which are products of one-dimensional correlations. Of special interest are correlations of the form

\[
R(t, u) = \prod \exp\left(-\theta |t_i - u_i|^d\right),
\]

where \( 0 < d \leq 2 \). The parameter \( \theta \) defines the correlation structure of \( Z \), and its role is critical in prediction. Indeed, prediction is harder when \( \theta \) is large (small correlations between observations) than when \( \theta \) is small (large correlations between observations) (Sacks, Schiller & Welch 1989).

The predictor \( \hat{y}(X) \) is usually the best linear unbiased predictor (BLUP) of \( y(X) \) obtained by minimizing its mean squared prediction error (MSPE) subject to an ‘unbiasedness’ requirement. If enough data are available, either cross validation or maximum likelihood estimation (MLE) is used to estimate the parameters (\( \theta \) and \( p \)) of the family of correlations in equation (3.2.20). It is typically assumed that \( E[Z(X)] = 0 \) and the joint distributions of the \( Z(X) \)s are Gaussian (\( Z \) is a Gaussian process), and the maximum likelihood is employed. The likelihood is a function of the \( \beta \)s in the regression model, the process variance \( \sigma_Z^2 \), and the correlation parameters. Given the correlation parameters, the MLE of the \( \beta \)s and the MLE of \( \sigma_Z^2 \) are obtained, and the problem is reduced to minimize \((\det R)^{1/n} \sigma_Z^2\), which is a function of only the correlation parameters and the data (Sacks, Welch, Mitchell & Wynn 1989). Once the parameters have been specified, equation (3.2.20) can be used in equation (3.2.18) to provide predictions of \( y(x) \) at an untried experimental point \( x \) (Sacks, Schiller & Welch 1989). Because \( \theta \) is generally not available at the design stage (prior to data collection), Sacks, Welch, Mitchell & Wynn (1989) suggested a robustness study in
which an assumed $\theta_A$ is choosen, so as to give a design strategy that will perform well for a wide range of true (but unknown) values of $\theta$. The robust $\theta$ is also used for prediction if the data collected are inadequate for estimating $\theta$ (Sacks, Schiller & Welch 1989).

If the measurement error is important, a simple variation of equation (3.2.18) is used to introduce it and the following model is obtained,

$$Y(X) = \sum_{i=1}^{p} \beta_i f_p(X) + Z(X) + \epsilon(X). \quad (3.2.21)$$

Here, $\epsilon$ is the measurement error and $Z$ is a stochastic process, as before (Sacks, Schiller & Welch 1989). Linear prediction from equation (3.2.21) is known as kriging in the geostatistics and other spatial statistics literature (Cressie 1986, Ripley 1981). If model (3.2.21) is employed, measurement errors which are independent of both each other and $Z$ are incorporated if the covariance structure reflects

$$\text{var}(Y(t)) = \sigma_Z^2 + \sigma_\epsilon^2. \quad (3.2.22)$$

**Neural networks**

Neural networks (NNs) are powerful learning methods that have been developed separately in two different fields (statistics and artificial intelligence), based on essentially identical models (Hastie et al. 2009). They can be described as information processing paradigms inspired by the way in which the brain processes information. NNs consist of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. The NN learning process is inspired by learning in biological systems, which involves adjustments to the synaptic connections that exist between the neurons (Balkin & Lin 2000). Despite their seemingly complex structure, NNs are just nonlinear statistical models that are obtained in two steps (Hastie et al. 2009). First, linear combinations of the inputs are extracted as derived features, and then the target is modeled as a nonlinear function of these features (Hastie et al. 2009). Let $\alpha_m$, $m = 1, \ldots, M$, be $p$-dimensional vectors of unknown parameters, and let $Z_m$ be the derived features. The most widely used NN, namely the single hidden layer back-propagation network (also known as the single layer perceptron or ‘vanilla’ neural net), consists of a
single layer of hidden units. NNs with more than one hidden layer exist too. The hidden units are in the middle of the network, and serve to compute the derived features \( Z_m \)

\[
Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), \quad m = 1, \ldots, M, \tag{3.2.23}
\]

where \( \sigma(v) \), commonly referred to as the activation function, is usually the sigmoid \( \sigma(v) = 1/(1 + e^{-v}) \). The sigmoid is a smooth threshold function which is used to mimic the firing process of the neurons (a neuron is fired when the total signal passed to that unit exceeds a certain threshold). Assume that the response variable is a \( q \)-dimensional vector \( Y = (Y_1, \ldots, Y_q) \). Each output \( Y_j, j = 1, \ldots, q, \) is obtained as a function of linear combinations of \( Z = (Z_1, Z_2, \ldots, Z_M) \),

\[
Y_j = f_j(X) = g_j(\beta_{0j} + \beta_j^T Z), \quad j = 1, \ldots, q. \tag{3.2.24}
\]

For regression, the identity function \( g_j(Z) = \beta_{0j} + \beta_j^T Z \) is usually chosen. The final response is therefore given by

\[
Y_j = f_j(X) = g_j(\beta_{0j} + \beta_j^T \sigma(\alpha_{0m} + \alpha_m^T X)), \tag{3.2.25}
\]

The parameters of a NN are known as weights, and consist of the \( M(p + 1) \) weights \( \{\alpha_{0m}, \alpha_m\} \) with \( m = 1, \ldots, M \) and the \( K(M + 1) \) weights \( \{\beta_{0j}, \beta_j\} \), with \( j = 1, \ldots, q \). Let \( \theta \) denote the complete set of weights. Given a set of training data \((x_1, y_1), \ldots, (x_N, y_N)\), where \( x_n = (x_{n1}, \ldots, x_{np}) \) and \( y_n = (y_{n1}, \ldots, y_{nq}) \), with \( n = 1, \ldots, N \), the weights are estimated by minimizing the penalized error function

\[
R(\theta) + \lambda J(\theta), \tag{3.2.26}
\]

where \( R(\theta) \) is the sum of squared errors

\[
R(\theta) = \sum_{j=1}^{q} \sum_{n=1}^{N} (y_{nj} - f_j(x_n))^2, \tag{3.2.27}
\]

\( \lambda \geq 0 \) is a tuning parameter, and \( J(\theta) \) is a penalty term which is used to avoid overfitting. The value of the tuning parameter controls the roughness of the fitted function (larger
values of $\lambda$ tend to shrink the weights toward zero) and is typically estimated by cross-validation. The penalty term is the sum of squares of the weights $\theta$, and is known as weight decay (Hastie et al. 2009).

**Thin plate splines**

Consider the output $Y$ and the $p$-dimensional vector of inputs $X = (X_1,\ldots,X_p)$ with realizations $y = (y_1,\ldots,y_N)^T$ and $X = (x_1^T,\ldots,x_N^T)^T$. Then, the pairs $(x_n,y_n)$, $n = 1,\ldots,N$, are observed with $y_n \in \mathbb{R}$ and $x_n = (x_{n1},\ldots,x_{np}) \in \mathbb{R}^p$. A thin plate spline is a smooth $p$-dimensional surface of the form

$$f(X) = \beta_0 + \beta^T X + \sum_{n=1}^{N} \alpha_n h_n(X), \quad (3.2.28)$$

where $h_n(X) = \eta(||X - x_n||)$, $\eta(z) = z^2 \log z^2$. Equation (3.2.28) is the solution of the problem

$$\min_f \sum_{n=1}^{N} (y_n - f(x_n))^2 + \lambda J[f], \quad (3.2.29)$$

where $J$ is an appropriate penalty functional for stabilizing a function $f$ in $\mathbb{R}^p$. For example, for $p = 2$ we have that $X = (X_1, X_2)$ and

$$J[f] = \int \int_{\mathbb{R}^2} \left[ \left( \frac{\partial^2 f(X)}{\partial X_1^2} \right)^2 + 2 \left( \frac{\partial^2 f(X)}{\partial X_1 \partial X_2} \right)^2 + \left( \frac{\partial^2 f(X)}{\partial X_2^2} \right)^2 \right] dX_1 dX_2. \quad (3.2.30)$$

The parameter $\lambda$ in equation (3.2.28) controls the smoothness of the resulting surface. As $\lambda \to 0$, the solution approaches an interpolating function; as $\lambda \to \infty$, the solution approaches the least squares plane; and for intermediate values of $\lambda$, the solution can be represented as a linear expansion of basis functions, whose coefficients are obtained by a form of generalized ridge regression (Hastie et al. 2009). Radial basis function approximations have been shown to produce good fits to arbitrary contours of both deterministic and stochastic response functions (Jin, Chen & Simpson 2001).

**Multivariate adaptive regression splines**

Multivariate adaptive regression splines (MARS) is an adaptive procedure for regression which was introduced by Friedman (1991) and is well suited to high dimensional problems.
(Hastie et al. 2009). Let $X_i$ be the $i$th input variable, and let $\xi$ be the associated knot. A collection $\mathcal{C}$ of piecewise-linear basis functions

$$\mathcal{C} = \{(X_i - \xi)_+, (\xi - X_i)_+\}$$  \hspace{1cm} (3.2.31)

is used to build the model. Here, $\xi \in \{x_{i1}, x_{i2}, \ldots, x_{Ni}\}$, with $i = 1, \ldots, p$. The basis functions in equation (3.2.31) are given by

$$(X_i - \xi)_+ = \begin{cases} X_i - \xi, & \text{if } X_i > \xi, \\ 0, & \text{otherwise}; \end{cases} \quad (\xi - X_i)_+ = \begin{cases} \xi - X_i, & \text{if } X_i < \xi, \\ 0, & \text{otherwise}. \end{cases}$$

The model has the form

$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X),$$  \hspace{1cm} (3.2.32)

where each $h_m(X)$ is a function from $\mathcal{C}$, or a product of two or more such functions. Given a choice for the $h_m$, the coefficients $\beta_m$ are estimated by minimizing the residual sum of squares, that is, by standard linear regression (Hastie et al. 2009). The model-building strategy is like a forward stepwise linear regression which uses functions from the set $\mathcal{C}$ and their products, instead of the original inputs. After the forward basis selection procedure, a backward deletion is applied to avoid overfitting. The term whose removal causes the smallest increase in the residual squared error is deleted from the model at each stage, producing an estimated best model $\hat{f}_\lambda$ of each size $\lambda$ (number of terms). Generalized cross-validation (GCV) is used to estimate the optimal value of $\lambda$. The GCV criterion, defined as

$$GCV(\lambda) = \frac{\sum_{n=1}^{N} (y_n - \hat{f}_\lambda(x_n))^2}{(1 - M(\lambda)/N)^2},$$  \hspace{1cm} (3.2.33)

is minimized to select the final model. In equation (3.2.33), the value $M(\lambda)$ is the effective number of parameters in the model, and is given by $M(\lambda) = r + cK$, where $r$ is the number of linearly independent basis functions in the model, $K$ is the number of knots selected in the forward process, and $c$ is the penalty paid for selecting a knot in a piecewise linear regression. On the basis of the mathematical and simulation results, a value of $c = 3$ is typically used. However, if the model is restricted to be additive, a value of $c = 2$ is employed (Hastie et al. 2009). If multiple responses are investigated, the MARS procedure can be employed to build a set of models simultaneously, one for each output variable.
Each model uses the same set of basis functions but different coefficient values. The GCV and RSS values averaged across the responses are used to build the model. The aim is to select the best set of MARS terms across all responses (Milborrow 2009).

**Mixed model-based penalized spline regression**

Mixed models are typically used to study longitudinal data and the grouping structure in the data by handling the within-subject correlation in linear regression models by the employment of a random intercept. However, mixed models can also be used to smooth a scatterplot (the smoothed scatterplot corresponds to a mixed model fit). In particular, smoothing methods using basis functions with penalization can be expressed as mixed models. As a result, the methodology and software for mixed model analysis can be used in the field of smoothing (Wand 2003). Semiparametric regression models (that is, regression models which contain at least one function which is modeled nonparametrically) are considered here. In what follows, the Gaussian mixed model is introduced and the parameter estimation procedure is described briefly. The connection with scatterplot smoothing is then pointed out.

The generic form of a mixed model is typically given by

\[
E(y|u) = g(X\beta + Zu), \quad u \sim (0, G),
\]

(3.2.34)

where \(y\) is the response vector, \(X\) and \(Z\) are the design matrices, \(\beta\) is the fixed effects vector, \(u\) is the random effects vector, and \(g\) is a scalar link function. The formulation \(u \sim (0, G)\) stands for \(E(u) = 0\) and \(\text{Cov}(u) = G\) (Wand 2003, Robinson 1991). The Gaussian mixed model is

\[
y = X\beta + Zu + \epsilon, \quad \begin{bmatrix} u \\ \epsilon \end{bmatrix} \sim N\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} G & 0 \\ 0 & R \end{bmatrix} \right).
\]

(3.2.35)

To derive the ML estimates, the model (3.2.35) is rewritten as

\[
y = X\beta + \epsilon^*, \quad \epsilon^* = Zu + \epsilon,
\]

(3.2.36)
which is a linear model with correlated errors, since

\[ \text{Cov}(\epsilon^*) \equiv V = ZGZ^T + R. \] (3.2.37)

The log-likelihood of \( y \) under this model is

\[ l(\beta, V) = -\frac{1}{2} \left\{ N \log(2\pi) + \log|V| + (y - X\beta)^T V^{-1} (y - X\beta) \right\}, \] (3.2.38)

and the ML estimate of \((\beta, V)\) is the one that maximizes the right-hand side of this expression. Optimization is first performed over \( \beta \), so that

\[ \hat{\beta} = (X^T V^{-1} X)^{-1} X^T V^{-1} y \] (3.2.39)

for all \( V \). Substituting equation (3.2.39) into equation (3.2.38), the profile log-likelihood for \( V \) is obtained:

\[ l_P(V) = -\frac{1}{2} \left\{ \log|V| + (y - X\hat{\beta})^T V^{-1} (y - X\hat{\beta}) + N \log 2\pi \right\} \]

\[ = -\frac{1}{2} \left[ \log|V| + y^T V^{-1} (I - X(X^T V^{-1} X)^{-1} X^T) V^{-1} y \right] - \frac{N}{2} \log(2\pi). \] (3.2.40)

Maximizing equation (3.2.40) gives the ML estimates of the parameters in \( V \) (Ruppert et al. 2003).

Estimates of \( G \) and \( R \) can also be obtained via REML. This involves maximizing the likelihood of linear combinations of the elements of \( y \) that do not depend on \( \beta \) (Ruppert et al. 2003, Wand 2003). Because the derivation of this approach is complicated and beyond the scope of this section, the interested reader is directed to more specific papers, such as those of Searle et al. (1992), Patterson & Thompson (1971). The final optimization criterion is the restricted log-likelihood,

\[ l_R(V) = l_P(V) - \frac{1}{2} \log|X^T V^{-1} X|. \] (3.2.41)

The main advantage of REML over ML is that REML takes into account the degrees of freedom for the fixed effects in the model. For example, in the special case where a random
sample, $x_1, \ldots, x_n$, is collected from the $N(\mu, \sigma^2)$ distribution, we have

$$\hat{\sigma}^2_{ML} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})^2, \quad \hat{\sigma}^2_{REML} = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \bar{x})^2$$

(3.2.42)

where $\bar{x} = N^{-1} \sum_{n=1}^{N} x_n$. The $N - 1$ in the denominator of $\hat{\sigma}^2_{REML}$ accounts for the estimation of $\mu$ via $\bar{x}$. REML is expected to be more accurate than ML for small sample sizes, but for large samples there is little difference between the two approaches (Ruppert et al. 2003).

Once the estimates of $\beta$, $G$ and $R$ have been obtained, the random effects vector can be predicted using best linear prediction (BLP). This entails the determination of the value of $\hat{u}$ that minimizes

$$E[(\hat{u} - u)^T(\hat{u} - u)],$$

(3.2.43)

for which the solution

$$\hat{u} = E(u|y)$$

(3.2.44)

is the best predictor. For the Gaussian mixed model in equation (3.2.35),

$$\hat{u} = GZ^T V^{-1} (y - X\beta).$$

(3.2.45)

In practice, the random effects vector is estimated by substituting the estimates of $\beta$, $G$ and $R$ for the actual values in equation (3.2.44). For this reason, $\hat{u}$ in equation (3.2.45) is usually referred to as the estimated best predictor (Wand 2003).

The estimation of $\beta$ and $u$ can be performed without the Gaussian assumption using the idea of best linear unbiased prediction (BLUP). This consists of minimizing the prediction error

$$E[(s^T X\hat{\beta} + t^T Z\hat{u}) - (s^T X\beta + t^T Zu)]^2,$$

subject to the unbiasedness condition

$$E[(s^T X\hat{\beta} + t^T Z\hat{u})] = E[(s^T X\beta + t^T Zu)],$$

(3.2.47)

where $s$ and $t$ are arbitrary $N$-dimensional vectors. It has been shown (Robinson 1991, Hayes & Haslett 1999) that the solutions of equation (3.2.46) subject to the condition in
equation (3.2.47) are
\[
\widehat{\beta} = (X^T V^{-1} X)^{-1} X^T V^{-1} y, \tag{3.2.48}
\]
\[
\widehat{u} = G Z^T V^{-1} (y - X \widehat{\beta}). \tag{3.2.49}
\]

Note that the BLUP estimate \( \widehat{\beta} \) in equation (3.2.48) corresponds to both the ML estimator in equation (3.2.39) and the generalized least squares (GLS) estimator, namely the classical estimator of \( \beta \) for the linear model in equation (3.2.35). Furthermore, the BLUP estimate \( \widehat{u} \) is the best linear predictor (BLP) in equation (3.2.49), with \( \beta \) replaced by its BLUP estimate \( \widehat{\beta} \) (Ruppert et al. 2003). Another way of deriving BLUP solutions is via the Henderson’s justification, which makes the distributional assumptions
\[
y | u \sim N(X \beta + Z u, R), \quad u \sim N(0, G),
\]
and maximizes the likelihood of the \((y, u)\) over the unknowns \(\beta\) and \(u\). This leads to the criterion
\[
(y - X \beta - Z u)^T R^{-1} (y - X \beta - Z u) + u^T B u, \tag{3.2.50}
\]
which involves generalized least squares with a penalty term. The criterion in equation (3.2.50) is later used to make explicit the connection between mixed models and penalized spline.

The Gaussian mixed models in equation (3.2.35) can be obtained by fitting piecewise-linear models to the data. Given that the relationship between \(y_n\) and \(x_n\) is smooth and unknown,
\[
y_n = f(x_n) + \epsilon_n, \tag{3.2.51}
\]
linear models of the form
\[
y_n = \beta_0 + \beta_1 x_n + \sum_{l=1}^{L} u_l (x_n - \xi_l)_+ + \epsilon_n \tag{3.2.52}
\]
are considered. By imposing
\[
u_l \overset{\text{ind.}}{\sim} N(0, \sigma_u^2), \tag{3.2.53}
\]
and for $\sigma_u^2 < \infty$, $u_l$ are no longer allowed to range between $-\infty$ and $\infty$, but they must obey the laws of a normal probability distribution with a zero mean. This tends to shrink the $u_l$ and leads to a smooth fit (Wand 2003). The model in equation (3.2.52) and the restriction in equation (3.2.53) can be rewritten as the Gaussian mixed model (3.2.35), where

$$X = [1 \ x_n]_{1 \leq n \leq N}, \quad Z = [(x_n - \xi_l)_+]_{1 \leq n \leq N, 1 \leq l \leq L},$$

(3.2.54)

$$\beta = [\beta_0, \beta_1]^T, \quad u = [u_1, \ldots, u_L]^T, \quad G = \sigma_u^2 I, \quad R = \sigma^2 \epsilon I.$$ For given values of $\sigma_u^2$ and $\sigma^2$, the application of (RE)ML and BP to obtain $\hat{\beta}$ and $\hat{u}$ is equivalent to obtaining the penalized least squares problem

$$\begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} = \arg\min_{\beta, u} \left( \|y - X\beta - Zu\|^2 + \lambda^2 \|u\| \right),$$

(3.2.55)

where $\lambda = \sigma^2 / \sigma_u^2$ and $\|v\| = \sqrt{v^Tv}$ (Ruppert et al. 2003, Wand 2003). Let us first describe how the optimization problem in equation (3.2.55) is obtained, and then point out the equivalence with the best linear unbiased predictors (BLUPs) of a mixed model.

Let

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \quad \text{and} \quad u = \begin{bmatrix} u_1 \\ \vdots \\ u_L \end{bmatrix}$$

be the coefficients of the polynomial functions and truncated line functions in equation (3.2.52), respectively. The corresponding vectors are defined as

$$X = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \quad \text{and} \quad Z = \begin{bmatrix} (x_1 - \xi_1)_+ & \ldots & (x_1 - \xi_L)_+ \\ \vdots & \ddots & \vdots \\ (x_N - \xi_1)_+ & \ldots & (x_N - \xi_L)_+ \end{bmatrix}.$$ 

The scatterplot can be smoothed by setting

$$\hat{y} = X\hat{\beta} + Z\hat{u},$$
where \( \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} \) minimizes \( \| y - X\hat{\beta} - Zu \|^2 \). To avoid wiggly fit, constraints on the coefficients \( u \) are imposed. The simplest constraint to be implemented is \( \sum u^2 \leq B \). If we define the \((L + 2) \times (L + 2)\) matrix
\[
D = \begin{bmatrix}
0_{2x2} & 0_{2xL} \\
0_{Lx2} & I_{LxL}
\end{bmatrix}
\]
then the minimization problem can be written as
\[
\minimize \| y - X\hat{\beta} - Zu \|^2 \text{ subject to } u^T Du \leq B.
\]

Using a Lagrange multiplier argument, this is equivalent to choosing \( \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} \) to minimize
\[
\| y - X\hat{\beta} - Zu \|^2 + \lambda^2 \begin{bmatrix} \beta \\ u \end{bmatrix}^T D \begin{bmatrix} \beta \\ u \end{bmatrix}.
\]

This results in the penalized spline fitting criterion
\[
\| y - X\hat{\beta} - Zu \|^2 + \lambda^2 \begin{bmatrix} \beta \\ u \end{bmatrix}^T D \begin{bmatrix} \beta \\ u \end{bmatrix},
\]
that simplifies into
\[
\| y - X\hat{\beta} - Zu \|^2 + \lambda^2 \| u \|^2,
\] (3.2.56)
where \( \| u \| = \sqrt{u^T u} \) (Ruppert et al. 2003, Wand 2003).

The optimization criteria in equations (3.2.56) and (3.2.50) are made equivalent by multiplying the criterion in equation (3.2.50) by \( R \), which results in
\[
(y - X\hat{\beta} - Zu)^T (y - X\hat{\beta} - Zu) + RG^{-1}u^T u.
\] (3.2.57)

As a consequence, \( \lambda^2 = \sigma^2 / \sigma_u^2 \) in equation (3.2.56). Along with the piecewise-linear basis functions employed in equation (3.2.52), the truncated-power and radial basis functions in equations (3.2.4)-(3.2.6) can be used. Even though the fit does not change when different
basis functions are used, certain basis functions are to be preferred if the achievement of higher numerical stability and fit with greater accuracy (Ruppert et al. 2003).

3.3 Optimization methods

3.3.1 Sequential design: two-stage and one-stage approaches

Once a model has been fit to the data, optimization is carried out. Smith-Miles & Gupta (1992) proposed a NN-based two-stage approach for optimization based on sample points. In the first stage, the optimization function is approximated using a multilayer feed-forward neural network (MFNN), based on a sample of evaluated data points. In the second stage, the approximated function is optimized by using a feedback neural network to perform the gradient descent (Smith & Gupta 2001). The main advantage of this methodology is its applicability to a broad range of continuous optimization problems, since it makes no assumptions about the nature of the objective function. Indeed, objective functions that are non-quadratic, non-smooth, discontinuous, non-differentiable, or even difficult to represent mathematically, can be approximated using a MFNN, which can provide a smooth, continuous and differentiable representation of the objective function. Furthermore, different approximation and optimization methods can be used in place of neural networks (Smith-Miles & Gupta 1992). To reduce the computational burden of the optimization procedure, sequential designs have been developed (Sacks, Welch, Mitchell & Wynn 1989). The simplest sequential approach consists of taking the current optimal point as additional data. However, in this approach the additional design points tend to exploit only the area of the current existing data. As a result, the method tends to converge to a local maximum (or minimum), due to the lack of global information. On the other hand, if additional data are taken to be far from the existing data, it is difficult to obtain either more detailed information near the optimal point or a solution with a high level of precision due to insufficient information near the optimal point (Jones 2001). For these reasons, it is vital for any globally convergent method to have a feature that forces it to pay attention to parts of the space that are relatively unexplored, and, from time to time, to go back and sample these regions (Jones 2001). This can be achieved when the additional design points are dense (Jones 2001). The developed methods are usually
combined with the deterministic Gaussian process model described in section 3.2, because it allows a measure of the possible error in the predictor to be computed (Jones 2001). In the presence of observation noise (see, for example, the model in equation (3.2.21)), the standard approach is to construct a model using the noisy data and treat the observation noise variance $\sigma^2$ as an additional parameter to be optimized. Three main sequential design approaches exist and are described here. The first approach consists of minimizing a statistical lower bound of the form

$$\hat{y}(x) - ks(x),$$  \hspace{1cm} (3.3.1)

where $x \in \mathbb{R}^p$ is a design point in the experimental region $E$, $s$ is the root mean squared error or standard error, and $k$ is a constant value. As long as $k > 0$, there is some emphasis on searching in relatively unsampled regions where the standard error is high. In the literature, this idea has been pursued by Cox & John (1997). The second approach, one of the most popular in literature, consists in finding the point that maximizes the probability of improving the function beyond some target $T$. At any given point, the uncertainty about the functions value can be modeled by considering this value to be like the realization of a random variable $Y$ with mean $\hat{y}(x)$ and standard error $s(x)$. If the current best function value is $f_{\text{min}}$, then the target value for the improvement will be some number $T < f_{\text{min}}$. The probability of improving this is the probability that $Y(x) \leq T$. Assuming that the random variable is normally distributed, this probability is given by

$$\text{Prob improvement} = \Phi\left(\frac{T - \hat{y}(x)}{s(x)}\right),$$ \hspace{1cm} (3.3.2)

where $\Phi(\cdot)$ is the Normal cumulative distribution function (Jones 2001). The key advantage of using the probability of improvement is that, under certain mild assumptions, the additional design points will be dense. Intuitively, as the function is sampled more and more around the current best point, the standard error in this region will become small. As a result, the term

$$\frac{T - \hat{y}(x)}{s(x)}$$

will become extremely negative (since we usually have $T < \bar{y}(x)$), and hence, the probability of improvement given by equation (3.3.2) will be small. Eventually, the probability of
improvement around the current best point will become so small that we will be driven to search elsewhere where the standard error is higher. For example, to search for an improvement of at least 25%, the target is set to $T = f_{\min} - 0.25|f_{\min}|$ (Jones 2001). The difficulty here is that this method is extremely sensitive to the choice of the target $T$. If the desired improvement is too small, the search will be highly local and will only move on to search globally after a nearly exhaustive search around the current best point. On the other hand, if $T$ is set too high, the search will be excessively global, and the algorithm will be slow to fine-tune any promising solutions (Jones 2001). To overcome this sensitivity, Jones (2001) proposed the use of several values of the target $T$, corresponding to low, medium, and high levels of desired improvement. This leads to the selection of several search points in each iteration, so that both local and global search are performed. As a result, as soon as the global part of the search stumbles into the basin of convergence of the global minimum, the local part of the search will immediately begin to fine-tune the solution in the next iteration (Jones 2001). Another way of resolving the sensitivity of the criteria in equation (3.3.2) to the choice of $T$, is the use of the expected improvement criterion. The EI criterion involves computing how much is the expected improvement when sampling at a given point. Let $Y(x)$ be a random variable describing the uncertainty about the value of the function at a point $x$. $Y(x)$ is typically assumed to be normally distributed with a mean and variance given by the GP predictor, that is, by $\hat{\gamma}(x)$ and $s^2(x)$. If the current best function value is $f_{\min}$, then an improvement of $I$ will be achieved if $Y(x) = f_{\min} - I$. The likelihood of achieving this improvement is given by the normal density function

$$
\frac{1}{\sqrt{2\pi s(x)}} \exp \left[ -\frac{(f_{\min} - I - \hat{\gamma}(x))^2}{2s^2(x)} \right].
$$

(3.3.3)

The expected improvement is simply the expected value of the improvement found by integrating over this density:

$$
E(I) = \int_0^\infty I \left\{ \frac{1}{\sqrt{2\pi s(x)}} \exp \left[ -\frac{(f_{\min} - I - \hat{\gamma}(x))^2}{2s^2(x)} \right] \right\} dI.
$$

(3.3.4)

The appeal of the expected improvement (EI) approach is three-fold. First, it avoids the need to specify a desired improvement (e.g., the target $T$ of the previous section). Second, it provides a simple stopping rule, that is, stop when the expected improvement
from further searches is less than some small positive number. Third, it has been proven (Locatelli 1997) that under certain assumptions the iterates (additional design points) from this method are dense, which guarantees convergence to the global minimum. The drawback of EI is the long time required to find the global minimum. This is because the initial sample is highly deceptive, very small estimates of the standard error. As a result, only points that are close to the current best point have a high expected improvement are obtained. It requires a fairly exhaustive search around the initial best point for the algorithm to begin to search more globally (Jones 2001). This potential for deception is a fundamental weakness of all of the previously described methods. This weakness derives from the fact that they are two-stage methods. Indeed, during stage one, a model is fitted to the observed data and all of the relevant parameters are estimated. In the second stage, the estimated parameters are taken as to be correct and used to determine additional design points. When the initial sample is sparse, two-stage methods can be deceived and can give a highly misleading view of the function. Because the misleading view is taken to be correct in the second stage, the algorithm may either stop prematurely or become excessively local in its selection of design points. One-stage methods (Gutmann 2001, Jones 2001) escape this pitfall, and their purpose is to avoid estimating the parameters of the model based only on the observed sample (Jones 2001). In a one-step approach, a target value or goal for the objective function is assumed. Instead of minimizing the function, the purpose is to achieve a value \( f^* \) which is considered to be desirable (e.g., a goal might be set by benchmarking to competitive products). The hypothesis is that the goal \( f^* \) is achieved at a point \( x^* \). To evaluate the stated hypothesis, a conditional likelihood is computed, namely the likelihood of the observed data conditional on the assumption that the surface goes through the point \((x^*, f^*)\). If a deterministic GP model is used, the conditional likelihood is given by

\[
\frac{1}{(2\pi)^{n/2}(\sigma^2)^{n/2}|C|^{1/2}} \exp \left[ -\frac{(y - m)^T C^{-1} (y - m)}{\sigma^2} \right],
\]

where \( m \) and \( C \) are the conditional mean and correlation matrix

\[
m = 1\mu + r(f^* - \mu)
\]
In maximizing the conditional likelihood in equation (3.3.5), the model parameters are optimized and a measure of the credibility of the stated hypothesis is provided (Jones 2001). The key point is that the selection of the additional design point is not based on parameters which are obtained by fitting a surface to the observed data alone (Jones 2001). The one-step methods described have two main drawbacks. First, if the goal is unknown or unclear (which is usually the case) the procedure must be repeated using several search points, thus increasing its computational cost (Jones 2001). Second, if GP models are used, the methodology is computationally very intensive. Significant improvements have been achieved by Gutmann (2001), who employed TPL modeling technique in place of a GP model.

3.3.2 Multiresponse optimization

The optimization problem is more complex in the presence of multiple responses than in the single-response case (Khuri & Cornell 1996). Conditions which are optimal for one response may be far from optimal or even physically impractical for the other responses. As a result, it is rare for all of the response variables to achieve their respective optima in the same conditions (Khuri & Cornell 1996). The main difficulty is due to the fact that when two or more response variables are under investigation simultaneously, the meaning of optimum becomes unclear since there is no unique way to order multivariate values of a multiresponse function (Khuri & Cornell 1996). Four main approaches exist to deal with multiple responses in RSM and are described further below. Let \( Y = (Y_1, \ldots, Y_q)^T \) be a vector of \( q \) responses. A relatively straightforward strategy is to overlay the contour plots for each response and then pinpoint a region where conditions can be ‘near’ optimal for all of the responses (Montgomery 2009, Khuri & Cornell 1996). This involves first building an appropriate response surface model for each response (Montgomery 2009). Although simple, this procedure works well only with a limited number of inputs and response variables. Its main limitation derives from the difficulty to identify one optimal point in the experimental region (Khuri & Cornell 1996). A different approach, proposed by (Myers & Carter 1973), deals with the optimization of a dual response system consisting

\[
C = R - rr^T. \quad (3.3.7)
\]
of a primary response and a secondary response. The purpose is to find conditions on a set of input variables that maximize (or minimize) the primary response subject to the condition that the secondary response takes on some specified or desirable values. This results in a constrained optimization problem for the primary and secondary (constraint) response functions and the problem solution is similar to that of a ridge analysis (Khuri & Cornell 1996). A similar method is referred to as the lexicographic approach in the multi-objective optimization community. The basic idea is to assign different priorities to different objectives, and then focus on optimizing the objectives in their order of priority. Hence, the performance measures of two or more candidate experiments are first compared in terms of the highest-priority objective. If one candidate is significantly better than the other with respect to that objective, the former is chosen. Otherwise the performance measure of the two candidates is compared with respect to the second objective. The process is repeated until a clear winner is found or until all the criteria have been used (Freitas 2004). The third methodology used in the RSM field is known as desirability function approach and has been introduced by Harrington (1965). It consists of transforming each predicted response function \( \hat{Y}_j \), \( j = 1, \ldots, q \), into a desirability function \( h_j \), \( 0 \leq h_j \leq 1 \), such that \( h_j \) increases as the desirability of the corresponding property increases (Khuri & Cornell 1996). Different desirability functions have been developed and a high degree of flexibility can be achieved. Examples are the exponential-type transformations \( h_j = \exp(-|\hat{Y}_j|^s) \), where \( s \) is a user-specified positive number, and functions of the form

\[
  h = \begin{cases} 
  (\frac{\hat{Y} - A}{B - C})^s, & A \leq \hat{Y} \leq B \\
  (\frac{\hat{Y} - C}{B - C})^s, & B \leq \hat{Y} \leq C
  \end{cases}
\]  

with \( h \) being 0 if \( \hat{Y} > C \) or \( \hat{Y} > A \) (Khuri & Cornell 1996, Myers & Montgomery 1995). The individual desirability functions, one for each response, are then incorporated into a single function which gives the overall assessment of the desirability of the combined responses Khuri & Cornell (1996). The geometric mean \( H = \{h_1 \cdot h_2 \cdots h_q\}^{1/q} \) is used for that purpose. The aim is to choose the conditions on the design variables (factors levels) that maximize \( H \) (Myers & Montgomery 1995). The optimization process is relatively simple because the overall desirability is a well-behaved continuous function of the input variables and is performed using univariate optimization techniques (Khuri & Cornell 1996).
value of $H$ close to 1 implies that all responses are in a desirable range simultaneously (Myers & Montgomery 1995). The fourth methodology, proposed by Khuri & Conlon (1981), is referred to as generalized distance approach. It assumes that all of the response functions in a multiresponse system depend on the same set of input variables and can be represented by polynomial regression models of the same degree within the experimental region. Let $x$ be a point in the experimental region, and let $\hat{\phi}_j$ be the optimum value of $\hat{Y}_j(x)$, where $j = 1,\ldots,q$, optimized individually over the experimental region, and let $\hat{\phi} = (\hat{\phi}_1,\ldots,\hat{\phi}_q)^T$. If the individual optima are attained at the same set $x$ of operating conditions (levels), then an ‘ideal’ optimum is said to be achieved. However, such an ideal optimum rarely exists and compromising conditions on the input variables that are somewhat favorable to all responses are searched. The purpose is therefore to select a near optimum for each of the $q$ predicted response functions. The deviation of the compromising conditions from the ideal optimum is formulated in terms of distance function between the predicted responses $\hat{Y}(x)$ and the vector of individual optima $\hat{\phi}$.

Various distance functions have been proposed and all measure the distance of $\hat{Y}(x)$ from $\hat{\phi}$ after is has been properly scaled. An example is the weighted distance

$$
\rho[\hat{Y}(x) - \hat{\phi}] = [(\hat{Y}(x) - \hat{\phi})^T \{Var[\hat{Y}(x)]\}^{-1} (\hat{Y}(x) - \hat{\phi})]^{1/2}.
$$

(3.3.9)

If $x_0$ is the point in the experimental region at which $\rho[\hat{Y}(x - \hat{\phi})]$ attains its absolute minimum, and if $m_0$ is the value of this minimum, then at $x_0$ the experimental conditions can be described as being near optimal for each of the $q$ response functions. The smaller the value of $m_0$ the closer these conditions are to representing an ‘ideal’ optimum (Khuri & Cornell 1996). An extremely similar concept has been developed in the multi-objective optimization context. The approach, referred to as the method of distance functions, consists in deriving a single objective function from multiple objectives

$$
Z = \left[ \sum_{j=1}^{q} |f_j(x) - y_j^*|^r \right]^{1/r}
$$

(3.3.10)

for $1 \leq r < \infty$. Here, $x$ is a point in the experimental region $E$ and $y^* = (y_1^*,\ldots,y_q^*)$ is a user-defined demand-level vector. For $r = 2$ the Euclidean distance is obtained (Srinivas & Deb 1994). Two additional methodologies commonly employed in the multi-objective
optimization field are the weighted-formula approach and the Pareto approach. The first consists in assigning a numerical weight \( w_i \) to each normalized objective function and then obtaining a single value by adding

\[
\min f = w_1 f_1'(x) + w_2 f_2'(x) + \ldots + w_q f_q'(x),
\]

(3.3.11)

or multiplying

\[
\min f = f_1'(x)^{w_1} + f_2'(x)^{w_2} + \ldots + f_q'(x)^{w_q},
\]

(3.3.12)

the values of the weighted objective functions. Here, \( f_j'(x) \) is the normalized objective function \( f_j(x) \) and \( \sum w_j = 1 \) (Freitas 2004). As opposed to the described methods, the Pareto approach uses a multi-objective algorithm to solve the original multi-objective problem. It doesn’t transform a multi-objective problem into a single-objective problem and then solve it by using a single-objective search method. This approach is based on the concept of Pareto dominance. If all of the objective functions \( f_j \) are for minimization, a feasible solution \( x_1 \) is said to dominate another feasible solution \( x_2 \), if and only if \( f_j(x_1) \leq f_j(x_2) \) for \( j = 1, \ldots, q \) and \( f_j(x_1) < f_j(x_2) \) for at least one objective function \( j \) (Konak et al. 2006). In other words, a solution \( x_1 \) is said to dominate a solution \( x_2 \) if and only if \( x_1 \) is strictly better than \( x_2 \) with respect to at least one of the objective functions being optimized and \( x_1 \) is not worse than \( x_2 \) with respect to all the objective function being optimized (Freitas 2004). All solutions that are not dominated by any other decision vector in the solution space are called non-dominated or Pareto-optimal solutions (Mansouri 2005). These are solutions for which no objective can be improved without detracting from at least one other objective (Mansouri 2005). This means that a reasonable solution to a multi-objective problem is to investigate a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by any other solution. The set of all feasible non-dominated solutions in \( E \) is referred to as the Pareto optimal set, and for a given Pareto optimal set, the corresponding objective function values in the objective space are called the Pareto front (Konak et al. 2006). Because of the conflicting nature of the objective functions it is almost impossible to find a solution that simultaneously optimize each objective function (Konak et al. 2006). That’s way optimization with respect to a single objective often results in unacceptable results with respect to the other objectives.
Heuristic methods

When the task is to optimize multiple highly nonlinear (or multimodal) objective functions, the standard approach is to adopt heuristic search procedures (Myers et al. 2004). Heuristic methods can find good (not necessarily optimal) solutions to difficult combinatorial optimization problems (Feo & Resende 1995). The most promising techniques include simulated annealing (Kirkpatrick 1984), tabu search (Glover 1989, 1990) and genetic algorithms (Goldberg 1989). Simulated annealing and tabu search refine a single candidate solution and are notoriously weak in dealing with local optima. In contrast, genetic algorithms (GAs) cope with local optima evolving several candidate solutions (population) simultaneously (Paterlini & Krink 2006). Because multiple individuals can search for multiple solutions in parallel, genetic algorithms are particularly suited for the multi-objective nature of the investigated systems (Fonseca & Fleming 2005). Introduced by Holland (1975), GA is a stochastic search heuristic inspired by Darwinian evolution and genetics. The main idea is to create a population of candidate solutions to an optimization problem, which is iteratively refined by alteration and selection of good solutions for the next iteration. Candidate solutions are selected according to the so-called fitness function, which evaluates their quality with respect to the optimization problem. In case of GAs, the alteration consists of mutation to randomly explore the local neighborhood of existing solutions and crossover to recombine information of different candidate solutions (Paterlini & Krink 2006). However, to locate a near-optimal solution, EAs require a computer model (fitness function) to be optimized. If the fitness function is computationally expensive or unknown it can be approximated by a surrogate model and the connection with DACE becomes evident. For a comprehensive survey about this topic see Jin (2005). Surrogate models are constructed from a limited number of data points that hopefully mimics the entire search landscape (Zhou et al. 2007). These data points are usually obtained during one or more generations of a classical evolutionary search. Subsequently, the surrogate model is updated online, based on the new data points generated as the search evolves (Zhou et al. 2007). Polynomial model, neural network (NN), radial basis function (RBF), and Gaussian process (GP) model are among some of the most prominent and commonly used modeling techniques. Other heuristics belonging to the class of population-based procedures are Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO).
PSO and ACO are swarm intelligence search heuristics (Tripathi et al. 2007). Introduced by Kennedy & Eberhart (1995) PSO mimics the social behavior of a flock of birds in order to guide swarms of particles towards the most promising regions of the search space (Tripathi et al. 2007). ACO was introduced by Dorigo (1992) and is inspired by the foraging behavior of ant colonies. Each ant evaluates its path (solution) according to a fitness function and moves in the search space thanks to the stigmergy, a form of indirect communication mediated by modification of the environment (Dorigo 1992).

3.4 Towards a customized approach

NPRSM can be seen as a process encompassing an initial experimental design and an optimization procedure. Optimization typically consists of two stages. During the first stage a response surface is fitted to the observed data and any required parameters are estimated. In the second stage the fitted surface is used to identify additional search points that are employed to augment the initial design. This two-stage approach has a potential pitfall deriving from the fact that, if the initial sample is sparse, the fitted response surface can provide a highly misleading view of the true unknown function. Because this misleading view is taken as correct during the second stage, the optimization procedure may stop prematurely or become excessively local (Jones 2001). One-stage methods have been developed to escape this pitfall. They skip the initial step of fitting a surface to the observed data and aim at achieving a target value which is considered to be desirable. One-stage approaches naturally combines with the distance-based approaches that have been developed by both the statistical and computational intelligence communities to optimize multiple responses. The purpose of distance-based approaches is to minimize the distance between the observed responses and an ideal optimum, the latter being a user-defined demand-level vector (computational intelligence community) or a vector of values obtained individually optimizing a polynomial model (statistical community). The less developed area in RSM and NPRSM is probably the analysis (experimental design, modeling and optimization) of multiresponse systems. It has been lagging in comparison with single-response counterpart (Ghosh 1990) because of several factors including the higher complexity and the lack of software (Khuri & Cornell 1996, Ghosh 1990).
4 focuses on multiresponse optimization and sequential design. A one-stage distribution-free approach is proposed to deal with high dimensional multiresponse systems. Chapters 5 and 6 concentrate on modeling of multiresponse systems providing a literature review and a meta-learning framework to generate a knowledge system for model selection purposes.
Chapter 4

The evolutionary model-based multiresponse experimental design

In this chapter a one-stage distribution-free methodology, named Evolutionary Model-based Multiresponse Approach (EMMA), is proposed. It combines NPRSM techniques, multi-objective optimization and computational intelligence methods to optimize a high dimensional system with multiple responses using a moderate number of experimental runs. The focus of this chapter is on optimization and sequential design. Modeling will be analyzed deeply in the second part of the thesis (chapters 5 and 6), with the aim to develop a framework for model selection in multiresponse environments. The chapter is organized as follows. Section 4.1 provides a detailed description of EMMA. Empirical results suggesting the use of a particle swarm optimization (PSO) heuristic are shown along with a description of the employed PSO algorithm. Section 4.2 investigates the performance of the proposed approach on a set of standard test functions and demonstrate the efficacy of EMMA to converge to the true optimum. In section 4.3, EMMA is applied to the chemical problem which motivated this study and the results obtained are presented. Finally, the advantages and disadvantages of the proposed approach are discussed in section 4.4, along with open questions and possible improvements.
4.1 The Evolutionary Model-based Multiresponse Approach (EMMA)

Critical is the development of a procedure which is capable of optimizing the multiple responses of a high dimensional system characterized by a wide experimental region and constraints on the input variables. Furthermore, the limited availability of the resources (e.g., budget, time, material) must be taken into consideration. The optimization is carried out over an experimental region $E$. Two different types of factors can be employed to define $E$, namely variables that may be easily set anywhere in a specified range and variables which are restricted to a finite set of values (Welch 1984). The latter includes qualitative variables as well as quantitative variables limited to a few levels for experimental convenience (Mitchell 1974, Welch 1984). Due to the nature of the inputs of the motivating problem, attention here is restricted to quantitative variables. A finite number of levels is considered for each factor, for various reasons. As was stated by Mitchell (1974), this approach has many advantages

1. the search procedure can be programmed easily,
2. it is easy to exclude points that are not experimentally feasible or desirable,
3. it does not matter whether the variables are qualitative or quantitative, and
4. the number of different levels of each factor can be kept reasonably low.

Point 4 can be quite important to the experimenter. Indeed, achieving a given level of a quantitative factor requires a considerable amount of time, preparation, and precision of the laboratory tools.

Due to the limit on the available resources, the aim is to achieve the optimum with the smallest possible number of experimental points investigated. The presence of multiple responses and their conflicting nature (typically, some responses are to be maximized and some others are to be minimized) constitute the main challenges of the optimization problem. Indeed, it is not feasible to optimize all of the response variables simultaneously, and a compromise solution must be identified. Because optimizing the individual objectives separately would result in an unacceptable solution, a procedure which is capable of
investigating all of the responses simultaneously is required. A distance-based method has been employed for that purpose. Distance-based optimization (see section 3.3.2) has been chosen because it combines naturally with the one-stage approach described in section 3.3.1. Figure 4.1 provides a global overview of the methodology developed. EMMA consists of defining an experimental region and selecting an initial design, and iteratively augmenting it with additional design points to optimize the responses investigated. The additional design points are selected sequentially according to the following procedure:

1. implement the designed experiments and measure the responses;

2. fit a model (predictor) to the collected data;

3. identify the target, namely a vector of values obtained by individually optimizing the predictor responses; and

Figure 4.1: EMMA consists of two elements: an initial design and a four-step iterative procedure. The four-step procedure aims to identify additional design points which are used to augment the initial design. At each iteration the designed experiments are carried out, and a model is fitted to the collected data and used to select a target. Finally, an optimization technique is employed to minimize the distance between the measured responses and the selected target.
4. select additional design points by minimizing the distance between the observed responses and the identified target. In particular, let \( y_n^T = (y_{n1}, \ldots, y_{nq}) \) be the \( q \)-dimensional vector of measured responses for the \( n \)th experimental point and let \( y^* = (y^*_1, \ldots, y^*_q) \) be the \( q \)-dimensional target vector. Then, the minimization criterion for the \( n \)th observation is

\[
\left[ \sum_{j=1}^{q} |y_{nj} - y^*_j|^r \right]^{1/r} \quad j = 1, \ldots, q \text{ and } n = 1, \ldots, N \tag{4.1.1}
\]

where \( 1 \leq r < \infty \). Because the response variables might have different relevance, a weight vector is introduced and used to define the weighted optimization criterion

\[
\left[ \sum_{j=1}^{q} |u_{nj} - u^*_j|^r \right]^{1/r} \quad j = 1, \ldots, q \text{ and } n = 1, \ldots, N \tag{4.1.2}
\]

where \( u_{nj} = w_j \cdot y_{nj}, u^*_j = w_j \cdot y^*_j \), and \( w \) is the \( q \)-dimensional vector of weights.

Steps 1 to 4 are iterated until a predefined maximum number of experiments is reached. A number of experimental design, modeling and optimization techniques can be included in the methodology proposed. To avoid the practitioner’s influence, a simple random sample from the experimental region has been used as initial design. However, other design strategies can be employed, including those described in section 3.1. In step 2, a variety of other modeling techniques can also be used in place of the highly adaptive models discussed in section 3.2. The major approaches which are applicable when multiple responses are investigated will be described in chapter 5 and will be studied in chapter 6. Among the optimization methods that can be employed in step 4, population-based heuristics (e.g., genetic algorithms, particle swarm optimization algorithms, and ant colony optimization algorithms) are of particular interest because of their ability to cope with local optima by evolving several candidate solutions simultaneously (Paterlini & Krink 2006). Within the proposed approach, the population of candidate solutions corresponds to an experimental design \( D = \{x^T_1, \ldots, x^T_N\} \), and each element of the population \( x^T_n = (x_{n1}, \ldots, x_{np}) \), with \( n = 1, \ldots, N \), defines an experimental point (Forlin et al. 2008). As the population of candidates evolves towards better solutions, new experimental designs are selected. The MARS model, described in section 3.2, and a time-variant Particle
Swarm Optimization algorithm (PSO) are used here as the modeling and optimization techniques, respectively. MARS has been chosen because of its many attractive features, including flexibility, the ability to cope with high-dimensional input and output spaces, the capacity to model complex nonlinear relationships, and clear interpretability (Hastie et al. 2009, Lee & Chen 2005). The PSO has been selected on the basis of empirical results comparing the performances of a genetic algorithm (GA), a PSO and a neural network-based approach (NN). The simulation study which has been conducted to select the most promising optimization algorithm is presented below, and the empirical results are discussed as well. Subsequently, the PSO algorithm and the implemented time-varying PSO heuristic are described.

4.1.1 Selection of the optimization heuristic

To select the most promising optimization algorithm, a simulation study has been conducted and three methods have been compared. The methods investigated are a simple GA, a simple PSO and a NN-based procedure. They all augment the initial design by identifying additional experimental points according to different mechanisms. In particular, GA iteratively combines, alters and selects good solutions. In PSO, the members of the population (particles) are moved towards better solutions by means of social-inspired concepts, on the basis of which, evaluation, comparison, and imitation of others, as well as learning from experience, allow humans to adapt to their environment and determine optimal patterns of behavior, attitudes, and suchlike (del Valle et al. 2008). The NN-based approach augments the initial design by selecting the best predicted responses obtained using a NN previously fitted to the collected data. The purpose of the study is to generate a simulated problem with characteristics similar to those of the chemical problem that motivated this work (see Chapter 2). Data collected during the first stage of the experimentation have been used for this purpose. In particular, 88 sol-gel coatings have been obtained and their quality has been measured in terms of spot circularity (Cir), intensity (Int), background (Back) and homogeneity (Hom). Information about the stability (Stab) of the sol-gel mixture is also collected. 40 of the 88 coatings were obtained during the preliminary study, which was aimed at defining suitable ranges for the input variables (see section 2.2 and Table 2.1). The remaining 48 coatings are experimental points, selected
by means of a simple random sample from the experimental region, which constitute the initial design. A multiresponse MARS model has been fitted to the collected data and used to predict the responses at the additional experimental points selected by the optimization algorithms. The unknown target has been guessed and the 88 observed response values have been taken into consideration in doing so. In this simulation, the role of the MARS model is not to define the target but to provide an approximation of the unknown functional relationship between the input and output variables. The algorithms investigated aim to minimize the distance between the simulated responses (MARS predicted values) and the user-defined target. The initial design is augmented using 15 additional experimental points for each iteration of the procedure. A total of 7 iterations (or time instants) has been employed. The number of additional experimental points and the number of iterations have been defined according to the maximum number of experiments that can be carried out for the motivating chemical problem given the resources available (see section 2.2). 100 simulation runs have been conducted, and the average performances are reported in Figure 4.2. The performances are measured using equation (4.1.2), which evaluates the distance between the simulated response values and the user-defined target. Because it identifies the solutions which are closer to the ideal optimum (target), the PSO-based approach has been chosen.

4.1.2 Implementation of the PSO

PSO is a stochastic optimization technique, developed by Kennedy & Eberhart (1995). It is inspired by the social behavior of bird flocking and fish schooling, and pertain to the area of swarm intelligence. PSO utilizes a population of particles that fly through the problem hyperspace with given velocities. Both the particle best and the swarm best are derived according to a user defined fitness function, namely the function to be optimized. They are employed to guide the movement of each particle, which naturally evolves to an optimal or near-optimal solution (del Valle et al. 2008). Let \( x_n = (x_{n1}, x_{n2}, \ldots, x_{np}) \) be the position of the \( n \)th particle in a \( p \)-dimensional search space. The velocity of the \( n \)th particle is given by the \( p \)-dimensional vector \( v_n = (v_{n1}, v_{n2}, \ldots, v_{np}) \), whereas \( pb_n = (pb_{n1}, pb_{n2}, \ldots, pb_{np}) \) stores the position corresponding to the particle’s best individual performance. The experience of the whole swarm is captured in the index \( g \), which corresponds to the particle with the
Figure 4.2: The performances of three optimization approaches are compared with a view to selecting the most promising method for the motivating problem. The graph depicts the performances, measured in terms of the distance between the target and the simulated responses, averaged across 100 simulation runs. 7 iterations are investigated. The initial design (iteration 1) is sequentially augmented from iterations 2 to 7. At each iteration, 15 additional design points are selected. The GA, NN-based and PSO performances are described by the dashed line, the dotted-dashed line and the solid line, respectively.

best overall performance in the swarm. The movement of the particle towards the optimal solution is governed by the following update equations:

\[ v_{ni} = w v_{ni} + c_1 r_1 (p b_{ni} - x_{ni}) + c_2 r_2 (p b_{gi} - x_{ni}) \] (4.1.3)

\[ x_{ni} = x_{ni} + v_{ni} \] (4.1.4)

where \( i = 1, \ldots, p \), and \( w, c_1, c_2 \geq 0 \). The parameter \( w \) is the inertia weight, and \( c_1 \) and \( c_2 \) are the acceleration coefficients. Finally, \( r_1 \) and \( r_2 \) are random numbers, generated uniformly in the range \([0, 1]\), and are responsible for providing randomness to the flight of the swarm. The inertia weight parameter \( w \) controls the influence of the previous velocity on the present velocity. Therefore, high values of \( w \) help in the global search.
for the optimal solution, whereas low values of $w$ help in the local search around the current search area. The term $c_1 r_1 (p_{bni} - x_{ni})$ in equation (4.1.3) is called the cognition term, while the term $c_2 r_2 (p_{bgi} - x_{ni})$ is called the social term. The cognition term takes into account only the particle’s individual experience, whereas the social term indicates the interaction between the particles. The acceleration coefficients $c_1$ and $c_2$ allow the particle to tune the cognition and social terms respectively in the velocity update equation (4.1.3). As a result, high values of $c_1$ ensure a larger deviation of the particle within the search space (exploitation), whereas high values of $c_2$ encourage convergence to the present global best (exploitation) (Tripathi et al. 2007). PSO has many advantages over other optimization algorithms. These include the fact that PSO is not affected much by the size and nonlinearity of the problem, and can converge to the optimal solution in many problems where most analytical methods fail to converge. Moreover, PSO has various benefits over other similar optimization techniques such as GA. First of all, PSO is easier to implement and is characterized by having fewer parameters to adjust. Second, in PSO every particle remembers both its own previous best value and the global best; it therefore has a more effective memory capability than the GA. Third, PSO is more efficient in maintaining the diversity of the swarm, because all of the particles use the information related to the most successful particle in order to improve themselves, whereas in GA, the worse solutions are discarded and only the good ones are saved; therefore, in GA the population evolves around a subset of the best individuals (del Valle et al. 2008). Furthermore, PSO can handle constraints simply. Two different approaches to this exist in the literature. One approach is to include the constraints in the fitness function using penalty functions, while the second approach deals with the constraints and fitness separately. The main advantage of the second approach is that no additional parameters are introduced in the PSO algorithm, and there is also no limit to the number or format of the constraints. The basic PSO equations for velocity and position updating remain unchanged. After the new positions have been determined for all the particles, each solution is checked to determine whether it belongs to the feasible space or not. If the feasibility conditions are not met, one of the following actions can be taken: the particle is reset to the previous position, or the particle is reset to its personal best, or the nonfeasible solution is kept but the personal best is not updated, or the particle is rerandomized (del
Valle et al. 2008). In EMMA, the particle closest to the selected unfeasible particle is considered.

To handle optimization problems of different characteristics effectively and to avoid the premature convergence of the algorithm, time varying parameters and a mutation operator have been introduced into the PSO heuristic (Carlisle & Dozier 2000, 2002, Ying et al. 2006, Tripathi et al. 2007). The same strategy is adopted here. Indeed, the time varying parameters and the mutation operator affect the search process by making it either more global or more local. A global search generates a broad exploration of the whole space, while a local search allows a detailed study of a small area (exploitation). By initially encouraging a global search and subsequently restricting the search to the most promising local area, a better compromise between exploration and exploitation is achieved (in theory). The parameters employed by Tripathi et al. (2007) are here considered and described further below. The value of the inertia weight $w$ is allowed to decrease linearly with the iteration from $w_1$ to $w_2$

$$w_t = (w_1 - w_2) \frac{t}{\max_t} + w_2,$$

(4.1.5)

where $\max_t$ is the maximum number of iterations and $t$ is the iteration number. The acceleration coefficient $c_1$ has been allowed to decrease from its initial value of $c_{1i}$ to $c_{1f}$, while $c_2$ has been allowed to increase from $c_{2i}$ to $c_{2f}$. The following equations are used

$$c_{1t} = (c_{1f} - c_{1i}) \frac{t}{\max_t} + c_{1i},$$

(4.1.6)

$$c_{2t} = (c_{2f} - c_{2i}) \frac{t}{\max_t} + c_{2i}.$$

(4.1.7)

Finally, the mutation operator is as follows. Given a particle, a randomly chosen factor level $x_{ni}$ (one of the coordinates of the particle) is mutated using

$$x_{ni}' = \begin{cases} 
  x_{ni} + \Delta(t, UB - x_{ni}) & \text{if } flip = 0, \\
  x_{ni} - \Delta(t, x_{ni} - LB) & \text{if } flip = 1.
\end{cases}$$

(4.1.8)
Here, *flip* denotes the random event of returning 0 or 1. *UB* denotes the upper limit of the variable $x_{ni}$, and *LB* denotes the lower limit of $x_{ni}$. The function $\Delta$ is defined as

$$\Delta(t, x) = x \left(1 - r \left(1 - \frac{1}{\max_t} \right)^b \right), \quad (4.1.9)$$

where $r$ is a random number generated in the range $[0, 1]$, $\max_t$ is the maximum number of iterations and $t$ is the iteration number. The parameter $b$ determines the degree of dependence of the mutation on the iteration number. As a result, the updating equations employed are

$$v_{ni} = w_t v_{ni} + c_{1i} r_1 (p b_{ni} - x_{ni}) + c_{2i} r_2 (p b_{gj} - x_{ni}) \quad (4.1.10)$$

$$x_{ni} = x_{ni} + v_{ni}. \quad (4.1.11)$$

The PSO parameter values used in Tripathi et al. (2007) have been employed. They are reported in Table 4.1.

### 4.2 EMMA: the efficacy in the optimization process

Test functions are commonly used in the literature to evaluate the performance of optimization algorithms. A classic example is the DeJong test suite (De Jong 1975), a set of five continuous functions that isolate the common difficulties encountered by many optimization problems (Smith & Gupta 2001). Because the DeJong test suite is considered by many to be the minimum standard for performance comparisons of optimization algorithms, it is used here to test the capacity of the developed approach to converge to the
true optimum. In what follows, the test functions are described and their two-dimensional graphical representation is provided. The first benchmark function, the sphere function, is continuous, convex and unimodal. It is defined as

\[ f_1(X) = \sum_{i=1}^{p} X_i^2, \]  

(4.2.1)

where the test area is usually restricted to the hypercube \(-5.12 \leq X_i \leq 5.12\), with \(i = 1, \ldots, p\). The global minimum \(f_1(X) = 0\) is achieved at \(X_i = 0\) (De Jong 1975). The second test function is the Rosenbrock’s saddle, also known as Rosenbrock’s valley or the banana function. It is characterized by difficult convergence to the global optimum, which lies inside a long, narrow, parabolic shaped flat valley (De Jong 1975). Its general definition is

\[ f_2(X) = \sum_{i=1}^{p-1} \left[ 100(X_{i+1} - X_i^2)^2 + (1 - X_i)^2 \right] \]  

(4.2.2)

and the test area is typically restricted to the hypercube \(-2.048 \leq X_i \leq 2.048\), with \(i = 1, \ldots, p\). The global minimum \(f_2(X) = 0\) is achieved at \(X_i = 1\). The third test benchmark, the step function, consists of many flat plateaus with steep, uniform ridges, and poses problems for algorithms which require gradient information to determine a search.

Figure 4.3: The sphere function: the first benchmark function of the DeJong test suite.
Figure 4.4: The Rosenbrock’s valley: the second benchmark function of the DeJong test suite.

direction (Smith & Gupta 2001). It is defined as

\[ f_3(X) = \sum_{i=1}^{p} \text{int}(X_i), \]  

(4.2.3)

with a test area which is usually restricted to the hypercube \(-5.12 \leq X_i \leq 5.12\), with \(i = 1, \ldots, p\). The global minimum \(f_3(X) = 0\) is achieved at \(X_i = [-5.12, -5]\). The fourth test benchmark, the quartic function, is a simple unimodal function, but is difficult to optimize because of the presence of high noise levels. It is formulated as

\[ f_4(X) = \sum_{i=1}^{p} iX_i^4 + \epsilon_i, \]  

(4.2.4)

with \(\epsilon_i \sim \mathcal{N}(0, 1)\) and a test area which is restricted to the hypercube \(-1.28 \leq X_i \leq 1.28\), with \(i = 1, 2\). The global minimum \(f_4(X) = 0\) is achieved at \(X_i = 0\). The last test function, Shekel’s foxholes, contains 25 foxholes of varying depths, surrounded by a flat surface. Many algorithms will get stuck in the first foxhole they fall into (Smith & Gupta 2001). The function is defined as

\[ f_5(X) = \frac{1}{0.002 + \sum_{j=1}^{25} \frac{1}{1 + \sum_{i=1}^{p} (X_i - a_{ij})^2}}, \]  

(4.2.5)
Figure 4.5: The step function: the third benchmark function of the DeJong test suite.

Figure 4.6: The quartic function: the fourth benchmark function of the DeJong test suite. The figure depicts the function generated without the noise term.
where

\[ a_{ij} = \begin{pmatrix}
-32 & -16 & 0 & 16 & 32 & \ldots & -32 & -16 & 0 & 16 & 32 \\
\end{pmatrix} \]

The test area is usually restricted to the hypercube \(-65.536 \leq X_i \leq 65.536\), with \(i = 1, \ldots, n\) (Smith & Gupta 2001). The global minimum \(f_5(X) = 1\) is achieved at \(X_i = -32\). A multimodal benchmark function very popular in optimization, the Ackley function, is considered too. It is defined as

\[ f_6(X) = -20 \cdot \exp \left( 0.2 \sqrt{\frac{1}{p} \sum_{i=1}^{p} X_i^2} \right) - \exp \left( \frac{1}{s} \sum_{i=1}^{s} \cos \pi \cdot X_i \right) + 20 + \exp(1) \quad (4.2.6) \]

with a test area restricted to the hypercube \(-15 \leq X_i \leq 30, i = 1, \ldots, n\). The function is characterized by several local minima and one global minimum \(f_6(X) = 0\) at \(X_i = 0\).

### 4.2.1 Empirical results

EMMA is tested on a 3-dimensional sphere function, a 2-dimensional Rosenbrock’s saddle, a 5-dimensional step function, a 4-dimensional quartic function, a 2-dimensional Shekel’s...
foxholes, and a 2-dimensional Ackley function. The dimensionality of the DeJong test functions has been chosen according to the work of Smith & Gupta (2001). A swarm of 58 particles is employed at the first iteration (initial design), and the number of particles is subsequently reduced to 18 (additional experimental points), with a view to containing the number of experiments to be carried out. The procedure is stopped when the maximum number of iterations is reached. This parameter has been set to 10 for all but the fifth DeJong test function, while a value of 20 has been employed for the latter, due to its higher level of complexity. A threshold value $\alpha = 0.001$ is used to evaluate the algorithm’s convergence. In particular, convergence is reached when the best solution identified by the procedure is in the range $[y^* \pm \alpha]$, where $y^*$ is the test function optimum. The number of function evaluations required for convergence is counted, and the purpose is to keep it low. For each function, 20 simulation runs are conducted and the average number of function evaluations is counted. The number of function evaluations corresponds to the number of experimental points selected by EMMA, and has been chosen as a measure of the procedure performance because the aim is to identify the best process conditions with the smallest possible number of design points. This allows to take into consideration the constraints deriving from the limited availability of the resources that typically exist in real-world applications (e.g., budget, time, materials, etc). Table 4.2 reports the empirical

Figure 4.8: The Ackley function is characterized by several local minima and one global minimum.
Table 4.2: Six benchmark functions are used to test the convergence properties of the proposed approach. The table reports the number of function evaluations required to converge to the true optimum with an accuracy of $\alpha = 0.001$. The number of function evaluations corresponds to the number of experiments required.

<table>
<thead>
<tr>
<th>Test function</th>
<th>Function evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere function</td>
<td>75</td>
</tr>
<tr>
<td>Rosenbrock’s saddle</td>
<td>139</td>
</tr>
<tr>
<td>Step function</td>
<td>88</td>
</tr>
<tr>
<td>Quartic function</td>
<td>116</td>
</tr>
<tr>
<td>Shekel’s foxholes</td>
<td>115</td>
</tr>
<tr>
<td>Ackley’s function</td>
<td>81</td>
</tr>
</tbody>
</table>

results. In order to be able to achieve the desired accuracy for convergence, the values of the responses should be in the interval $[y^* \pm \alpha]$. This occurs if a suitable number of levels per factor (input variable) is used. As a consequence, the number of levels per factor is chosen to generate a sufficiently fine grid of points. To study the convergence on the quartic function in equation (4.2.4), the values of the input variables are investigated in place of the response values, because of the presence of the noise. To identify the values of the input variables which result in response values in the range $[y^* \pm \alpha]$, the response values are generated in the absence of the noise term. Values of the input variables in $[-0.2, 0.2]$ correspond to optimal responses values. As a result, convergence is achieved when the developed procedure identifies solutions with inputs in the range $[-0.2, 0.2]$. Of all of the problems in the DeJong test suite, the Rosenbrock’s saddle is the most challenging for the developed approach. EMMA is shown to cope with the noise in the quartic function. The multimodality and unsmoothness of the Shekel’s foxholes are also faced successfully, and convergence is reached for the Ackley function as well.

Of interest is also the comparison of EMMA with other approaches. However, the literature on NPRSM and DACE seems to be lacking of published works evaluating the procedure on standard benchmark functions and measuring the performances in terms of the number of function evaluations required to achieve a desired accuracy. For example, Zhou et al. (2007) proposed a model-based evolutionary optimization framework, that uses both global and local surrogate models, and which is aimed to reduce the computational cost of the evolutionary algorithm employed. The performance analysis of the procedure is
performed on a set of benchmark functions, including the sphere, the Rosenbrock, and the Ackley test function. However, the convergence trend of the procedure is employed for comparison with other approaches. The same holds for other papers including for example Hutter et al. (2009) and Huang et al. (2006). Other works evaluate the procedure performance either on nonstandard test function (Jin, Chen & Simpson 2001, Rutherford 2006, Simpson et al. 1998) or only in terms of the predictive accuracy (Hu et al. 2008, Jin et al. 2002, Qian et al. 2006). The work by Li et al. (2009) enables for a comparison. The authors propose a two-stage optimization procedure that employs a thin plate spline model and an improved selection criteria, which is based on the expected improvement. They evaluate the procedure proposed on a two-dimensional modified Rosenbrock function and on a two-dimensional Ackley function, and they investigate the accuracy of the approach in terms of number of function evaluations. However, they performed only 3 simulation runs, and this is hardly considered to provide reliable results. For comparison purposes, the performances of EMMA has been investigated on the same set of benchmark functions and the threshold values declared in Li et al. (2009) have been considered. 20 simulation runs have been conducted and the average number of function evaluations has been calculated. EMMA resulted in slightly better results compared to the methodology proposed by Li et al. (2009). Indeed, EMMA employed an average number of function evaluations equal to 43 for the Rosenbrock function, and 65 for the Ackley function. Instead, the procedure proposed by Li et al. (2009) employed about 47 and 66 function evaluations for the Rosenbrock and Ackley benchmark functions, respectively. These are promising results, but further analysis and simulation are required and need to be addressed.

4.3 EMMA: the success in the chemical problem

As was stated in section 2.2, $X = (X_1, \ldots, X_6)$ denotes the 6-dimensional vector of chemical components (factors), and $Y = (Y_1, \ldots, Y_5)$ denotes the 5-dimensional vector of responses measuring the chemical composition stability and the obtained coating quality. According to Table 2.1, each factor is studied at 6 levels, giving rise to an experimental region $E$ composed of a grid of $6^6$ design points. Because of the constraints in equations
(2.2.2) and (2.2.3), this number results in $N = 8424$ feasible experimental points. Let $x_n = (x_{n1}, \ldots, x_{np})^T$ be the $n$th experimental point in $E$, where $x_{ni}, i = 1, \ldots, p$, is the level value of the $i$th factor for the $n$th design point, and it indicates the number of moles of the $i$th chemical component in the $n$th chemical mixture. Similarly, let $y_n = (y_{n1}, \ldots, y_{nq})^T$ be the $q$-dimensional vector of measured response values for the $n$th experimental point. In particular, $y_{n1}$ measures the stability of the $n$th sol-gel composition, whereas the responses $y_{nj}, j = 2, \ldots, q$, measure the goodness of the $n$th coating in terms of the spots’ characteristics. Each time a new set of coated slides is synthesized, a reference sample is prepared. To obtain the reference, the composition identified using the best-guess approach is employed. The best-guess approach consists of selecting an arbitrary combination of the factors, test them, and define a new experiment according to the results of the previous test. This strategy of experimentation is used frequently in practice by engineers and scientists. It is worth noting that the best-guess approach can work reasonably well if the experimenter have a great deal of technical or theoretical knowledge of the system investigated, as well as considerable practical experience (Montgomery 2009). The use of the reference permits a comparison of the performances of the coatings obtained using the two different approaches (EMMA vs. best-guess). As was stated formally in section 2.2, the aim is to find the chemical composition that generates spots with optimal features.

The optimization problem is as stated in section 2.2. Challenges in the multi-objective optimization problem arise from the competing nature of Int and Back. Indeed, the intensity and the background signals both depend on the same elements, such as the amount of bounded dye molecules and grafting conditions, and are expected to react in the same ways to these elements. However, the interest here is in the maximization of Int and the minimization of Back, and this necessarily require a compromise solution.

The initial design $D_0$ is obtained by randomly sampling $N_0 = 48$ design points from the experimental region $E$. This is how the swarm is initialized. The experiments in $D_0$ are carried out and the responses are measured as described in section 2.1.2. A multiresponse MARS model is then fitted to the collected data and used to predict the responses of the remaining $N - N_0$ points in the experimental region. This allow us to obtain all of the values in the $N \times q$ response data matrix $Y$, that has column vectors $y_1, \ldots, y_q$ and $y_j = (y_{1j}, \ldots, y_{Nj})$ for $j = 1, \ldots, q$. Stable compositions are selected and the response $Y_1 = \text{Stab}$
is temporarily removed from the analysis. Note that unstable compositions cannot be used to cover the slides, and as a result, all of the responses assume a value of zero. The response variable \( Y_4 = \text{Back} \) is transformed into \( 1 - \text{Back} \), so that all of the responses can be maximized. Both the measured and predicted response values of the \( N \) experimental points in the design region \( E \) are normalized in the range \([0, 1]\). The response values are then weighted using the vector \( w = (w_2, \ldots, w_q) \) of weights, where \( w_2 = 0.4, w_3 = 0.4, w_4 = 0.12, \) and \( w_5 = 0.08 \) (see section 2.2). The target, or demand-level vector, \( y^* \), is identified by combining the maximum value of each single response (Khuri & Conlon 1981, Srinivas & Deb 1994). That is, the target is given by \( y^* = (\max(y_2), \ldots, \max(y_5))^T \). For each design point in \( D_0 \), the distance between the measured response values \( y_n \) and the target \( y^* \) is calculated. In particular, the weighted response values for the \( n \)th experimental point \( u_{nj} = w_j \cdot y_{nj} \), and the weighted objective point \( u^*_j = w_j \cdot y^*_j \), are used to calculate the distance

\[
d_n = \begin{cases} 
1000 & \text{if } y_{n1} = 0 \\
\left( \sum_{j=1}^{q} |u_{nj} - u^*_j| \right)^{1/r} & \text{if } y_{n1} = 1 
\end{cases}.
\]

(4.3.1)

Here, \( r \) has been set equal to 1. The distance function in equation (4.3.1) is used to identify the global best and the personal bests. Given the set \( S \) of all of the design points investigated, the global best is the experimental point \( g \in S \), whose measured responses are closest to the target. Similarly, given the set \( T \) of all of the design points investigated by the same particle, the personal best is the experimental point in \( T \), whose measured responses are closest to the target. Indeed, each particle is allowed to move and study new experimental points in \( E \). Clearly, the number of personal bests is equal to the number of particles and, consequently, to the size of the experimental design. In the subsequent steps of the procedure, the dimensionality of the initial design is reduced from \( N_0 = 48 \) to \( N_t = 15 \) design points in order to reduce the number of experiments required. This is reasonable also for the optimization procedure that has been shown to successfully converge to the true optimum when reducing the initial size of the swarm. In particular, the \( N_t \) experimental points in \( D_0 \) associated to the \( N_t \) smallest distance values are selected. Then, for each iteration \( t = 1, \ldots, C \), the subsequent steps are repeated. The global best and the personal bests are obtained, and each particle’s velocity and position are updated using equations (4.1.3) and (4.1.4), so that \( N_t \) additional design points are selected. Finally, to
introduce a certain degree of randomness into the particles’ flights, the mutation operator in equation (4.1.8) is applied to each value $x_{ni}$. The probability with which the factor level $x_{ni}$ are mutated decreases with $t$, and has been varied from 0.1 at $t = 0$, to 0.07 at $t = 1$, 0.04 at $t = 2$, and 0.01 at $t > 2$. At each iteration $t$, the selected experiments are carried out, the coatings are obtained, the spots quality is measured, a multiresponse MARS model is fitted to the collected data, the target is identified, and the $N_t$ new experimental points are selected by minimizing the distance-based function in equation (4.3.1) (see Figure 4.1). The final output produced by the algorithm is the best chemical component found during the whole procedure.

### 4.3.1 Applicative results

A total of 123 recipes have been investigated in six iterations ($t = 0, \ldots, 5$). The best recipe identified by the proposed approach is as follows: EtOH=0.17 moles, HCl 1M=0.00046 moles, H2O=0.014 moles, TEOS=0.00263 moles, MTES=0.0016 moles, APTES= 0.000588 moles. Figure 4.9 shows the scanned images of two spots: one measured on the best coating found using the best-guess approach approach (left) and the other measured on the best coating found using EMMA (right). Compared to the best composition found using best-guess approach, significant improvements have been obtained for the majority of the responses. In particular, stable compositions and perfectly circular spots have
been found by both of the approaches investigated (best-guess approach and EMMA). A remarkable improvement of 380% has been measured for the response Int, and an enhancement has been obtained for Back too (39%). Finally, a slight worsening has been obtained in terms of Hom (89%). With respect to the best best-guess approach coating, EMMA has therefore been able to identify a better composition in terms of the signal-to-noise ratio (246% enhancement), thus successfully facing also the challenge deriving from the competing nature of the response variables Int and Back.

4.4 EMMA: towards the multiresponse model selection

A one-stage distribution-free approach, named Evolutionary Model-based Multiresponse Approach (EMMA), has been developed for the optimization of high dimensional systems with multiple responses. An initial design is generated and sequentially augmented by employing a multiresponse MARS model and a time-varying PSO algorithm. The MARS model is used to select the target, namely the vector of response values obtained by optimizing each predicted response separately. The PSO is subsequently used to select additional experimental points by minimizing the distance between the observed responses and the target. Compared to the existing two-stage approaches, the developed technique does not rely strongly on the fitted (or surrogate) model to augment the initial design. The main difference between the proposed methodology and the two-stage methods lies in the purpose of the modeling step. In the former, the aim is to select a target, whereas in the latter, the goal is to provide a good approximation of the unknown function. The underlying assumption of two-stage methods is that the approximating function provides a faithful representation of the real system. As a result, the fitted model is optimized to identify the optimum conditions for the real system. However, if the initial sample is sparse, the surrogate model (predictor) can give a highly misleading view of the approximated function. Because the misleading view is taken to be correct in the second stage, the algorithm may either stop prematurely or become excessively local in its selection of new experimental points (Jones 2001). Improved two-stage approaches have therefore been developed in order to avoid premature convergence. Their aim is to identify additional experimental points in parts of the experimental region that has
been relatively unexplored, namely where the predictor uncertainty (or predictor error) is higher. From a global optimization perspective, searching the prediction error amounts to exploring the search space, whereas searching the predictor itself is equivalent to exploiting basins of attraction which are currently known to be promising (Söbester et al. 2005). The approach described in this chapter deals with a balance between exploration and exploitation through the use of a PSO heuristic with time varying parameters and a mutation operator. Exploration is encouraged at the initial stages of the heuristic, thanks to the use of a relatively big initial design and to the greater importance which is assigned to each particle’s individual experience. As the algorithm evolves, the influence of the single particles is reduced and a greater relevance is assigned to the swarm’s experience. This enables the exploitation of the most promising region of the search space. Compared to the one-stage approaches in the literature (Jones 2001, Gutmann 2001), EMMA has the advantage of not requiring the target to be specified. Indeed, the target is selected by the MARS model automatically, according to a data-driven strategy. As a consequence, there is no need to either specify a goal or employ multiple targets in a trial-and-error approach. Furthermore, as the algorithm evolves, the target is expected to get closer and closer to the true optimal values of the responses. Indeed, promising regions of the experimental space are sampled increasingly thus enhancing the accuracy of the MARS model in these areas. With respect to methods which employ GP models, EMMA is characterized by a reduced computational cost. Indeed, for $N$ data points, GP models require $O(N^3)$ steps to construct the GP, $O(N)$ to predict the mean function value at a new point, and $O(N^2)$ to predict the standard deviation (Büche et al. 2004). On the other hand, the computational cost of the MARS procedure can be made proportional to $pN M_{\text{max}}$, where $p$ is the number of inputs, $N$ is the number of data points, and $M_{\text{max}}$ is the maximum number of basis functions (Friedman 1991). One advantage of the proposed methodology over the two-stage approaches, is that it can handle extremely deceptive problems (Jones 2001). The use of the MARS model allows to reduce the computational cost of the procedure and to automate the target selection step. Furthermore, EMMA is distribution-free and the Gaussian random variable assumption typically required by other methods is not needed. The MARS procedure has been employed in the developed methodology. However, various other models can also be included simply. The challenge is to select the modeling
strategy capable of providing the highest performance. As was stated by Breiman & Friedman (1997), this ‘...may depend on the detailed nature of the problem at hand in terms of the number of observations, the number of response variables, their correlation structure, signal-to-noise ratio, collinearity of the predictor variables, etc.’. Of interest is therefore the study of the performance of modeling techniques as well as their relationship with the problem characteristics. This motivates the second part of the thesis. Models which have been developed to deal with multiple responses are described in chapter 5. A research methodological framework is proposed in chapter 6 to generate a knowledge system for the selection of multivariate models based on the characteristics of the data.
Chapter 5

Multiresponse regression

This chapter provides a literature review on techniques that can be employed to model the relationship between multiple inputs (predictors) and multiple outputs (responses). Two different approaches exist, namely separate (uniresponse) multiple regression modeling and multivariate multiple regression modeling (Breiman & Friedman 1997, Hastie et al. 2009). The first consists on fitting a model to each response separately, whereas the latter considers all the outputs simultaneously in the attempt to select the set of model terms that is best across all the responses (Hastie et al. 2009). Interest on separate and multivariate multiple regression arises from questions such as ‘Is MARS the best modeling technique usable in the one-stage distribution-free approach proposed in chapter 4?’ and ‘Would separate MARS models have been better than the employed multivariate MARS model?’. To help answering these questions a meta-learning framework is proposed and described in chapter 6. The aim of the current chapter is to provide the theoretical background of the modeling techniques that will be analyzed in the proposed meta-learning framework.

The chapter is organized as follows. Section 5.1 describes the multivariate regression problem and introduces the separate and multivariate multiple regression approaches. Subsequently a number of modeling techniques are described. In particular, section 5.2 gives an account of separate multiple regression approaches and section 5.3 depicts multivariate multiple regression methodologies.
5.1 The multivariate regression problem

Let $Y = (Y_1, \ldots, Y_q)^T$ be a vector of $q$ output variables and let $X = (X_1, \ldots, X_p)^T$ be a vector of $p$ input variables. Assume a set of training data is available, namely $(x_1, y_1) \ldots (x_N, y_N)$. Denote by $X$ the $N \times p$ matrix of observed inputs where each row is a vector $x_n = (x_{n1}, \ldots, x_{np})$, $n = 1, \ldots, N$, and each column is denoted by $x_i$, $i = 1, \ldots, p$. Similarly, denote by $Y$ the $N \times q$ matrix of observed outputs where each row is a response vector $y_n = (y_{n1}, \ldots, y_{nq})$ and each column is denoted by $y_j$, $j = 1, \ldots, q$. Finally, denote by $\epsilon = (\epsilon_1, \ldots, \epsilon_q)$ the $N \times q$ matrix of errors, and by $f(X)$ the vector function $f_1(X), \ldots, f_q(X)$. The purpose in multivariate regression is to study to what extent the behaviour of the $q$ output variables is influenced by the set of $p$ input variables. In doing this it has to be considered that not only the components of $Y$ might be correlated with the components of $X$, but also that the components of $Y$ might be correlated with each other (similarly to how the components of $X$ might be correlated with each other) (Izenman 2008). Two approaches exist to cope with multiresponse systems. They are separate and multivariate multiple regression. In separate multiple regression each response is regarded separately and the functional relationship with the predictors differs for each output:

$$Y_1 = f_1(X) + \epsilon_1$$

$$\ldots$$

$$Y_q = f_q(X) + \epsilon_q.$$

A predictive relationship is to be modeled between each $Y_j$, $j = 1, \ldots, q$, and the complement set $X$ of variables, designated as predictors. In multivariate multiple regression the responses are assumed to share the same functional relationship with the predictors, namely

$$Y_1 = f(X) + \epsilon_1$$

$$\ldots$$

$$Y_q = f(X) + \epsilon_q.$$
Because $Y_1, \ldots, Y_q$ are assumed to have the same dependence on the predictor variables, the responses are correlated. Therefore, the degree of similarity of the dependence of a pair of responses $(Y_i, Y_j)$ on the predictors is reflected in their correlation. A large positive (or negative) correlation between $Y_i$ and $Y_j$ means that the corresponding (true) coefficient vectors, say $\beta_i$ and $\beta_j$, should be closely related (that is $\beta_i \sim \beta_j$ or $\beta_i \sim -\beta_j$). Similarly, small correlations imply no special relationship ([Frank & Friedman 1993]). The purpose of multivariate multiple regression is to estimate the regression relationship between $Y$ and $X$ taking into account not only the various dependencies between $X$ and $Y$, but also the dependencies within $X$ and within $Y$ ([Izenman 2008]). Strategies that do not take into account the correlations between the responses belong to the class of separate regression techniques described in section 5.2, whereas strategies that exploit correlations in the different responses are part of the multivariate regression methods, described in section 5.3.

It is well-known that, if a linear model is assumed, separate and multivariate regression coincide. Indeed, the least squares estimates of a multivariate multiple linear model are simply the individual least squares estimates for each of the outputs ([Hastie et al. 2009]). Denote by $\beta$ the $q \times p$ matrix of parameters with each column a vector $\beta_j = (\beta_{j1}, \ldots, \beta_{jp})$, $j = 1, \ldots, q$. Then, the assumed linear model (in matrix notation) is given by

$$Y = X\beta + \epsilon.$$  \hfill (5.1.1)

A straightforward generalization of the unireponse loss function is

$$RSS(\beta) = tr[(Y - X\beta)^T (Y - X\beta)].$$ \hfill (5.1.2)

The least squares estimates obtained minimizing criteria (5.1.2) have exactly the same form as in the unireponse case, namely

$$\hat{\beta} = (X^TX)^{-1}X^TY$$ \hfill (5.1.3)
and this result holds even if the errors are assumed to be correlated. Specifically, suppose \( \text{Cov}(\epsilon) = \Sigma \), then the multivariate weighted criterion

\[
\text{RSS}(\beta; \Sigma) = \text{tr}\left[(Y - X\beta)^T \Sigma^{-1} (Y - X\beta)\right]
\] (5.1.4)

arises naturally from multivariate Gaussian theory. It can be shown that again the solution for (5.1.4) is given by (5.1.3), that is \( q \) separate regressions that ignore the correlations between the responses. However, if \( \Sigma_i, i = 1, \ldots, N \), vary among observations, then this is no longer the case and the solution for \( \beta \) no longer decouples (Hastie et al. 2009). Various alternatives have been proposed to improve the predictive accuracy of model (5.1.1). These techniques employ feature selection methods, shrinkage methods and/or approaches for feature extraction. Feature selection methods aim to produce a model that is interpretable and has possibly lower prediction error than the full model by retaining a subset of the predictors and discarding the rest. However, because it is a discrete process (variables are either retained or discarded) it often exhibits high variance, and so does not reduce the prediction error of the full model. Shrinkage methods are preferable because they are more continuous and do not suffer as much from high variability. Finally, feature extraction refers to the creation of a reduced set of linear or nonlinear transformation of the input variables by projection methods. The purpose is to find those low-dimensional projections of the input data that enjoy some sort of optimality properties (Hastie et al. 2009, Izenman 2008). Examples of feature selection methods are best subset selection and forward- and backward-stepwise selection. Examples of shrinkage methods are ridge regression (RR) and the Lasso. Examples of feature extraction methods are principal component analysis (PCA) and canonical correlation analysis (CCA) (Hastie et al. 2009, Izenman 2008). The main approaches developed to cope with multiple responses have roots in the aforementioned methods and are described further below.

5.2 Separate (uniresponse) multiple regression

Ordinary least squares (OLS) estimates can be dominated substantially by a variety of biased or regularized alternatives (Breiman & Friedman 1997). Biased or regularized regression methods are applied to the multivariate problem, described in section 5.1, to
improve the predictive accuracy of each single response $Y_j, j = 1,\ldots,q$. This is achieved by simply replacing each unireponse OLS regression with a corresponding biased or regularized regression (Frank & Friedman 1993). Even if this procedure does not take advantage of the dependence between the responses, it can improve substantially the model predictive accuracy. In what follows the theoretical background of biased and regularized regression is provided. In this section, each single response is considered separately, that is $Y$ denotes the real-valued output and $\mathbf{y} = (y_1,\ldots,y_N)^T$ denotes the $N$-dimensional vector of output values in the training set.

Suppose that the relationship between $Y$ and $X$ is linear. Then, the linear regression model has the form

$$\mathbf{y} = \mathbf{X}^T \beta + \epsilon$$

(5.2.1)

where $\beta = (\beta_1,\ldots,\beta_p)^T$ is the parameter vector and $\epsilon$ is the error term (Hastie et al. 2009). Ordinary least squares (OLS) estimates are given by

$$\hat{\beta}_{OLS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X} \mathbf{y}.$$  (5.2.2)

Because they depend upon $(\mathbf{X}^T \mathbf{X})^{-1}$, numerical complications is experienced in computing $\hat{\beta}_{OLS}$ if $\mathbf{X}^T \mathbf{X}$ is singular or nearly singular ($\mathbf{X}^T \mathbf{X}$ is singular if $\mathbf{X}$ is less than full rank). This occurs if $\mathbf{X}$ is ill-conditioned, the columns of $\mathbf{X}$ are collinear, or there are more variables than observations. If $\mathbf{X}$ is ill-conditioned, the quantities to be computed are sensitive to small changes in the data and small changes in $\mathbf{X}$ lead to large changes in $(\mathbf{X}^T \mathbf{X})^{-1}$. The estimator in equation (5.2.2) becomes therefore computationally unstable, and the individual component estimates may either have the wrong sign or be too large in magnitude. So, even though the regression model may be a good fit to the learning data, it will not generalize sufficiently well to the test data (Izenman 2008, Hastie et al. 2009). One way out of this situation is to consider a biased or regularized estimator of $\beta$ (Izenman 2008, Frank & Friedman 1993). Biased methods deal with singular $\mathbf{X}^T \mathbf{X}$ in three different ways. The first is to use a generalized inverse in place of $(\mathbf{X}^T \mathbf{X})^{-1}$ (Izenman 2008, Hastie et al. 2009). The generalized inverse of a matrix $\mathbf{C}$ is defined when $\mathbf{C}$ is either singular or nonsymmetric (or even not square). A $g$-inverse of a $J \times K$–matrix $\mathbf{C}$ is any $(K \times J)$–matrix $\mathbf{C}^-$ such that, for any $J$-dimensional vector $\mathbf{Y}$ for which $\mathbf{C} \mathbf{X} = \mathbf{Y}$ is a
consistent equation, \( X = C - Y \) is a solution. It can be shown that \( C^- \) exists iff

\[
CC^- C = C. \quad (5.2.3)
\]

A \( g \)-inverse always exists although in general it is not unique (Mardia et al. 1979, Izenman 2008). A unique \( g \)-inverse can however be defined for the \( J \times K \)-matrix \( C \) using its singular value decomposition \( C = U \Psi V^T \). In particular, set

\[
C^+ = V \Psi^+ U^T, \quad (5.2.4)
\]

where \( \Psi^+ \) is a diagonal matrix whose diagonal elements are the reciprocals of the nonzero elements of \( \Psi = \Lambda^{1/2} \), and zero otherwise. The \( K \times J \)-matrix \( C^+ \) is the unique Moore-Penrose generalized inverse of \( C \) (Izenman 2008). Suppose \( X^T X \) has known rank \( t \) \((1 \leq t \leq p)\), so that the smallest \( p - t \) eigenvalues of \( X^T X \) are all zero. Then, the singular value decomposition of \( X^T X \) can be written as

\[
A = V \Lambda V^T, \quad (5.2.5)
\]

where \( \lambda = \text{diag}(\lambda_1, \ldots, \lambda_t) \) is a diagonal matrix of the first \( t \) eigenvalues of \( A \) with diagonal elements ordered in magnitude from largest to smallest, and \( V = (v_1, \ldots, v_t) \) is and \( p \times t \)-matrix whose columns are the eigenvectors associated with the eigenvalues in \( \Lambda \). The unique rank-\( t \) Moore-Penrose inverse \( G \) of \( X^T X \) is given by

\[
G = (X^T X)^+ = V \Lambda^{-1} V^T = \sum_{i=1}^{p} \lambda_i^{-1} v_i v_i^T. \quad (5.2.6)
\]

However, the actual rank of \( X^T X \) is typically unknown. Furthermore, if the used rank is different from the actual one, GIR could produce a biased estimator (Izenman 2008). The second way to deal with singular matrices is to use a rotated predictor matrix in place of \( X \). This is the approach adopted in principal component regression (PCR) and partial least squares regression (PLSR), which are described further below. The third way to cope with singular \( X^T X \) consists in adding a small constant value \( \lambda \) to the diagonal entries of the matrix \( X^T X \) before taking its inverse. This is the principle used in ridge regression (RR). Along with biased methods, regularized methods can be used. Regularized methods
simultaneously take advantage of subset selection and shrinkage to improve the model predictive accuracy (Izenman 2008). Examples of regularized methods are the Lasso and elastic net. Regularized estimates are obtained minimizing a penalized least squares criterion that can be written in the following general form

$$\phi(\beta) = (y - X\beta)^T(y - X\beta) + \lambda p(\beta), \quad (5.2.7)$$

where $p(\cdot)$ is a given penalty function and $\lambda$ is the regularization parameter. A family of penalized least-squares estimators, indexed by $k$, can be defined using the penalty function

$$p_k(\beta) = \sum_{i=1}^{p} |\beta_i|^k, \quad (5.2.8)$$

that bounds the $L_k$-norm of the parameters in the model as

$$\sum_i |\beta_i|^k \leq c. \quad (5.2.9)$$

By substituting the penalty function $p_k(\beta)$ in equation (5.2.8) in place of $p(\beta)$ in equation (5.2.7), the following criterion is obtained

$$\phi_k(\beta) = (y - X\beta)^T(y - X\beta) + \lambda p_k(\beta) \quad (5.2.10)$$

where $k > 0$. The penalty $\phi_k(\beta)$ is a function that is smooth and convex when $k > 1$, convex for $k = 1$, and non-convex when $k < 1$ (see Figure 5.1). As a consequence, if $k \geq 1$ classical optimization methods can be used to minimize $\phi_k(\beta)$, while the minimization problem is more complicated if $k < 1$ (Izenman 2008). By changing the values of $k$, different methods are obtained. Ridge regression corresponds to $k = 2$ and its penalty function is a rotationally invariant hypersphere centered at the origin (see Figure 5.2, right). The tuning parameter $c$ in equation (5.2.9) controls the size of the hypersphere and, hence, how much the parameter estimates $\hat{\beta}$ are shrunk toward the origin. When $k < 2$ the penalty function is no longer rotationally invariant and collapses toward the coordinate axes combining elements of ridge regression and variable selection (the coefficients are shrunk toward zero while some of them are set to be exactly zero). When $k$ is set very close to 0, the result is variable selection. Finally, if $k = 1$ we obtain the Lasso (see Figure
Figure 5.1: Two-dimensional contours of the symmetric penalty function $p_k(\beta) = |\beta_1|^k + |\beta_2|^k = 1$ for $k = 0.2, 0.5, 1, 2, 5$. A value of $q = 1$ (blue diamond) yields the Lasso, whereas a value of $q = 2$ (red circle) yields ridge regression (Izenman 2008).

Figure 5.2: Contours of the error and constraint functions for the Lasso (left) and ridge regression (right). The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq c$ and $\beta_1^2 + \beta_2^2 \leq c^2$, respectively. The red ellipses are the contours of the least squares error function (Hastie et al. 2009).
The penalty functions used in ridge regression and Lasso can be combined to face strong correlations among the input variables. The resulting technique is referred to as elastic net (Izenman 2008).

### 5.2.1 Principal component regression

Principal component regression (PCR) has been introduced by Massy (1965). It regresses the response variable on linear combinations of the original inputs, the principal components. Because principal components depend on the scaling of the inputs, they are typically standardized to have zero mean and unit standard deviation. PCR is based on the eigenanalysis of the predictor covariance matrix $\mathbf{X}^T \mathbf{X}$

$$\mathbf{X}^T \mathbf{X} = \mathbf{U} \mathbf{E}^2 \mathbf{U}^T,$$

(5.2.11)

where

$$\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I}_p, \quad \text{and} \quad \mathbf{E}^2 = \text{diag}(e_1^2, \ldots, e_t^2, 0, \ldots).$$

(5.2.12)

Here, $t < p$ is the rank of $\mathbf{X}^T \mathbf{X}$. The first $t$ eigenvalues $\mathbf{E}_t = \{e_1^2, \ldots, e_t^2\}$ are nonzero and in descending order. Similarly, let $\mathbf{U}_t$ denote the first $t$ columns of the matrix of eigenvectors $\mathbf{U}$. $\mathbf{E}_t$ and $\mathbf{U}_t$ correspond to $\mathbf{\Lambda}$ and $\mathbf{V}$ in equation (5.2.5), respectively. Denote by $\mathbf{Z}_t$ the $N \times t$ matrix formed by the first $t$ columns of the rotated predictor data matrix

$$\mathbf{Z}_t = \mathbf{X} \mathbf{U}_t,$$

(5.2.13)

that is the matrix whose rows are the scores of the first $t$ principal components of $\mathbf{X}$. Then, $\mathbf{Z}_t$ is used in place of $\mathbf{X}$ and the response $\mathbf{y}$ is regressed on the principal components $\mathbf{Z}_t$, so that the regression coefficient estimates associated with the last $p - t$ columns are all defined to have 0 value. This is equivalent to using the Moore-Penrose generalized inverse of $\mathbf{X}^T \mathbf{X}$, as previously described (Breiman & Friedman 1997). The drawback of the described approach is that the resulting (non-zero) coefficient estimates are likely to be highly variable owing to the fact that $\mathbf{Z}_t^T \mathbf{Z}_t$ is still likely to be poorly conditioned. This can be remedied by making the rank value $t$ a model selection parameter to be estimated through cross-validation (Breiman & Friedman 1997). This is the approach adopted in
PCR. Indeed, PCR first forms the matrix \( Z_t \) with input columns \( z_m, m = 1, \ldots, M \), and then regress \( y \) on \( z_1, z_2, \ldots, z_M \) for some \( M \leq p \). Since the \( z_m \) are orthogonal, the regression is just a sum of univariate regressions

\[
\hat{y}^{(M)} = \bar{y} + \sum_{m=1}^{M} \hat{\beta}_m z_m,
\]

where \( \bar{y} = \sum_{n=1}^{N} y_n \) and

\[
\hat{\beta}^{(M)}_{\text{PCR}} = \left( z_m^T z_m \right)^{-1} z_m^T y = E^{-1} U^T s
\]

are the least squares estimates of the \( m \)th model and \( s = X^T y \). The result in PCR is a sequence of regression models indexed by \( M \) and the goal is to choose the model with the lowest prediction mean squared error (Hastie et al. 2009). When \( M = p \) the usual least squares estimates are generated because the columns of \( Z \) span the column space of \( X \). For \( M < p \) a reduced rank regression is obtained. The quantity \( M \) can thus be considered a meta parameter of the procedure whose value is to be estimated from the training data through some model-selection procedure (Breiman & Friedman 1997). Cross-validation (CV) is nearly always used for that purpose even if many other statistical model selection criteria can be employed (e.g., generalized cross-validation, minimum descriptive length, Bayesian information criterion, and Mallows’s \( C_p \)) (Hastie et al. 2009). Because the extraction of the principal components is accomplished without any reference to the output variable \( Y \), PCR can fail dramatically. Indeed, we have no reason to expect \( Y \) to be highly correlated with any of the principal components, in particular those having the largest eigenvalues. \( Y \) may actually have its highest correlation with one of the last few principal components or even only the last one which is always dropped from the regression equation (Jolliffe et al. 2003, Hastie et al. 2009).

### 5.2.2 Partial least squares regression

Partial least squares regression (PLSR) has been introduced by Wold (1966). In many respects, PLSR is similar to PCR. First of all, PLSR constructs a set of linear combinations of the inputs for regression. However, because \( y \) is used in this construction in addition to \( X \), PLRS is less likely to fail with respect to PCR (Hastie et al. 2009). Indeed, in PLSR the derived variables (latent variables) are specifically constructed to retain the most of the
information in $X$ that helps predict $y$, while at the same time reducing the dimensionality of the regression (Izenman 2008). Another similarity with PCR is that PLSR produces a sequence of models indexed by $M$ and selects the best one using CV. As with PCR, if $M = p$ the solution is equivalent to the usual least squares estimates, whereas if $M < p$ a reduced regression is produces (Frank & Friedman 1993, Hastie et al. 2009). Finally, PLSR is not scale invariant so each $X_i$ is initially standardized to have zero mean and unit variance (Izenman 2008).

Typically, PLSR is obtained using an algorithm rather than as the result of an optimization procedure. Several algorithms have been proposed. The most popular one is a sequential procedure according to which, at each step $M$, $y$ residuals from the previous step ($y_{M-1}$) are partially regressed on $X$ residuals from the previous step ($X_{M-1}$). At the beginning of the procedure the residuals are initialized to the original (standardized) data. The partial regression consists of computing the covariance vector $w_M$ and then using it to form a linear combination $z_M$ of the $X$ residuals. The $y$ residuals are then regressed on this linear combination, the result is added to the model and subtracted from the current $y$ residuals to form the new $y$ residuals $y_M$ for the next step. New $X$ residuals ($X_M$) are then computed by subtracting from $X_{M-1}$ its projection on $z_M$ (Frank & Friedman 1993). The final test will cause the algorithm to terminate after $M$ steps. Let

$$
\langle x, y \rangle = \sum_{n=1}^{N} x^T y
$$

(5.2.16)

denote the inner product between $xy$ In synthesis,

1. initialize: $y_0 \leftarrow y$, $X_0 \leftarrow X$, $\hat{y}_0 \leftarrow 0$

2. for $M = 1$ to $p$ do:
   - $w_M = \langle y_{M-1}, X_{M-1} \rangle$
   - $z_M = w_M^T X_{M-1}$
   - $r_M = [\langle y_{M-1}, z_M \rangle / \langle z_M, z_M \rangle] z_M$
   - $\hat{y}_M = \hat{y}_{M-1} + r_M$
   - $y_M = y_{M-1} - r_M$
\[ X_M = X_{M-1} - \frac{\langle z_M, x_{M-1} \rangle}{\langle z_M, z_M \rangle} z_M \]

\[ \text{if } \langle X_M^T, X_M \rangle = 0 \text{ then Exit} \]

The result is a sequence of prediction models indexed by \( M \), where the \( M \)th model predicts the output variable \( y \) through a linear function of the first \( M \) latent variables. The best model is that model that minimizes a cross-validation estimate of the prediction error. Differently from PCR, PLSR is not a linear modeling procedure. Indeed, the response values \( y \) enter nonlinearly into the model estimates \( \widehat{y} \). It can be shown that the PLSR estimator is a shrinkage estimator, even if this is a difficult result to prove (Izenman 2008).

### 5.2.3 Separate (uniresponse) ridge regression

The ridge regression (RR) estimator is obtained by adding a small constant value \( \lambda \) to the diagonal entries of the matrix \( X^T X \) before taking its inverse. It is given by

\[
\widehat{\beta}_{RR} = (X^T X + \lambda I)^{-1} X^T y
\]  

(5.2.17)

This makes the problem nonsingular, even if \( X^T X \) is not of full rank (Hastie et al. 2009). \( \widehat{\beta}_{RR} \) in equation (5.2.17) is the solution to the penalized least square criterion

\[
RSS(\lambda) = (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta.
\]  

(5.2.18)

RR can therefore be seen also as a regularization (shrinkage) method that shrinks the regression coefficients by imposing a penalty on their size. By shrinking some coefficients to zero, prediction accuracy of the least squares estimates can sometimes be improved. Because of the bias-variance trade-off this is done at the cost of scarifying a little bit of bias (Hastie et al. 2009). The requirement of an unbiased estimator of \( \beta \) is therefore abandoned and biased estimators are obtained. In the presence of near collinearity (e.g., many variables are highly correlated), these biased estimators have a greater prediction accuracy (superior in terms of MSE) with respect to \( \widehat{\beta}_{OLS} \) (Hastie et al. 2009). To characterize the ridge estimator as a shrinkage estimator, consider the singular value decomposition of the matrix \( X^T X \) in equation (5.2.5). Let \( Z = XV = V\Lambda^{1/2} \) so that \( Z^T Z = \Lambda \). Here, \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_t) \) is the matrix of the nonzero eigenvalues and \( V \) is the matrix whose
columns are the corresponding eigenvectors. Then, the RR estimator can be written as

$$\hat{\beta}_{RR} = X(\Lambda + kI_r)^{-1}Z^Ty.$$  \hfill (5.2.19)

Now, let $\alpha = V^T\beta$ (so that $\beta = V\alpha$), then, the canonical form of the multiple regression model is

$$y = X\beta + \epsilon = Z\alpha + \epsilon$$  \hfill (5.2.20)

whence the OLS estimator of $\alpha$ is $\hat{\alpha}_{OLS} = (Z^TZ)^{-1}Z^Ty = \Lambda^{-1}V^Ts$, where $s = X^Ty$. It can be shown that

$$\hat{\alpha}_{RR}(k) = (\Lambda + kI_r)^{-1}\Lambda\hat{\alpha}_{OLS}.$$  \hfill (5.2.21)

Then, the $j$th component of $\hat{\alpha}_{RR}(k)$ is given by

$$\hat{\alpha}_{RR,j}(k) = \left(\frac{\lambda_j}{\lambda_j + k}\right)\hat{\alpha}_{OLS,j} = f_k(\lambda_j)\hat{\alpha}_{OLS,j},$$  \hfill (5.2.22)

where $0 \leq f_k(\lambda_j) \leq 1$, $i = 1, 2, \ldots, p$. For $k > 0$, $\hat{\alpha}_{RR,j} < \hat{\alpha}_{OLS,j}$, so that $\hat{\alpha}_{RR,j}(k)$ shrinks $\hat{\alpha}_{OLS,j}$ toward zero. Finally, by setting $\hat{\beta}_{RR}(k) = V\hat{\alpha}_{RR}(k)$ the canonical system is transformed back into the original coordinate system (Izenman 2008).

Separate multiple RR performs a separate RR on each individual response. Therefore, the regression coefficient estimates are given by

$$\hat{\beta}_{RR,j} = (X^TX + \lambda I)^{-1}X^Ty_j$$  \hfill (5.2.23)

where $y_j$ is the $N$-dimensional vector of measured values for the $j$th response, and $\lambda_j$ is the associated ridge parameter. Equation (5.2.23) biases the OLS coefficient estimates toward smaller absolute values and discourages dispersion among their values. The ridge parameter $\lambda_j$ regulates the strength of this effect differently for each response. The value of $\lambda_j$ is estimated through model selection techniques such as cross-validation, generalized cross-validation, minimum descriptive length, Bayesian information criterion, and Mallows’s Cp (Breiman & Friedman 1997, Frank & Friedman 1993). Although the uniresponse multiple RR approach ignores the correlation structure of the response variables, it can
provide considerably more accurate estimates with respect to OLS (Breiman & Friedman 1997).

5.2.4 The Lasso

The Lasso (Least absolute shrinkage and selection operator) is a constrained OLS minimization problem that estimate \( \beta \) by minimizing

\[
\phi(\beta) = (y - X\beta)^T(y - X\beta) + \lambda \sum_{i=1}^{p} |\beta_i|.
\] (5.2.24)

The constraint \( \sum_{i} |\beta_i| \leq c \) generates a diamond-shape penalty function. This problem can be solved using complicated quadratic programming methods subject to linear inequality constraints. The Lasso has a number of desirable features that have made it a popular regression algorithm. It is, at the same time, a shrinkage estimator of \( \beta_{\text{OLS}} \) (OLS coefficients are shrunk towards the origin) and a variable-selection technique (Izenman 2008). It performs a kind of continuous subset selection (Hastie et al. 2009). The value of \( c \) controls both the amount of shrinkage and the number of nonzero coefficients. In particular, a smaller value of \( c \) produce a smaller subset of nonzero coefficients (Izenman 2008). Typically, profiles of Lasso coefficients are obtained by plotting the coefficients versus the standardized tuning parameter values \( s = c/\sum_{j} |\hat{\beta}_j| \) obtained varying \( c \). The Lasso coefficients coincide to the OLS estimates when \( c = 1.0 \), and they decrease to 0 as \( c \to 0 \) (Hastie et al. 2009). The entire Lasso sequence of paths can be generated by a slight modification of the LARS algorithm, which is a procedure that efficiently combine Lasso, Forward-Stagewise and LAR algorithms. The latter stands for least-angle regression and is an automatic variable-selection method that improves upon Forward Selection in multiple regression (Izenman 2008).

5.2.5 Elastic net

If the explanatory variables are strongly correlated, both Lasso and RR do not work properly. The Lasso penalty is somewhat indifferent to the choice among a set of strong but correlated variables, while RR tends to shrink the coefficients of the correlated variables
toward each other. The elastic net penalty

$$\sum_{i=1}^{p} \left( \alpha |\beta_i| + (1 - \alpha) \beta_i^2 \right)$$

is a compromise between the penalty used for Lasso and RR. The second term encourages highly correlated features to be averaged, while the first term encourages a sparse solution in the coefficients of these averaged features. The elastic net penalty can be used with any linear model, both for regression and classification (Hastie et al. 2009).

### 5.2.6 OLS, RR, PCR, and PLSR: a comparison

RR, PCR and PLSR can be recast into a common framework. Indeed, all these techniques belong to the class of biased methods and their aim is to bias the solution coefficient vector $\hat{\beta}_{OLS}$ away from directions for which the projected sample predictor variables have small spread (Izenman 2008). This is seen to control the variance of the estimate (Frank & Friedman 1993). The degree of the bias is regulated by the value of the model-selection parameter, namely the ridge parameter $\lambda$ in RR, and the number $M$ of used components in PCR and PLSR (Frank & Friedman 1993). For RR, setting $\lambda = 0$ yields the unbiased OLS solution, whereas $\lambda > 0$ introduces increasing bias toward larger values of $\text{var}(\beta^T X)$. If $M = p$, PCR and PLSR produce an unbiased OLS solution, whereas for $M < p$, bias is introduced. The smaller the value of $M$, the larger the bias. A biased estimator can be expressed in the following general form

$$\hat{\beta} = \sum_{i} f(\lambda_i) \lambda_i^{-1} v_i v_i^T s$$

(5.2.26)

where $f(\lambda_i)$ is the $i$th shrinkage factor, $v_i$ is the eigenvector associated to the $i$th largest eigenvalue $\lambda_i$ of $X^T X$, and $s = X^T y$. It is shown that for a $M$-component PCR the shrinkage factor is $f(\lambda_i) = 1$ if $i \leq M$, and 0 otherwise; for a $M$-component PLSR, $f(\lambda_i)$ is a polynomial of degree $M$; and for RR with ridge parameter $k > 0$, $f(\lambda_i) = f_k(\lambda_i) = \lambda_i / (\lambda_i + k)$ (Izenman 2008).
OLS, PCR, PLSR and RR can be compared by considering the regression procedure as a two-step process. The first step consists in defining a \( M \)-dimensional subspace of \( p \)-dimensional Euclidean space. The second step consists in performing the regression under the restriction that the coefficient vector \( \beta \) lies in that subspace, namely

\[
\beta = \sum_{m=1}^{M} \beta_m c_m,
\]

where \( c_m \) are unit vectors that span the prescribed subspace with \( c_m^T c_m = 1 \). OLS, PCR, PLSR and RR differ in terms of the way in which they define the subspace \( c_1, \ldots, c_M \) and the manner in which the (constrained) regression is performed. In particular, in OLS the subspace is defined by the vector that maximizes the squared sample correlation between the response and the corresponding linear combination of the predictor variables, namely

\[
c_{\text{OLS}} = \arg \max_{c^T c = 1} \text{corr}^2(y, c^T X),
\]

The OLS solution is then a simple least squares regression of \( y \) on \( c_{\text{OLS}}^T X \). In RR the subspace is defined according to the following criterion

\[
c_{\text{RR}} = \arg \max_{c^T c = 1} \frac{\text{corr}^2(y, c^T X) \cdot \text{var}(c^T X)}{\text{var}(c^T X) + \lambda},
\]

where \( \lambda \) is the ridge parameter. The ridge solution is then taken to be a (shrinking) ridge regression of \( y \) on \( c_{\text{RR}}^T X \). PCR defines a sequence of \( M \)-dimensional subspaces each spanned by the first \( M \) eigenvectors of \( A \) in (5.2.5). Thus each \( c_m \) is the solution to

\[
c_{\text{PCR}}^{(M)} = \arg \max_{\begin{subarray}{c} c^T A c = 0 \\ c^T c = 1 \\ \end{subarray}} \text{var}(c^T X).
\]

The first constraint in equation (5.2.30), which is referred to as \( A \) orthogonality, ensures that the linear combinations associated with the different solution vectors are uncorrelated over the training sample

\[
\text{corr}(c_m^T X, c_l^T X) = 0, \quad m \neq l.
\]

As a consequence of this and the criterion (5.2.30), they also turn out to be orthogonal, that is \( c_m^T c_l = 0 \), for \( m \neq l \). The \( M \)th PCR model is given by a least squares regression
of the response on the $M$ linear combinations $c_m^T X$, $m = 1, \ldots, M$. PLSR regression also produces a sequence of $M$-dimensional subspaces spanned by successive unit vectors, and then the $M$th PLSR solution is obtained by a least squares fit of the response onto the corresponding $M$-linear combinations in a strategy similar to PCR. PLSR and PCR differ in the criterion used to define the vectors that span the $M$-dimensional subspace and hence the corresponding linear combinations. The criterion that gives rise to PLSR is

$$c^{(M)}_{\text{PLSR}} = \arg \max_{\{c^T M_{c} = 0\} \cap c^T c = 1} \frac{\text{corr}^2(y, c^T X) \text{var}(c^T X)}{c^T c}.$$  \hspace{1cm} (5.2.32)

As with PCR the vectors $c^{(M)}_{\text{PLSR}}$ are constrained to be mutually orthogonal so that the corresponding linear combinations are uncorrelated over the training sample (5.2.31). This causes the $M$-dimensional least squares fit to be equivalent to the sum of $M$ separate regressions on each linear combination separately, as with PCR. Unlike PCR, however, the $c^{(M)}_{\text{PLSR}}$ are not orthogonal owing to the different criterion (5.2.32) used to obtain them.

### 5.3 Multivariate multiple regression

Multivariate multiple regression employs shrinkage and selection strategies that exploit correlations in the different responses. Combining responses is at the heart of canonical correlation analysis (CCA), a data reduction technique developed for the multiple output case. CCA constitutes the foundations of the majority of the multivariate multiple regression techniques and is described further below (Hastie et al. 2009, Breiman & Friedman 1997).

Assume that $X$ and $Y$ are jointly distributed with mean vector given by

$$E \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}$$ \hspace{1cm} (5.3.1)

and covariance matrix given by

$$E \begin{pmatrix} X - \mu_X \\ X - \mu_X \end{pmatrix}^T \begin{pmatrix} X - \mu_X \\ Y - \mu_Y \end{pmatrix} = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix}$$ \hspace{1cm} (5.3.2)
CCA aims at finding the vectors \( v \) and \( t \) that maximize the correlation

\[
\text{corr}(\xi, \omega) = \text{corr}(t^T Y, v^T X) = v^T \Sigma_{XY} t,
\]

where

\[
v^T X = v_1 X_1 + v_2 X_2 + \ldots + v_p X_p,
\]
\[
t^T Y = t_1 Y_1 + t_2 Y_2 + \ldots + t_q Y_q.
\]

To achieve this, set

\[
f(v, t) = v^T \Sigma_{XY} t - \frac{1}{2} \lambda (v^T \Sigma_{XX} v - 1) - \frac{1}{2} \mu (t^T \Sigma_{YY} t - 1),
\]

where \( \lambda \) and \( \mu \) are Langrangian multipliers. The function \( f(v, t) \) in (5.3.6) is then differentiated and its partial derivatives are set equal to zero, namely

\[
\frac{\partial f}{\partial v} = \Sigma_{XY} t - \lambda \Sigma_{XX} v = 0,
\]
\[
\frac{\partial f}{\partial t} = \Sigma_{YX} v - \mu \Sigma_{YY} t = 0.
\]

Multiplying (5.3.7) on the left by \( v^T \) and (5.3.8) on the left by \( t^T \), the following equations are obtained

\[
v^T \Sigma_{XY} t - \lambda v^T \Sigma_{XX} v = 0,
\]
\[
t^T \Sigma_{YX} v - \mu t^T \Sigma_{YY} t = 0.
\]

whence, the correlation between \( \xi \) and \( \omega \) satisfies

\[
v^T \Sigma_{XY} t = \lambda = \mu.
\]

Then, the terms in equation (5.3.9) are rearranged and \( \lambda \) is substituted for \( \mu \) into equation (5.3.10) so that the subsequent equations are determined

\[- \lambda \Sigma_{XX} v + \Sigma_{XY} t = 0,
\]
\[ \Sigma_{YX}v - \lambda \Sigma_{YY}t = 0. \]  
(5.3.13)

Premultiplying (5.3.12) by \( \Sigma_{YX} \Sigma_{XX}^{-1} \)

\[ \Sigma_{YX} \Sigma_{XX}^{-1} (-\lambda \Sigma_{XX}v + \Sigma_{XY}t) = 0 \]

\[ -\lambda \Sigma_{YX}v + \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY}t = 0 \]  
(5.3.14)

and substituting (5.3.13), \( \Sigma_{YX}v = \lambda \Sigma_{YY}t \), into (5.3.14)

\[ -\lambda \Sigma_{YX}v + \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY}t = 0. \]

This gives

\[ (\Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} - \lambda^2 \Sigma_{YY})t = 0 \]  
(5.3.15)

which is equivalent to

\[ (\Sigma_{XY}^{-1/2} \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1/2} - \lambda^2 I_K)t = 0. \]  
(5.3.16)

For there to be a nontrivial solution to this equation, the following determinant has to be zero:

\[ |\Sigma_{YX}^{-1/2} \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1/2} - \lambda^2 I_K| = 0. \]  
(5.3.17)

It can be shown that the determinant in (5.3.17) is a polynomial in \( \lambda^2 \) of degree \( K \), having \( K = \min(p, q) \) real roots, \( \lambda_1^2 \geq \lambda_2^2 \geq \ldots \geq \lambda_K^2 \geq 0 \), say, which are the eigenvalues of

\[ Q = \Sigma_{YY}^{-1/2} \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1/2} \]  
(5.3.18)

with associated eigenvectors \( e_1, e_2, \ldots, e_K \). The maximal correlation between \( \xi \) and \( \omega \) would, therefore, be achieved if we took \( \lambda = \lambda_1 \), the largest eigenvalue of \( Q \). The resultant choice of coefficients \( v \) and \( t \) of \( \xi \) and \( \omega \), respectively, are given by the vectors

\[ v_1 = \Sigma_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1/2} e_1, \quad t_1 = \Sigma_{YY}^{-1/2} e_1. \]  
(5.3.19)
In other words, the first pair of canonical variates is given by \((\xi_1, \omega_1)\), where \(\xi_1 = v_1^T X\) and \(\omega_1 = t_1^T Y\), and their correlation is \(\text{corr}(\xi_1, \omega_1) = v_1^T \Sigma_{XY} t_1 = \lambda_1\). Given \((\xi_1, \omega_1)\), let \(\xi = v^T X\) and \(\omega = t^T Y\) denote a second pair of arbitrary linear projections with unit variances. We require \((\xi, \omega)\) to have maximal correlation among all such linear combinations of \(X\) and \(Y\), respectively, which are also uncorrelated with \((\xi_1, \omega_1)\). This last condition translates into
\[
\Sigma_{XX} v_1 \Sigma_{XY} t_1 = 0.
\]
Furthermore, by (5.3.12) and (5.3.13), we require
\[
\text{corr}(\xi, \omega_1) = v^T \Sigma_{XY} t_1 = \lambda_1 v^T \Sigma_{XX} v_1 = 0
\]
and
\[
\text{corr}(\omega, \xi_1) = t^T \Sigma_{YX} v_1 = \lambda_1 t^T \Sigma_{YY} t_1 = 0.
\]

The vectors \(v\) and \(t\) are chosen to maximize (5.3.3) subject to conditions (5.3.20) and (5.3.21). Set
\[
f(v, t) = v^T \Sigma_{XY} t - \frac{1}{2} \lambda (v^T \Sigma_{XX} v - 1) - \frac{1}{2} \mu (t^T \Sigma_{YY} t - 1) + \eta v^T \Sigma_{XX} v_1 + \nu t^T \Sigma_{YY} t_1,
\]
where \(\lambda\), \(\mu\), \(\eta\), and \(\nu\) are Lagrangian multipliers. Differentiate \(f(v, t)\) with respect to \(v\) and \(t\), and then set both partial derivatives equal to zero:
\[
\frac{\partial f}{\partial v} = \Sigma_{XY} t - \lambda \Sigma_{XX} v + \eta \Sigma_{XX} v_1 = 0,
\]
and
\[
\frac{\partial f}{\partial t} = \Sigma_{YX} v - \mu \Sigma_{YY} t + \nu \Sigma_{YY} t_1 = 0.
\]

Multiplying equation (5.3.23) on the left by \(v^T\) and equation (5.3.24) on the left by \(t^T\), and considering the conditions in equation (5.3.20) and (5.3.21), we obtain equations (5.3.12) and (5.3.13), respectively. Therefore, the second pair of canonical variates is \((\xi_2, \omega_2)\), where
\[
v_2 = \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2} e_2, \quad t_2 = \Sigma_{YY}^{-1/2} e_2,
\]
and their correlation is \(\text{corr}(\xi_2, \omega_2) = v_2^T \Sigma_{XY} t_2 = \lambda_2\).

This sequential procedure is continued, deriving eigenvalues and eigenvectors, until no further solutions can be found. This returns sets of coefficients for the pairs of canonical variates, \((\xi_1, \omega_1), (\xi_2, \omega_2), \ldots, (\xi_K, \omega_K)\), \(K = \min(p, q)\), where the \(i\)th pair of canonical variates \(\xi_i, \omega_i\) is obtained by choosing the coefficients \(v_i\) and \(t_i\) such that \(\xi_i, \omega_i\) has the largest
correlation among all pairs of linear combinations of $X$ and $Y$ that are also uncorrelated with all previously derived pairs, $(\xi_j, \omega_j)$, $j = 1, 2, \ldots, i - 1$ (Izenman 2008).

### 5.3.1 Multivariate ridge regression

In multivariate multiple RR only one ridge coefficient is estimated according to

$$\hat{\beta}_{RR} = (X^TX + \lambda I)^{-1}X^Ty.$$  (5.3.26)

Here, $\lambda$ is the unique ridge parameter value and all of the $q$ outputs are used to estimate it. The criterion in equation (5.3.26) differs from the criterion in equation (5.2.23) in that the latter allows different amounts of regularization to be applied to different outcomes and require estimation of $q$ separate regularization parameters $\lambda_1, \ldots, \lambda_q$. In a small simulation study, Frank & Friedman (1993) showed that, surprisingly, the multivariate RR did not dramatically better than the corresponding separate procedure applied separately to the individual response, even though the situations were designed to be most favourable to the multiresponse methods.

### 5.3.2 Two-block partial least squares regression

Two-block partial least squares regression, also referred to as PLSR2, is the multivariate version of PLSR (see section 5.2.2) and is used when multiple outputs are investigated (Izenman 2008). With PLSR2, the responses matrix $Y$ and the predictors matrix $X$ are separately collected together into groups (blocks) which are then treated in a common manner (Frank & Friedman 1993). The PLSR2 algorithm is as follows:

1. initialize: $Y_0 \leftarrow Y$, $X_0 \leftarrow X$, $\hat{Y}_0 \leftarrow 0$

2. for $M = 1$ to $p$ do:
   - $u^T \leftarrow (1, 0, \ldots, 0)$
   - Loop until convergence
     - $w_M = \langle u^TY_{M-1}, X_{M-1} \rangle$
     - $u = \langle w_M^TX_{M-1}, Y_{M-1} \rangle$
• End loop
• \[ z_M = w_M^T X_{M-1} \]
• \[ r_M = \frac{\langle Y_{M-1}, z_M \rangle}{\langle z_M, z_M \rangle} z_M \]
• \[ \bar{Y}_M = \bar{Y}_{M-1} + r_M \]
• \[ Y_M = Y_{M-1} - r_M \]
• \[ X_M = X_{M-1} - \left( \frac{\langle z_M, X_{M-1} \rangle}{\langle z_M, z_M \rangle} z_M \right) \]
• if \( \langle X_M^T, X_M \rangle = 0 \) then Exit

It is worth noting that PLSR2 does not generate \( q \) separate PLSR models (one for each response). Indeed, in the inner loop of the PLSR2 algorithm a single covariance vector \( w_M \) is computed for all the responses. This single \( w_M \) is then used to update all of the models \( \bar{Y}_M \) and all of the response residuals to obtain \( Y_M \). Furthermore, a single set of \( X \) residuals \( X_M \) is maintained using the single covariance vector \( w_M \). The inner loop in the PLSR2 algorithm is an iterative procedure for finding linear combinations of the response residuals \( u^T Y_{M-1} \) and the predictor residuals \( w_M^T X_{M-1} \) that have maximal joint covariance. This algorithm starts with an arbitrary coefficient vector \( u = (1, 0, \ldots, 0) \). After convergence of the inner loop, the resulting \( X \) residual linear combination covariance vector, \( w_M \), is used for all updates. Similarly to PLSR, PLSR2 algorithm produces \( M \) models, spanning a full spectrum of solutions from the sample means for \( M = 0 \) to the OLS solutions for \( M = p \). The number of components \( M \) is considered a meta parameter of the procedure and is selected through CV.

PLSR2 can be viewed as a penalized canonical correlation criterion. Similar to the two-step procedure adopted in section 5.2.6, a \( M \)-dimensional subspace of a \( p \)-dimensional Euclidean space is first defined as being spanned by the unit vectors \( c_m, m = 1, \ldots, M \). Then \( q \)-OLS regressions are performed under the constraints that the OLS solution coefficient vectors \( \hat{\beta}_j, j = 1, \ldots, q \) \( \beta = (\beta_1, \ldots, \beta_q) \) in (5.2.2) lie in that subspace,

\[
\hat{\beta}_j = \sum_{m=1}^M \hat{\beta}_{mj} c_m. \tag{5.3.27}
\]
A regression procedure is then prescribed by defining the ordered sequence of unit vectors $c_m, m = 1, \ldots, M$. Defining each of these unit vectors to be the solution to

$$
c_m = \arg \max_{\var{c}_i} \arg \max_{\var{u}_i} \{ \text{var}(u^TY) \text{corr}^2[(u^TY), (c^TX)] \cdot \text{var}(c^TX) \} \tag{5.3.28}
$$

gives (in this framework) the same sequence of models $\bar{Y}_m$ obtained using the two-block PLSR (PLSR2). The constraints on $c_m$ require them to be unit vectors and to be $A$ orthogonal so that the corresponding linear combinations are uncorrelated. Using the middle factor $\text{corr}^2[(u^TY), (c^TX)]$ alone for the criterion would give rise to standard canonical correlation analysis, producing a sequence of uncorrelated linear combinations $c_m^TX$ that maximally predict the corresponding optimal linear combinations of the responses, $u^TY$.

The PLSR2 criterion is seen to include two additional factors $\text{var}(u^TY)$ and $\text{var}(c^TX)$ that serve as penalties to bias the solutions away from low spread directions in both the $X$ and $Y$ spaces. The penalty imposed on the predictor variable linear combination coefficient vector $c$ is the same as that used for single response PLSR. This mainly serves to control the variance of the estimated model. The introduction of the $Y$-space penalty factor, along with optimization with respect to its associated linear combination coefficient vector $u$, serves to place an additional penalty on the $X$-linear combination coefficient vectors $c_m$ that define the sequence of PLSR models. Indeed, they are not only biased away from low (data) spread directions in the predictor variable space but also toward $X$ directions that preferentially predict the high spread directions in the response-variable space (Frank & Friedman 1993).

### 5.3.3 Reduced rank regression

Reduced rank regression (RRR) has been introduced by Izenman (1975). It is characterized by a rank constraint on the estimated regression coefficients matrix, that brings a true multivariate feature into the model (Izenman 2008). Consider the multivariate linear regression model

$$
Y = \mu + CX + \epsilon \tag{5.3.29}
$$
where $\mu$ and $C$ are unknown regression parameters and $e$ is the unobservable error variate. The error variate has mean $E(e) = 0$, covariance matrix $\text{cov}(e) = E(ee^T) = \Sigma_{ee}$ and is distributed independently of $X$. In equation (5.3.29) the rank of the regression coefficient matrix $C$ is allowed to be deficient, that is

$$\text{rank}(C) = t \leq \min(p,q) \quad (5.3.30)$$

The reduced-rank condition in equation (5.3.30) on the regression coefficient matrix $C$ implies that there may be a number of linear constraints on the set of regression coefficients in the model (Izenman 2008). When $C$ has reduced-rank $t$, then, there exist two (nonunique) full-rank matrices, a $q \times t$ matrix $A$ and a $t \times p$ matrix $B$, such that $C = AB$. The nonuniqueness occurs because we can always find a nonsingular $t \times t$-matrix $T$ such that $C = (AT)(T^{-1}B) = DE$, which gives a different decomposition of $C$. Then, the model (5.3.30) can be written as

$$Y = \mu + ABX + \epsilon. \quad (5.3.31)$$

The unknown parameters $\mu$, $A$, and $B$ (and, hence, $C$) are estimated by minimizing the following weighted sum of squares criterion that depends on the value of $t$:

$$W(t) = E \left\{ (Y - \mu - ABX)^T \Gamma (Y - \mu - ABX) \right\}. \quad (5.3.32)$$

Here, $\Gamma$ is a positive-definite symmetric matrix of weights. The criterion in equation (5.3.32) is minimized by taking $\mu$, $A$ and $B$ to be the following functions of $t$,

$$\mu^{(t)} = \mu_Y - A^{(t)}B^{(t)}\mu_X, \quad (5.3.33)$$

$$A^{(t)} = \Gamma^{-1/2}V_t, \quad (5.3.34)$$

$$B^{(t)} = V_t^T\Gamma^{1/2}\Sigma_{YY}\Sigma_{XX}^{-1}, \quad (5.3.35)$$

where $\mu_X$ and $\mu_Y$ are as in equation (5.3.1), and $V_t = (v_1, \ldots, v_t)$ is a matrix whose $j$th column $v_j$ is the eigenvector associated with the $j$th largest eigenvalue $\lambda_j$ of the symmetric matrix

$$\Gamma^{1/2}\Sigma_{YY}\Sigma_{XX}^{-1}\Sigma_{XY}\Gamma^{1/2}. \quad (5.3.36)$$
The minimizing $C$ with reduced-rank $t$ is given by

$$C(t) = \Gamma^{-1/2}(\sum_{j=1}^{t} v_j v_j^T)\Gamma^{1/2}\Sigma_{YX}\Sigma_{XX}^{-1}$$  \hspace{1cm} (5.3.37)

and is called the \textit{reduced-rank regression coefficient matrix} with rank $t$ and weight matrix $\Gamma$ (Izenman 2008). The RRR coefficient matrix $C$ in equation (5.3.37) can be expressed also as

$$C(t) = A(t)\hat{\beta}_{OLS}$$  \hspace{1cm} (5.3.38)

where $\hat{\beta}_{OLS} \in \mathbb{R}^{q \times p}$ is the matrix of OLS estimates and $A(t) \in \mathbb{R}^{q \times q}$ is a reduced rank ‘shrinking’ matrix. The matrix $A(t)$ can be expressed as

$$A(t) = \hat{T}^{-1}I(t)\hat{T}$$  \hspace{1cm} (5.3.39)

where $T$ is the matrix of sample canonical co-ordinates and

$$I = \text{diag}\{1(j \leq t)\} \quad j = 1,\ldots,q.$$  \hspace{1cm} (5.3.40)

Here, the rank value $t$ is regarded as a regularization parameter of the procedure and its value is estimated through model selection (Breiman & Friedman 1997).

It is worth noting that PCA and CCA are special cases of RRR. In particular, PCA is obtained by setting $X \equiv Y$ (and $p = q$), whereas CCA are obtained by setting $\Gamma = \Sigma_{YY}^{-1}$.

\subsection*{5.3.4 Curds and whey}

Curds and whey (C&W) is a multivariate regression technique proposed by Breiman & Friedman (1997). It constitutes a smooth version of RRR and takes advantage of correlations between the response variables to improve predictive accuracy (Hastie et al. 2009, Breiman & Friedman 1997). To explain the potential improvement in estimation, consider the following example described in Breiman & Friedman (1997). Suppose that the data are of the form $(x_1, y_1) \ldots (x_N, y_N)$ where, for $n = 1,\ldots,N$, $x_n = (x_{n1},\ldots,x_{np})^T$ and $y_n = (y_{n1},\ldots,y_{nq})^T$. Further suppose that $q = 2$, so that $Y = (Y_1, Y_2)$. Predictors for $Y_1$ and $Y_2$ can be obtained by means of separate regression on $X = (X_1,\ldots,N_p)$. Then, the $p \times 2$
matrix of parameters $\beta = (\beta_1, \beta_2)$ are estimated by $\hat{\beta}_1 = (\hat{\beta}_{11}, \ldots, \hat{\beta}_{1p})$ and $\hat{\beta}_2 = (\hat{\beta}_{21}, \ldots, \hat{\beta}_{2p})$, which are solutions to

$$
\hat{\beta}_1 = \arg \min_{\beta} \left\{ \sum_{n=1}^{N} (y_{n1} - \beta^T x_n)^2 \right\},
$$

$$
\hat{\beta}_2 = \arg \min_{\beta} \left\{ \sum_{n=1}^{N} (y_{n2} - \beta^T x_n)^2 \right\},
$$

where all variables have been centered. The prediction equations for $Y_1$ and $Y_2$ are

$$
\bar{y}_1(X) = \bar{y}_1 + \hat{\beta}_1^T (X - X),
$$

(5.3.41)

and

$$
\bar{y}_2(X) = \bar{y}_2 + \hat{\beta}_2^T (X - X),
$$

(5.3.42)

where $(\bar{y}_j, \bar{X})$ are the sample means before centering. Suppose that the unknown truth happens to be $y_{n1} = \beta_{10} + \beta^T x_n + \epsilon_{n1}$ and $y_{n2} = \beta_{20} + \beta^T x_n + \epsilon_{n2}$ where $\epsilon_{n1}$ and $\epsilon_{n2}$ are independent and identically distributed $\mathcal{N}(0, \sigma^2)$. Because of the same dependence on the predictor variables $(\beta^T x_n)$, $Y_1$ and $Y_2$ are correlated. For each of the two responses, accuracy can be improved by using the predictors

$$
\bar{y}_j = \bar{y}_j + \frac{1}{2}(\bar{y}_1 - \bar{y}_1) + \frac{1}{2}(\bar{y}_2 - \bar{y}_2) \quad (j = 1, 2),
$$

instead of $\bar{y}_1$ and $\bar{y}_2$, in equations (5.3.41) and (5.3.42), respectively. More generally, if there are $q$ correlated responses $y_1, \ldots, y_q$ a predictor that is more accurate than the ordinary OLS predictors $\bar{y}_j$, can be obtained by using a linear combination

$$
\bar{y}_j = \bar{y}_j + \sum_{k=1}^{q} \beta_{jk} (\bar{y}_k - \bar{y}_k), \quad j, k = 1, \ldots, q,
$$

(5.3.43)

of the ordinary OLS predictors

$$
\bar{y}_j = \bar{y}_j + \sum_{i=1}^{p} \alpha_{ji} (x_i - \bar{x}_i),
$$

(5.3.44)

$$
[\alpha_{ji}]_{i=1}^{p} = \arg \min_{\alpha_i} \left[ \sum_{n=1}^{N} \left\{ y_{nj} - \bar{y}_j - \sum_{i=1}^{p} \alpha_{ji} (x_{ni} - \bar{x}_i) \right\}^2 \right],
$$

(5.3.45)
rather than with the least squares predictors themselves. As note by the authors, equations (5.3.43) and (5.3.44) imply that the coefficients, but not the means, of the OLS estimates are modified.

Assuming that the response and predictor variables are all centred, equation (5.3.43) can be expressed in matrix notation as

$$\tilde{Y} = B\hat{Y},$$  \hspace{1cm} (5.3.46)

where $\tilde{Y}$ and $\hat{Y}$ are the $N \times q$ matrices with column vectors $\tilde{y}_1, \ldots, \tilde{y}_q$ and $\hat{y}_1, \ldots, \hat{y}_q$, respectively. $B = (b_1, \ldots, b_q)$ is a $q \times q$ shrinking matrix that transforms the OLS estimate $\hat{Y}$ to the biased estimate $\tilde{Y}$. The goal is to obtain an estimate of the optimal shrinking matrix $B^*$ whose elements are defined by

$$b^*_{jk} = \arg\min_{b_k} \left[ E \left( y_j - \sum_{k=1}^q b_k \hat{y}_k \right)^2 \right], \quad j, k = 1, \ldots, q.$$  \hspace{1cm} (5.3.47)

where the expected value is over the joint distribution $F(X, Y)$ of the data to be predicted. The solution to equation (5.3.47) is a least squares regression (through the origin) of each response $y_j$ on the (sample-based) OLS estimates $\hat{y}_j$, namely

$$B^* = E(Y\tilde{Y}^T)E(\tilde{Y}\tilde{Y}^T)^{-1}.$$  \hspace{1cm} (5.3.48)

As noted by Breiman & Friedman (1997), the use of $B^*$ in equation (5.3.48) in equation (5.3.46) will result in a reduced mean-squared error for each response

$$E[y_j - (B^*\tilde{Y})_j]^2 \leq E[y_j - \tilde{y}_j]^2, \quad j = 1, \ldots, q.$$  \hspace{1cm} (5.3.49)

The equality in equation (5.3.49) is obtained only if $B^* = I_q$, where $I_q$ is the $q \times q$ identity matrix. Therefore, the expected squared error loss will be reduced for every response individually, rather than only with respect to a loss criterion involving all of the responses (such as weighted average quadratic loss).
In C&W the shrinkage matrix $B^*$ is obtained by means of a CCA. As previously described, CCA seeks $K = \min(p,q)$ pairs of linear combinations according to the following criterion

$$(t_k, v_k) = \arg \max \left\{ \frac{\text{corr}(t_k^T Y, v_k^T X)}{\text{corr}(t_k^T Y, v_k^T X) = 0} \right\},$$

(5.3.50)

where, each successive pair of linear combinations is constrained to be uncorrelated with the previous pairs. Furthermore, the linear combinations are all standardized to have unit variances

$$E(t_k^T Y)^2 = E(v_k^T X)^2 = 1, \quad k = 1, \ldots, K.$$  

(5.3.51)

The solution to equations (5.3.50) and (5.3.51) for $t_k$ are obtained from eigenanalysis of the $q \times q$ matrix $Q$ in equation (5.3.18) which can be expressed as

$$Q = T^T C^2 T^{-T}.$$  

(5.3.52)

Here, $T$ is a $q \times q$ matrix whose rows (eigenvectors) are the $Y$ canonical co-ordinates $t_k$ and $C^2$ is a diagonal matrix

$$C^2 = \text{diag}(c_1^2, \ldots, c_K^2)$$  

(5.3.53)

containing the squared canonical correlations $c_k = \text{corr}(t_k^T Y, v_k^T X)$ (Breiman & Friedman 1997). Assume that each response is a linear function of the predictors with additive IID error, namely

$$Y_j = \sum_{i=1}^{p} \beta_{ji} X_i + \epsilon_j, \quad j = 1, \ldots, q.$$  

(5.3.54)

In matrix notation

$$Y = \beta X + \epsilon.$$  

(5.3.55)

The predictors $X = (X_1, \ldots, X_p)^T$ and the errors $\epsilon = (\epsilon_1, \ldots, \epsilon_q)^T$ are random samples with distribution $F_X(X)$ and $F_\epsilon(\epsilon)$, respectively. The errors are assumed to be independent of the predictor variables, that is their joint distribution is given by

$$F(X, \epsilon) = F_X(X)F_\epsilon(\epsilon).$$  

(5.3.56)
Further assume that
\[
E(X) = E(\epsilon) = 0,
\]
\[
E(XX^T) = V \in \mathbb{R}^{p \times p},
\]
\[
E(\epsilon \epsilon^T) = \Sigma \in \mathbb{R}^{q \times q}
\]
where the expected values are over the joint distribution in equation (5.3.56). In this setting the errors are assumed to be independent between observations, but possibly correlated among the responses for each observation (Breiman & Friedman 1997). For simplicity, it is further assumed that the sample means and covariance matrix of the predictor variables are the same as that of the population distribution. Assume that the predictor and response data matrix \( X \) and \( Y \) are available. Then, under model (5.3.54)-(5.3.57), the matrix \( Q \) is given by

\[
Q = \Sigma^{-1}_{YY} - \Sigma^{-1}_{XX} \Sigma^{-1}_{XY}
\]
\[
= E(YY^T)^{-1} E(YX^T) E(XX^T)^{-1} E(XY^T)
\]
\[
= E((\beta X + \epsilon) (\beta X + \epsilon)^T)^{-1} E((\beta X + \epsilon)X^T) E(XX^T)^{-1} E(X(\beta X + \epsilon)^T)
\]
\[
= E(\beta XX^T \beta^T + \beta X \epsilon^T + \epsilon X^T \beta + \epsilon \epsilon^T)^{-1} E(\beta XX^T + \epsilon X^T) E(XX^T)^{-1} E(XX^T) \beta
\]
\[
= (\beta E(XX^T) \beta^T + E(\epsilon \epsilon^T))^{-1} \beta E(XX^T) E(XX^T)^{-1} E(XX^T) \beta
\]
\[
= \beta V \beta^T + \Sigma^{-1} \beta V V^{-1} V \beta^T
\]
\[
= (F + \Sigma)^{-1} F
\]
\[
= (F + \Sigma)^{-1} F F^{-1}
\]
\[
= (I_q + R^T)^{-1},
\]
(5.3.58)

where \( F = E[f(X)f^T(X)] = \beta V \beta^T \) and

\[
R = \Sigma F^{-1}
\]
(5.3.59)
is the noise-to-signal matrix. Breiman & Friedman (1997) show that, under model (3.2.52)-(5.3.57), the optimal shrinking matrix $B^*$ is given by

$$B^* = (I_q + rR)^{-1}$$  \hspace{1cm} (5.3.60)

where

$$r = \frac{p}{N}$$  \hspace{1cm} (5.3.61)

is the ratio of the number of predictors to the training sample size. As a consequence, $B^*$ is determined by the noise-to-signal structure in the response space, that is by the matrix $R$ in equation (5.3.59) which can be obtained from equation (5.3.58) as follows

$$Q = (I_q + R^T)^{-1}$$

$$Q(I_q + R^T) = I$$

$$Q + QR^T = I$$

$$QR^T = I - Q$$

$$R^T = (I - Q)Q^{-1}$$

$$R^T = Q^{-1} - I_q$$

$$R = (Q^{-1} - I_q)^T.$$

$R$ into equation (5.3.59) can then be substituted into equation (5.3.60) to obtain

$$B^* = (I_q + rR)^{-1}$$

$$= (I_q + r(Q^{-1} - I_q)^T)^{-1}$$

$$= (I_q + rQ^{-T} - rI_q)^{-1}$$

$$= ((1 - r)I_q + rQ^{-T})^{-1}.$$  \hspace{1cm} (5.3.62)

By substituting $Q = T^T C^2 T^{-T}$ (equation (5.3.52)) into equation (5.3.62) we obtain

$$B^* = ((1 - r)I_q + r(T^T C^2 T^{-T})^{-T})^{-1}$$

$$= ((1 - r)I_q + rT(C^2)^{-1}T^{-1})^{-1}. $$
This shows that \( B^* \) is diagonal in the \( Y \) canonical co-ordinate system (5.3.52), that is

\[
B^* = T^{-1}DT, \quad D = \text{diag}(d_1, \ldots, d_q)
\]  

(5.3.63)

with

\[
d_j = \left\{ \frac{(1-r) + r}{c_j^2} \right\}^{-1} = \left\{ \frac{c_j^2(1-r) + r}{c_j^2} \right\}^{-1} = \left\{ \frac{c_j^2 - rc_j^2 + r}{c_j^2} \right\}^{-1} = \frac{c_j^2}{c_j^2 + r(1-c_j^2)}.
\]  

(5.3.64)

Here, \( d_j \) is set to

\[
d_j \leftarrow \max(d_j, 0)
\]  

(5.3.65)

to perform ‘positive part’ shrinkage (Breiman & Friedman 1997). Substituting equation (5.3.63) into equation (5.3.46) we obtain that

\[
T \tilde{Y} = D(T \tilde{Y})
\]  

(5.3.66)

so that equation (5.3.46) reduces to separate proportional shrinkage of each OLS solution in the \( Y \) canonical co-ordinate system. This corresponds to:

1. transform \( Y \) to the canonical co-ordinate system, \( Y' = TY \).
2. perform a separate OLS regression of each \( Y_j' \) on \( X \) (\( j = 1, \ldots, q \)), obtaining \( \hat{Y}_j' \).
3. separately scale (shrink) each \( \hat{Y}_j' \) by the factor \( d_j \) in equation (5.3.64), obtaining \( \tilde{Y} = T^{-1} \hat{Y}' \).

(Breiman & Friedman 1997). To estimate \( B^* \) a sample-based estimate of \( Q \) is required. A natural choice is the ‘plug-in’ estimate

\[
\hat{Q} = (Y^T Y)^{-1} Y^T X (Y^T Y)^{-1} X^T Y = \tilde{T}^T \tilde{C}^2 \tilde{T}^{-T}
\]  

(5.3.67)
where $Y$ and $X$ are the centered data matrices. The derived shrinkage matrix takes the form

$$ B = \hat{T}^{-1} \hat{D} \hat{T} $$

(5.3.68)

with the elements of $\hat{D}$ given by

$$ \hat{d}_j = \frac{\hat{c}^2_j}{\hat{c}^2_j + r(1 - \hat{c}^2_j)}, \quad j = 1, \ldots, q. $$

(5.3.69)

However, the use of $\hat{Q}$ in equation (5.3.67), does not provide enough shrinkage because the sample canonical correlations $\hat{c}_j$ in equation (5.3.69) overestimate their corresponding population values $c_j$ in equation (5.3.53). Indeed, if the sample-based estimates are used in place of the correct (unbiased) population values, less shrinkage is obtained. This derives from the fact that the same sample is used to estimate both OLS solution and its goodness of fit. The proper (less biased) amount of shrinkage can be estimated through cross-validation (CV) or generalized cross-validation (GCV). In what follows, CV and GCV are introduced and subsequently C&W-CV and C&W-CV are described.

**CV- and GCV-based C&W**

CV is the simplest and most widely used method for estimating prediction error, that is the prediction accuracy of a model (Hastie et al. 2009). Suppose we have a learning set $L$. In regression problems, we first fit a model to the learning set $L$ and then use that model to predict the output values of either $L$ (given input values from $L$) or the test set $T$ (given input values from $T$). Prediction error is the mean (computed only over the appropriate data set) of the squared errors of prediction. If we average over $L$, the prediction error is called *regression learning error*, while if we average over $T$, the prediction error is called *regression test error* (Izenman 2008). Ideally, if we have enough data, we would set aside a validation set and use it to assess the performance of the prediction model (Hastie et al. 2009). However, if the learning set $L$ is moderately sized, it is preferable to use data-splitting methods for estimating test error, so that we avoid a waste of good data (Izenman 2008). $V$-fold CV performs this splitting as follows: randomly divide the entire data set into $V$ nonoverlapping groups of roughly equal size, remove one of the groups and fits the model using the combined data from the other $V - 1$ groups (which forms
the learning set), use the omitted group as the test set, predict its output values using
the fitted model and compute the prediction error for the omitted group, repeat this
procedure \( V \) times, each time removing a different group. The resulting \( V \) prediction
errors are finally averaged to estimate the test error. The number of groups \( V \) can be any
number from 2 to the sample size. Typical choices of \( V \) are 5 or 10. If \( V \) is set equal to
the sample size \( N \), leave-one-out cross-validation is obtained (Hastie et al. 2009). The
cross-validation estimate of prediction error is

\[
CV(\hat{f}) = \frac{1}{N} \sum_{n=1}^{N} L(y_n, \hat{f}^{(-k)}(x_n))
\]

(5.3.70)

where \( N \) is the sample size, \( L(Y, \hat{f}) \) is the loss function for measuring error between \( Y \)
and \( \hat{f} \), and \( \hat{f}^{(-k)}(x) \) is the fitted function, computed with the \( k \)th part of the data removed
(Hastie et al. 2009). Typical choices of the loss function are

\[
L(Y, \hat{f}) = \begin{cases} 
(Y - \hat{f}(X))^2 & \text{squared error} \\
|Y - \hat{f}(X)| & \text{absolute error}
\end{cases}
\]

(5.3.71)

(Hastie et al. 2009). With \( V = N \) the cross-validation estimator is approximately unbi-
ased for the true (expected) prediction error, but can have high variance because the \( N \)
‘training sets’ are so similar to one another. The computational burden is also consider-
able, requiring \( N \) applications of the learning method. With \( V = 5 \), cross-validation has
lower variance but bias could be a problem, depending on how the performance of the
learning method varies with the size of the training set. Overall, five- or tenfold cross-
validation are recommended as a good compromise (Breiman & Spector 1992, Kohavi
1995). GCV provides a convinient approximation to leave-one-out CV, for linear fitting
under squared-error loss. A linear fitting method is one for which we can write

\[
\hat{y} = Hy.
\]

(5.3.72)

For many linear fitting methods,

\[
\frac{1}{N} \sum_{n=1}^{N} [y_n - \hat{f}^{(-n)}(x_n)]^2 = \frac{1}{N} \sum_{n=1}^{N} \left[ \frac{y_n - \hat{f}(x_n)}{1 - h_{nn}} \right]^2,
\]

(5.3.73)
where $h_{nn}$ is the $n$th diagonal element of $H$. The GCV approximation is

$$
\text{GCV}(\hat{f}) = \frac{1}{N} \sum_{n=1}^{N} \left[ \frac{y_n - \hat{f}(x_n)}{1 - \text{trace}(H)/N} \right]^2.
$$

(5.3.74)

where the quantity $\text{trace}(H)$ is the effective number of parameters. GCV can have a computational advantage in some settings, where the trace of $H$ can be computed more easily than the individual elements of $h_{nn}$. In smoothing problems, GCV can also alleviate the tendency of cross-validation to undersmooth (Hastie et al. 2009).

In the CV-based C&W procedure, the optimal shrinking matrix $B^*$ is obtained by regression of the responses $y_j$ on the (sample-based) OLS estimates $\hat{y}_j$ over (all future) data that are not part of the training sample. Each observation $(x_n, y_n)$ is (in turn) removed from the training sample and treated as a ‘future’ observation. Using cross-validation, the elements of the shrinking matrix $B$ are obtained by solving the following equation:

$$
b_{jk} = \arg \min_{b_k} \left\{ \sum_{n=1}^{N} \left[ y_{ni} - \sum_{k=1}^{q} b_k \hat{y}_k^{(-n)} \right]^2 \right\}, \quad j, k = 1, \ldots, q,
$$

(5.3.75)

where $\hat{y}_k^{(-n)}$ is the OLS prediction of the $k$th response for the $n$th observation, obtained with it removed from the training sample. Equivalently,

$$
\hat{Y}^{(-n)} = (1 - g_n) Y_n + g_n \hat{Y}_n,
$$

(5.3.76)

where $Y_n$ is the response matrix with all of the observations, $\hat{Y}_n$ is the OLS estimate on the full sample and

$$
g_n = \frac{1}{1 - h_{nn}}.
$$

(5.3.77)

As before, $h_{nn}, n = 1, \ldots, N$ is the diagonal elements of the $N \times N$ ‘hat’ matrix

$$
H = X(X^T X)^{-1} X^T,
$$

(5.3.78)

and $X$ is the predictor data matrix. Substituting equation (5.3.76) into equation (5.3.75) we obtain the cross-validated estimate of the shrinking matrix $B$. 

---

116
In the GCV-based C&W procedure, each \( h_{nn} \) is approximated by its average over \( N \) observations
\[
\overline{h} = \frac{1}{N} \sum_{n=1}^{N} h_{nn} = \frac{1}{N} \text{trace}(H) = r
\]  
(5.3.79)
with \( r = p/N \). Using this approximation the solution for the elements of the shrinking matrix \( B \) in equation (5.3.75) becomes
\[
b_{ik} = \arg \min_{\beta_k} \left( \sum_{n=1}^{N} \left( y_{ni} - \sum_{k=1}^{q} \beta_k (1 - g) y_{nk} + g \hat{y}_{nk} \right) \right)^2, \quad j, k = 1, \ldots, q,
\]  
(5.3.80)
where
\[
g = 1/(1 - r).
\]  
(5.3.81)

The normal equations for the solution, in matrix notation, are obtained as follows
\[
L(Y, \hat{Y}, B) = (Y - B [(1 - g)Y + g \hat{Y}])^2
\]  
(5.3.82)

\[
\frac{\partial L}{\partial B} = -2 \left( Y - B [(1 - g)Y + g \hat{Y}] \right) [(1 - g)Y + g \hat{Y}] = 0
\]
\[
Y [(1 - g)Y + g \hat{Y}] - B [(1 - g)Y + g \hat{Y}]^T [(1 - g)Y + g \hat{Y}] = 0
\]

\[
B [(1 - g)Y^T + g \hat{Y}^T] [(1 - g)Y + g \hat{Y}] = (1 - g)Y^T Y + g Y^T \hat{Y}
\]  
(5.3.83)

From equation (5.3.83) we obtain
\[
B [(1 - g)^2 Y^T Y + g(1 - g)Y^T \hat{Y} + g(1 - g)\hat{Y}^T Y + g^2 \hat{Y}^T Y] = (1 - g)Y^T Y + g Y^T \hat{Y}
\]
\[
B [(1 - g)^2 Y^T Y + g Y^T \hat{Y} - g^2 Y^T \hat{Y} + g \hat{Y}^T Y - g^2 \hat{Y}^T Y + g^2 \hat{Y}^T Y] = (1 - g)Y^T Y + g Y^T \hat{Y}
\]
\[
(Y^T Y)^{-1} B [(1 - g)^2 Y^T Y + 2g Y^T \hat{Y} - g^2 Y^T \hat{Y}] = (1 - g)Y^T Y + g Y^T \hat{Y}(Y^T Y)^{-1}
\]
\[
B [(1 - g)^2 I_q + 2g(Y^T \hat{Y})(Y^T Y)^{-1} - g^2 (Y^T \hat{Y})(Y^T Y)^{-1}] = (1 - g)I_q + g(Y^T \hat{Y})(Y^T Y)^{-1}.
\]
Because $H = X(X^T X)^{-1} X^T$ and $\bar{Y} = HY$, we can write $\bar{Q} = (Y^T Y)^{-1} Y^T X(X^T X)^{-1} X^T Y = (Y^T Y)^{-1} Y^T H Y = (Y^T Y)^{-1} Y \bar{Y}$. Therefore,

\[
B[(1-g)^2 I_q + 2g \bar{Q}^T - g^2 \bar{Q}^T] = (1-g)I_q + g \bar{Q}^T
\]

\[
B[(1-g)^2 I_q + (2g - g^2) \bar{Q}^T] = (1-g)I_q + g \bar{Q}^T
\]

\[
B[(1-g)^2 I_q + (2g - g^2) \bar{T}^{-1} \bar{C}^2 \bar{T}] = (1-g)I_q \bar{T}^{-1} \bar{C}^2 \bar{T}
\]

This shows that the solution $B$ is a diagonal matrix in the same co-ordinate system that diagonalizes $\bar{Q}$. In particular,

\[
B = \bar{T}^{-1} \bar{D} \bar{T}, \quad \bar{D} = \text{diag}[\hat{d}_1, \ldots, \hat{d}_q]
\]

with the diagonal elements $\hat{d}_j$ derived as follows

\[
\hat{d}_j[(1-g)^2 + (2g - g^2) \bar{c}_j^2] = (1-g) + g \bar{c}_j^2
\]

\[
\hat{d}_j = \frac{(1-g) + g \bar{c}_j^2}{(1-g)^2 + (2g - g^2) \bar{c}_j^2}
\]

By substituting $g = 1/(1-r)$ we obtain

\[
\hat{d}_j = \frac{(1 - r) + \frac{\bar{c}_j^2}{1-r}}{(1-r)^2 + \frac{2\bar{c}_j^2}{1-r} - \frac{1}{1-r^2} \bar{c}_j^2}
\]

\[
= \frac{\bar{c}_j^2 - r}{(1-r)^2 + \frac{2\bar{c}_j^2}{1-r} - \frac{1}{1-r^2}}
\]

\[
= \frac{\bar{c}_j^2 - r}{(1-r)^2 + \frac{2\bar{c}_j^2}{1-r} - \frac{1}{1-r^2}}
\]

\[
= \frac{\bar{c}_j^2 - r}{(1-r)^2 + \frac{2\bar{c}_j^2}{1-r} - \frac{1}{1-r^2}}
\]

\[
= \frac{(1-r)(\bar{c}_j^2 - r)}{(1-r)^2 2\bar{c}_j^2 + r^2 (1-\bar{c}_j^2)}, \quad j = 1, \ldots, q.
\]

The GCV canonical shrinkage factors $\hat{d}_j$ in equation (5.3.85) are universally smaller (more shrinked) than the corresponding population-based values $d_j$ in equation (5.3.64) (assuming $c_j = \bar{c}_j$) for all values of $\bar{c}_j^2$ and $r$. This compensates for the upward bias in the
estimates \( \hat{c}_j \) of the population values \( c_j \). Such an effect becomes more pronounced as \( r \) increases because the GCV estimate of the upward bias becomes larger with increasing \( r \).

5.3.5 Curds and whey ridge regression

C&W provides a general framework for combining multiple-response estimates of any type (Breiman & Friedman 1997). C&W can be suitably generalized and made applicable to settings in which either \( X^T X \) or \( Y^T Y \) is singular or nearly singular. Indeed, C&W is not strictly defined when either \( Y^T Y \) or \( X^T X \) is singular. This is a consequence of the fact that C&W estimates the canonical correlation matrix by means of an eigenanalysis of \( \hat{Q} = (Y^T Y)^{-1} Y^T X (X^T X)^{-1} X^T Y \). As was stated by Breiman & Friedman (1997), singular \( Y^T Y \) causes no special problem. Indeed, the response linear combinations (eigenvectors of \( Y^T Y \)) corresponding to zero variance (eigenvalues) are simply defined to have zero (canonical) correlation with the predictors, and the usual canonical correlation analysis is then confined to the non-zero variance subspace of the responses by using the generalized inverse of \( Y^T Y \). As described in section 5.2, GIR, PCR and RR can be used to deal with the singularity of \( X^T X \), and thus of \( \hat{Q} \) (Izenman 2008). Using RR, Breiman & Friedman (1997) developed C&W-RR, an approach that combines C&W-GCV with RR. In C&W-RR the coefficient matrix \( \hat{\beta}_\lambda \in \mathbb{R}^{q \times p} \) is obtained from separate RRs of each response on the predictors

\[
\hat{\beta}_\lambda = (X^T X + \lambda I_p)^{-1} X^T Y
\]  

(5.3.86)

using a common value of the ridge parameter \( \lambda \) for all of the responses. This leads to the corresponding RR response estimates

\[
\hat{Y}(\lambda) = \hat{\beta}_\lambda X.
\]  

(5.3.87)

The value \( \hat{\lambda} \) of the (common) ridge parameter \( \lambda \) is chosen by (fivefold) cross-validation

\[
\hat{\lambda} = \arg \min_\lambda {1 \over N} \sum_{j=1}^q \sum_{n=1}^N (\hat{y}_{ni} - \hat{y}_j^{(-n)}(\lambda))^2.
\]  

(5.3.88)
The principle of C&W is to shrink the OLS estimates $\hat{Y}$

$$\bar{Y} = B\hat{Y}$$  \hspace{1cm} (5.3.89)

using an optimal shrinking matrix $B$ that take the form $B = T^{-1}DT$. $T$ is the matrix of the canonical coordinates and $D$ is a diagonal matrix. Analogously, the C&W-RR estimates are then given by

$$\bar{Y} = (\bar{T}^{-1}D\bar{T})\hat{\beta}_\lambda X,$$  \hspace{1cm} (5.3.90)

$$D = \text{diag}(d_1, \ldots, d_q),$$  \hspace{1cm} (5.3.91)

where $\bar{T} \in \mathbb{R}^{q \times q}$ is obtained by a canonical correlation analysis between the sample responses $Y$ and their corresponding ridge estimates $\hat{Y}_\lambda \in \mathbb{R}^{N \times q}$.

The diagonal matrix $D$ in equation (5.3.91) is given by the C&W-GCV formula

$$\hat{d}_j = \frac{(1 - r)(\bar{c}_j^2 - r)}{(1 - r)\bar{c}_j^2 + r^2(1 - \bar{c}_j^2)}, \hspace{1cm} j = 1, \ldots, q.$$  \hspace{1cm} (5.3.94)

and $\hat{d}_j \leftarrow \max(\hat{d}_j, 0)$. Here, $\bar{c}_j^2$ is given by equation (5.3.93) and $r = p/N$ is replaced with

$$\bar{r} = \frac{1}{N} \text{trace}[X(X^TX + \hat{\lambda}I_p)^{-1}X^T].$$  \hspace{1cm} (5.3.95)

Note that this C&W-RR procedure generalizes C&W-GCV in that it reduces to C&W-GCV when $\hat{\lambda} = 0$ (Breiman & Friedman 1997).

### 5.4 Towards problem-based model selection

The least squares estimates of a multivariate multiple linear regression are given by a separate least squares regression of each response on the predictor variables. This means that multiple outputs do not affect one anothers least squares estimates (Hastie et al. 2009).
The same result is obtained even if the errors are correlated and a weighted criterion is used. Strategies that exploit correlations in the different responses have been proposed (e.g., reduced rank regression, two-block partial least squares, curds and whey, curds and whey ridge regression). Worthier competitors to these multivariate strategies are biased unireponse methods (e.g., ridge regression, principal component regression and partial least squares) and regularized unireponse methods (e.g., the Lasso and elastic net). These competitors do not take into consideration the correlations in the response variables but can substantially improve the OLS estimates in terms of model prediction accuracy.

As stated by Breiman & Friedman (1997) ‘An important issue is whether any of the multivariate multiple regression procedures offer sufficient improvement over separate multiple regression methods...’. Of interest is therefore the investigation of separate (unireponse) and multivariate modeling techniques, as well as the selection of the most promising approach to be used in the one-stage evolutionary approach (see section 4). In such a context relevant are the No Free Lunch (NFL) theorems by Wolpert & Macready (1997). The NFL theorems state that ‘There is no universally best algorithm (or modeling technique) to tackle a broad problem domain’. Indeed, for any algorithm (or model), any elevated performance over one class of problems is offset by diminished performance over another class (Wolpert & Macready 1997). Furthermore, as pointed out by Breiman & Friedman (1997), the model performance ‘... may depend on the detailed nature of the problem at hand in terms of the number of observations, the number of response variables, their correlation structure, signal-to-noise ratio, collinearity of the predictor variables, etc.’ Crucial is therefore the understanding of the problem characteristics in order to select the most appropriate technique that will ideally take into consideration known structural properties of the problem at hand (Smith-Miles 2008). In the next chapter, a meta-learning architecture is proposed to generate a knowledge system capable of providing some guidelines to select the most promising modeling technique in multireponse settings.
Chapter 6

A meta-learning framework for the model selection

In this chapter a meta-learning framework for model selection is proposed. The aim is to develop a system capable of selecting the most promising model (or a subclass of promising models) to be combined with EMMA, given the characteristics of the problem at hand. The separate and multivariate multiple regression methods introduced in Chapter 5, and some models described in Chapter 3, are investigated with the final objective to build some general rules relating the features of the problems analysed and the performances of the models fitted. This can be achieved by means of a large computational study. The aim of the chapter is to lay the theoretical foundation for such a large computational study and to develop the components necessary for its implementation. The chapter is organized as follows. The meta-learning approach is first described by depicting its origin, stating its general purpose and introducing its basic components. Subsequently, the framework basic components are separately investigated. In particular, section 6.1 gives an account of the procedures developed to simulate the data (instances). Section 6.2 lists the investigated modeling techniques and the measure adopted to evaluate their performances. Section 6.3 describes the metrics generated to measure the features of the derived instances. Section 6.4 draws conclusions and gives an idea of the framework outcomes providing examples of the expected results. Ongoing research and future directions are depicted too.
Model selection is typically performed experimentally, that is evaluating the performance of a large number of models on the given problem, and selecting the best candidate according to a predefined criteria, such as AIC, BIC, cross-validation (Brazdil Pavel 1998, Hastie et al. 2009). An alternative approach to model selection employs meta-learning. Meta-learning means ‘learning about learning’ and, in this context, it refers to learning about learning algorithms (models) performance. Meta-learning studies how learning systems can increase in efficiency thanks to the knowledge gained from the analysis of several tasks (Vilalta & Drissi 2002). As reviewed by Vilalta & Drissi (2002), different facets of meta-learning exist. The one adopted here builds meta-rules matching domains with algorithm performance. In this context, algorithm and model selection are basically the same concepts (Rice 1976), therefore the terms algorithm and model are here used as synonymous. The meta-learning approach requires the availability of a large collection of problem instances (real-world or simulated data) of various complexities, a large number of diverse models for tackling problem instances, metrics to evaluate model performance, and suitable metrics to characterize the features of the instances (Smith-Miles 2008). As described by Smith-Miles (2008), a commonality of goal exist between meta-learning and the so-called algorithm selection problem. Indeed, the algorithm selection problem seeks to answer the question: ‘Which algorithm is likely to perform best for my problem?’ A formal abstract model has been presented by Rice (1976) to explore the algorithm selection problem. The model, schematically depicted in Figure 6.1, is characterized by four components, namely the problem space, the feature space, the algorithm space, and the performance space. The problem space $P$ represents the set of data (instances) of a problem class. The feature space $F$ contains measurable characteristics of the problems. The algorithm space $A$ is the set of all considered algorithms for tackling the problem. The performance space $Y$ represents the mapping of each algorithm to a set of performance metrics. The task is to learn the mapping between the measured features and the algorithm performances. The meta-data, defined as $<P,A,Y,F>$, are used to produce a prediction of which algorithm will perform best for unseen test instances (Smith-Miles 2008). Relevant in this context are the two European projects StatLog (Comparative Testing of Statistical and Logical Learning), aimed to provide an objective assessment of the strengths and weaknesses of the various approaches to classification, and METAL (a meta-learning...
Figure 6.1: Schematic diagram of the algorithm selection problem model by Rice (1976). For a given problem instance \( x \in P \), with features \( f(x) \in F \), the objective is to determine the selection mapping \( S(f(x)) \) into the algorithm space \( A \), such that the selected algorithm \( \alpha \in A \) maximizes the performance mapping \( y(\alpha(x)) \in Y \) (the best algorithm \( \alpha \) is returned) (Smith-Miles 2008).

Assistant for providing user support in machine learning and data mining, aimed to develop model selection for both classification and regression problems (Smith-Miles 2008). Meta-learning was originally employed for classification problems (Aha 1992) and subsequently extended to other domains including regression, but also time-series forecasting, sorting, constraint satisfaction, and optimization (Smith-Miles 2008). The objective is here to widen the meta-learning approach to the multiresponse regression domain. A customized representation of the model selection framework for the newly investigated domain is depicted in Figure 6.2. The procedures adopted to generate suitable meta-data \( <P, A, Y, F> \) are described further below.

### 6.1 The problem space

The problem space should present a large number of multiresponse instances characterized by various levels of complexities. Because there are simply not enough real datasets at present, simulations have been used. However, the use of simulated data may not sufficiently cover the space of a (previously unseen) real dataset and interest emerges in how to simulate datasets which are similar to a given real dataset in the sense that they encompass the space around (Kopf et al. 2000). This section describes the procedures developed to simulate instances with multiple predictors and multiple responses, in a
Figure 6.2: Schematic diagram of the meta-learning framework for multiresponse model selection, as inspired by the algorithm selection problem model by Rice (1976). The objective is to model the relationship between the instance features and the multiresponse models performances so that, given a multiresponse instance, the model with the highest estimated predictive accuracy is selected.

view to simulate real-world-like datasets. The response values are obtained according to

\[ Y = f(X) + \epsilon. \]

(6.1.1)

where \( Y \) is a \( N \times q \) response data matrix, \( X \) is a \( N \times p \) predictors data matrix, \( f = (f_1, \ldots, f_q) \) is a set of \( q \) functions, and \( \epsilon \) is a \( N \times q \) matrix of errors. Let \( x_n = (x_{n1}, \ldots, x_{np}) \) be the vector of predictor measurements for the \( n \)th case, and let \( x_1, \ldots, x_p \) be the column vectors of \( X \), spanning the subspace \( \mathbb{R}^N \). Similarly, let \( y_n = (y_{n1}, \ldots, y_{nq}) \) be the vector of predictor measurements for the \( n \)th case, and let \( y_1, \ldots, y_q \) be the column vectors of \( Y \), spanning the subspace \( \mathbb{R}^N \). Finally, let \( \epsilon = (\epsilon_1, \ldots, \epsilon_q) \) be the \( N \times q \) matrix of errors. As described in section 5.1, the responses can either share the same functional relationship with the predictors (that is, \( f_1 = f_2 = \ldots = f_q \)), or be different functions of \( X \). Two approaches have been developed to simulate the problems. The aim of the first approach is to generate data with characteristics similar to those of the chemical problem described in Chapter 2, and to simulate responses sharing the same structural relationship with the predictors. Instead, the goal of the second approach is to simulate datasets with more general features and with a structural relationship between \( X \) and \( Y \) that differs for each response. This is
achieved by generating $q$ different functions of the predictors or their transformations. In both the approaches, the error term $\epsilon$ is simulated using either the multivariate Normal probability distribution or a copula-based methodology. The first approach is described in section 6.1.1, whereas the second approach is depicted in section 6.1.2. The procedure to generate the error term is shown in section 6.1.3 and comes with some background theory about copulas.

### 6.1.1 First approach

The first aim of the approach is to simulate responses that share the same functional relationship with the predictors. This structure is obtained using simple polynomial models involving the same predictors but different coefficient value. The second aim of the approach is to generate instances with features similar to those of the real-world problem described in chapter 2. To achieve this, the data collected during the preliminary study (investigation conducted to define the predictor ranges and levels – section 2.2) and during the experimentation (experiments designed by the one-stage evolutionary approach – Chapter 4) are analysed. Prior to analysis, experimental points resulting in unstable compositions are removed from the dataset. Then, the selected data are investigated to reproduce the predictors probability densities and to simulate the coefficients of simple polynomial models.

**Simulation of the predictors**

The inputs of the motivating problem are continuous variables whose values have been limited to a finite set of levels. Furthermore, constraints exist on a subset of the predictors. They are the linear constraint in equation (2.2.2) and the inequality constraint in equation (2.2.3). Figure 6.3 depicts the probability densities of the chemical components (inputs) used to produce the coatings. The values in the $x$-axis are the predictor levels, which have been standardized to be in the range $[0,1]$. As observed in Figure 6.3, each predictor is characterized by one level (spike) whose probability is higher with respect to the others. A decreasing trend is observed for the probabilities of the remaining levels. It can be further noted that, if the spike is located in a marginal position, its probability is in the range $[0.4,0.6]$ and the remaining probabilities tend to decrease exponentially. If the spike is in a
central position, its probability is in the range [0.25,0.40] and the remaining probabilities tend to decrease in correspondence to more marginal factor levels. In simulating the predictors, two aspects need to be considered, namely the constraints on a subset of the input variables and the number of levels per variable. For simplicity only the linear constraint in equation (2.2.2) is here considered. The simplest approach to generate a set of linearly constrained predictors, would be to calculate the values of one input as a function of the others. However, the number of levels obtained can be different from the number of levels required. To face this difficulty, the constrained predictors are regarded as mixture variables, that is variables that assume values in [0,1] and such that $\sum_i X_i = 1$, with
Table 6.1: Mixture experiments for three predictors with three levels.

\[
\begin{array}{ccc}
  x_1 & x_2 & x_3 \\
  1 & 0 & 0 \\
  0.5 & 0.5 & 0 \\
  0 & 1 & 0 \\
  0.5 & 0 & 0.5 \\
  0 & 0.5 & 0.5 \\
  0 & 0 & 1 \\
\end{array}
\]

\(i = 1, \ldots, p\) (Montgomery 2009). The following procedure has been defined to simulate the predictors. Let \(X_i, i = 1, \ldots, p\), be input variables with \(0 \leq X_i \leq 1\), and let \(n_u\) be the number of unconstrained predictors. Furthermore, let \(n_l\) be the number of levels per predictor and let \(l_{iz}\) indicate the \(z\)th level of the \(i\)th input. To obtain the predictor levels, the interval \([0,1]\) is divided in \(n_l - 1\) subintervals of equal width. The unconstrained predictors and the first constrained input variable, namely the set \(X_{1st} = \{X_1, \ldots, X_{n_u+1}\}\), are then generated and subsequently the remaining inputs, namely the set \(X_{2nd} = \{X_{n_u+2}, \ldots, X_p\}\), are obtained. To derive the unconstrained predictors and the first constrained input variable, the following procedure is applied. For each predictor in \(X_{1st}\), a level \(l_{iz}\) is randomly selected (spike) and according to its position (value of \(z\)) the level probabilities, say \(P(l_{iz})\), are obtained as a function of the level values \((l_i1, \ldots, l_{in})\). In particular, if the spike is in a non-central position the employed function is \((e^{l_{iz}})^3\), otherwise, the used function is \(e^{l_{iz}}\). Subsequently, the obtained numbers are normalized to have minimum value \(P_{\text{min}}\) and maximum value \(P_{\text{max}}\). If the spike is in a non-central position \(P_{\text{min}} \sim \mathcal{U}(0, 0.1)\) and \(P_{\text{max}} \sim \mathcal{U}(0.4, 0.6)\), whereas if the spike is in a central position \(P_{\text{min}} \sim \mathcal{U}(0.1, 0.2)\) and \(P_{\text{max}} \sim \mathcal{U}(0.25, 0.4)\). To introduce some irregularities in the predictors probability densities, random draws from a \(\mathcal{N}(0, 0.01)\) are added to the derived probabilities \(P(l_{iz})\). To derive the probabilities of the remaining predictors (variables in \(X_{2nd}\)), a set of mixture experiments, satisfying the linear constraint \(\sum_{i=n_u+1}^{p} X_i = 1\), is first generated using the R package AlgDesign (Wheeler 2010). To understand the adopted mechanism, consider the example in Table 6.1. Let \(x_{ni}\) be the \(n\)th observation of the \(i\)th predictor. Assume that the simulated value of the first constrained predictor, \(x_{11}\), is equal to 0.5 for three observations, say 3, 10, and 42. Then, \(x_{n1} = 0.5\) and \(n = \{3, 10, 42\}\). According to Table 6.1, two mixture experiments are identified, namely \((0.5, 0.5, 0)\) and \((0.5, 0, 0.5)\). Values \((0.5, 0)\) or \((0, 0.5)\) can therefore be assigned to \((x_{n2}, x_{n3})\). In summary, according to the simulated value of the first constrained
predictor, $x_{n+1}$, a set of alternatives is identified using the previously determined mixture experiments. The available alternatives are then randomly sampled, and the values of the remaining constrained predictors is determined. Finally, the generated values are divided by their sum, so that the sum of the generated numbers is one and probabilities are obtained. Once the probabilities of the levels have been simulated, the predictors frequencies are derived considering the predefined size $N$ of the dataset. Figure 6.4 depicts the probability densities of six predictors with six levels simulated using the described procedure.
Simulation of the functional relationship

In the first approach, the functional relationship \( f(X) \) is obtained using simple polynomial models, namely the first-order model

\[
Y_j = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \epsilon_j \quad j = 1, \ldots, q
\]  

(6.1.2)

and the second-order model

\[
Y_j = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \sum_{i=1}^{p} \beta_{ii} X_i^2 + \sum_{i<k} \sum_{i<k} \beta_{ik} X_i X_k + \epsilon_j \quad j = 1, \ldots, q.
\]  

(6.1.3)

The responses of a specific instance are obtained using the same model (either the first- or the second-order model) but different coefficient values. As a result, the responses share the same functional relationship with the predictors and are expected to be correlated. The simulated coefficient values are obtained on the basis of a preliminary study conducted on the chemical data collected. In particular, predictors are normalized in the range \([0,1]\) and responses are standardized to have unit standard deviation. The models in equations (6.1.2) and (6.1.3) are fitted to the transformed data and the estimated coefficients are investigated. The estimated coefficients of the first-order model have mean equal to 0.46 and standard deviation equal to 1.72. Therefore, the simulated coefficients are generated according to \( N(\mu = 0.5, \sigma = 1.7) \). Figure 6.5 shows the estimated coefficients, whereas Figure 6.6 depicts an example of the simulated coefficients for a problem with six predictors and four responses. The estimated coefficients of the second-order model are depicted in Figures 6.7-6.9. They can be divided into three classes, namely the first-order coefficients \( \beta_i, i = 1, \ldots, p \), the second-order coefficients \( \beta_{ii} \), and the interaction coefficients \( \beta_{ik} \), with \( i,k = 1, \ldots, p \) and \( i < k \). As highlighted by the dashed lines, each class can be further divided into three subclasses (low, central, and high) in order to be able to reproduce the observed patterns. The coefficient values used to identify the subclasses are \([-10,10]\) for the first-order term coefficients (Figure 6.7), \([-5,5]\) for the second-order term coefficients (Figure 6.8), and \([-10,10]\) for the interaction term coefficients (Figure 6.9). For each subclass, mean and standard deviation values are obtained and summarized in Tables 6.2-6.4. These values are then used to define the parameters of a \( N(\mu, \sigma) \) from
Figure 6.5: Scatterplots of the coefficients of the first-order model fitted to the standardized chemical data.

Table 6.2: Range, mean and standard deviation of the subclasses for the first-order term coefficients estimated by fitting a second-order linear model to the chemical data. The three coefficient subclasses are identified by means of the values −10 and 10.
Figure 6.6: Scatterplots of the simulated coefficients for a first-order model.

Table 6.3: Range, mean and standard deviation of the subclasses for the second-order term coefficients estimated by fitting a second-order linear model to the chemical data. The three subclasses are identified by means of the values $-5$ and $5$.

<table>
<thead>
<tr>
<th>Subclass</th>
<th>Range</th>
<th>Mean</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>$[-16.32, -5)$</td>
<td>-11.91</td>
<td>3.29</td>
</tr>
<tr>
<td>central</td>
<td>$[-5, 5)$</td>
<td>-0.68</td>
<td>1.82</td>
</tr>
</tbody>
</table>

Table 6.4: Range, mean and standard deviation of the subclasses for the interaction term coefficients estimated by fitting a second-order linear model to the chemical data. The three subclasses are identified by means of the values $-10$ and $10$.

<table>
<thead>
<tr>
<th>Subclass</th>
<th>Range</th>
<th>Mean</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>$[-29.43, -10)$</td>
<td>-19.13</td>
<td>8.24</td>
</tr>
<tr>
<td>central</td>
<td>$[-10, 10)$</td>
<td>0.11</td>
<td>3.94</td>
</tr>
<tr>
<td>high</td>
<td>$[10, 70.76]$</td>
<td>56.18</td>
<td>9.98</td>
</tr>
</tbody>
</table>
Figure 6.7: Scatterplots of the first-order term coefficients for the second-order model fitted to the observed data. Two dashed lines are drawn in correspondence to the values −10 and 10, so that the low, central, and high subclasses are identified.

which to simulate the regression coefficients of each subclass. In particular, to simulate the first-order term coefficients, the 50% of the values are drawn from $\mathcal{N}(-1, 5)$, while the remaining coefficients are drawn partly from $\mathcal{N}(20, 8)$ and partly from $\mathcal{N}(-52, 9.5)$. To simulate the second-order term coefficients, the 50% of the values are drawn from $\mathcal{N}(-0.7, 1.82)$, and the remaining coefficients are drawn either from $\mathcal{N}(22, 8.4)$ or from $\mathcal{N}(-12, 3.3)$. Similarly, the interaction term coefficients are simulated by drawing the 80% of the values from $\mathcal{N}(0.11, 3.9)$, whereas the remaining coefficients are drawn either from $\mathcal{N}(56, 10)$ or from $\mathcal{N}(-19, 2)$. The low subclass of the interaction term coefficients is simulated using a standard deviation smaller with respect to the observed one. Indeed,
if the value of 8.24 reported in Table 6.4 is used, some of the simulated coefficients for this subclass wrongly become part of the central subclass. This is due to the fact that the interaction term coefficients of the low subclass are significantly different for the responses $I_{\text{max}} (\mu = -11.87, \sigma = 1.62)$ and $I_{\text{back}} (\mu = -26.4, \sigma = 2.94)$. Examples of simulated coefficients for the second-order model are shown in Figures 6.10-6.12.
Figure 6.9: Scatterplots of the interaction term coefficients for the second-order model fitted to the observed chemical data. Two dashed lines are drawn in correspondence to the values $-10$ and $10$, so that the low, central, and high subclasses are identified.

6.1.2 Second approach

The aim of the second approach is to generate instances with general features (not necessarily similar to those of the motivating problem) and whose responses do not share the same functional structure. The predictors $X$ and the functional relationships $f(X)$ are obtained as described further below.

Similarly to what described in Section 4.1, the continuous predictors are restricted to a finite set of values (levels). Each predictor is assumed to be in the range $[0,1]$ and to have $n_l$ levels. The $n_l$ levels of each predictor are generated by dividing the interval $[0,1]$ in
$n_l - 1$ subintervals of the same width and the experimental region is obtained by generating all of the possible combinations of the input levels. Therefore, with $p$ predictors and $n_l$ levels per predictor, the experimental region is composed by $n_l^p$ points. The experimental region is subsequently reduced by means of a linear constraint on the last $p - n_u$ variables, where $n_u$ is the number of unconstrained predictors. The employed constraint is:

$$\sum_{i=n_u+1}^{n_p} X_i < Q_3 \quad (6.1.4)$$

where $Q_3$ corresponds to the third quartile of $\sum_{i=n_u+1}^{n_p} X_i$. 

---

**Figure 6.10:** Scatterplots of simulated first-order term coefficients for a second-order model.
Figure 6.11: Scatterplots of simulated second-order term coefficients for a second-order model.

To simulate the functional relationships $f(X)$ a customized framework is used. A methodology that intrinsically presents different levels of complexity is developed using the software R (R Development Core Team 2009). The proposed tool takes advantage of a set of functions introduced by Huband et al. (2005) to generate test problems for multi-objective evolutionary algorithms (MOEAs). Each response is generated as a function of $n_p$ randomly selected predictors, with $n_p \in \{1, \ldots, p\}$. Each selected predictor is transformed using a transformation function randomly selected from a set of five transformation functions, namely polynomial, flat region, linear shift, deceptive, and multi-modal. Then, the transformed inputs $T_1, \ldots, T_{n_p}$ are combined with each others using a combination function randomly drawn from a set of three combination functions, namely weighted
Figure 6.12: Scatterplots of simulated interaction term coefficients for a second-order model.

The employed functions are depicted in Figures 6.13-6.14 and are described further below. The polynomial transformation

$$b_{\text{poly}}(X, \alpha) = X^\alpha$$ (6.1.5)

has parameter $\alpha > 0$ and $\alpha \neq 1$; for values of $\alpha > 1$ the input $X$ is biased towards zero, whereas for values of $\alpha < 1$ it is biased towards one. The flat region transformation

$$b_{\text{flat}}(X, A, B, C) = A + \min(0, \lfloor X - B \rfloor) \frac{A(B - X)}{B} - \min(0, \lfloor C - X \rfloor) \frac{(1 - A)(X - C)}{1 - C}$$ (6.1.6)
has parameters $A, B, C \in [0,1]$. The parameters are as follows: $B < C$, if $B = 0$ then $A = 0$ or $C \neq 1$, if $C = 1$ then $A = 1$ or $B \neq 0$. Equation (6.1.6) maps all the values of $X$ in the range $[B, C]$ to the value of $A$, thus generating a flat region. The linear shift transformation
\[ s_{\text{linear}}(X, A) = \frac{|X - A|}{|A - X| + A} \] (6.1.7)
has parameter $A \in (0, 1)$; with this transformation the value $X = A$ is mapped to zero. The deceptive transformation
\[ s_{\text{decept}}(X, A, B, C) = 1 + ((|X - A| - B) \times \
\frac{|X - A + B|(1 - C + \frac{A+B}{B})}{A - B} + \frac{|A + B - X|(1 - C + \frac{1-A+B}{B})}{1 - A - B} + \frac{1}{B}) \] (6.1.8)
has parameters $A$, $B$, and $C$, with $A \in (0,1)$, $0 < B << 1$, $0 < C << 1$, $A - B > 0$, and $A + B < 1$. This function generates transformations which are difficult to optimize because characterized by a global minimum and two local minima (deceptive minima). The global minimum is hard to find, and typically one of the two deceptive minima is wrongly selected to be the point that minimize the function. $A$ is the value at which $X$ is mapped to zero, $B$ is the width of the ‘aperture’, and $C$ is the value of the deceptive minima. The multi-modal transformation
\[ s_{\text{multi}}(X, A, B, C) = 1 + \cos \left[ (4A + 2)\pi \left( 0.5 - \frac{|X-C|}{2L(C-X)+C} \right) \right] + 4B \left( \frac{|X-C|}{2L(C-X)+C} \right)^{2} \] (6.1.9)
has parameters $A$, $B$, and $C$, with $A \in \{1, 2, \ldots\}$, $B \geq 0$, $(4A + 2)\pi > 4B$, and $C \in (0,1)$. This function generates multiple minima, whose number is controlled by $A$, and whose size is controlled by $B$. The values of $X$ are mapped to zero in correspondence to $C$. When $B = 0$, $2A + 1$ values of $X$ (one at $C$) are mapped to zero. When $B \neq 0$, there are $2A$ local minima, and one global minimum at $C$. Larger values of $A$ and smaller values of $B$ create more difficult problems. The transformed predictors, $T = (T_1, \ldots, T_{n_p})$, are combined using the following combination functions. The weighted sum is given by
\[ r_{\text{wsum}}(T, w) = \left( \frac{\sum_{k=1}^{n_p} w_k T_k}{\sum_{k=1}^{n_p} w_k} \right) \] (6.1.10)
where \( w = (w_1, \ldots, w_{n_p}) \) is a vector of weights and \( n_p \) is the number of transformed predictors. The weights \( w_k \in [0,1] \) are such that \( \sum_{k=1}^{n_p} w_k = 1 \). The weighted product is given by
\[
 r_{\text{wpod}}(T, w) = \left( \prod_{k=1}^{n_p} w_k T_k \right) / \left( \prod_{k=1}^{n_p} w_k \right) \tag{6.1.11}
\]
with \( w \) defined as for the weighted sum. Finally, the nonseparability combination function is given by
\[
 r_{\text{nonsep}}(T, A) = \sum_{k=1}^{n_p} \frac{T_k + \sum_{j=0}^{A-2} |T_k + T_{1+(k+j) \mod n_p}|}{|A/2(1 + 2A - 2\lfloor A/2 \rfloor)|} \tag{6.1.12}
\]
with parameter \( A \in \{1, \ldots, n_p\} \) controlling the degree of separability, and \( n_p \cdot \mod A = 0 \). This function considerably increases the difficulty of the simulated response in that good predictions of the response can be obtained only if all the transformed inputs are included in the model. Examples of the described functions are shown in Figures 6.13 and 6.14. The problem complexity depends on the values assigned to the parameters of the employed functions, and can be further improved if more than one transformation is applied to the same predictor. Consider for example Figure 6.15. The final response (right panel) is obtained as a function of two transformed input variables. The first transformed input (green line – left panel) is obtained applying a linear shift (black line – left panel) and a polynomial transformation function (red line). Similarly, the second transformed input (green line – central panel) is obtained using the multimodal function (black line – central panel) and the flat region transformations (red line – central panel).

The simulated response values and the values of the predictors that define the whole experimental region, are combined in a unique set of data, from which a subset of \( N \) data points are selected. Such a subset simulate the experiments investigated in the chemistry problem, that is the points in the experimental region which have been selected to be part of the initial design and to subsequently augment it (see Chapter 4). The \( N \) data points are sampled both randomly and nonrandomly. The purpose of nonrandom sampling is to promote the selection of good experimental points and simulate the uneven distribution of the points selected by EMMA. Indeed, EMMA samples promising regions of the search space with higher frequency and, as a consequence, a higher concentration of particles is detected in certain areas. Nonrandom sampling is achieved by assigning higher probability to well performing experiments, that is experiments that are close
Figure 6.13: Example of transformation functions employed in generating the responses by means of the second approach. From left to right and from top to bottom, the functions are as follows. Polynomial transformation with parameter $\alpha = 9$; flat region transformation with parameters $A = 0.64$, $B = 0.52$, and $C = 0.93$; linear shift transformation with parameter $A = 0.33$; deceptive shift transformation with parameters $A = 0.86$, $B = 0.04$, and $C = 0.002$. 
Figure 6.14: Example of transformation and combination functions employed generating the responses by means of the second approach. From left to right and from top to bottom, the functions are as follows. Multimodal transformation with parameters $A = 1$, $B = 4.23$, and $C = 0.18$; weighted sum combination with parameter $w = \{0.2, 0.8\}$; weighted product combination with parameter $w = \{0.2, 0.8\}$; nonseparability combination with parameter $A = 2$. 
to the optimal experimental point. Closeness to the optimal point is calculated using equation (4.1.2).

### 6.1.3 Error structures

The aim is to simulate instances with an error structure incorporating that of the real-world chemical problem described in Chapter 2. To investigate the error structure of the experimental data collected using EMMA, a MARS model (see chapter 3) is fitted to the observed data and its residuals are studied. The multivariate normality of the MARS residuals is tested using the Shapiro Wilk test (Royston 1982) available in the R package mvnormtest (Jarek 2009). Because the p-value of the test is equal to $2.035^{-05}$, the null hypothesis is rejected and evidence against the multivariate normality of the residuals is observed.

Three different error structures are simulated. Standard error structures are obtained using random draws from a multivariate Normal distribution with zero mean and covariance matrix given by either

$$
\Sigma = \left( \frac{1}{k} \sigma_m \right)^2 I_q
$$

(6.1.13)

or

$$
\Sigma = \left( \frac{1}{k} \sigma \right)^2 I_q.
$$

(6.1.14)

Here $k$ is the signal-to-noise ratio, $\sigma = (\sigma_1, \ldots, \sigma_q)$ is a $q$-vector of standard deviations and $\sigma_m = (\sum_{j=1}^{q} \sigma_j)/q$. A signal-to-noise ratio equal to 3, means that the simulated noise is three times smaller than the signal standard deviation, $\sigma_j(f_j(X))$, $j = 1, \ldots, q$. To simulate an
error structure similar to that of the observed data, a copula-based approach is adopted. In the following section, copulas are briefly introduced and the copula-based approach is described. For a more comprehensive description about copulas see Nelsen (2006) and Yan (2006).

**Copulas**

Copulas can be generally described as functions that join (or couple) multivariate distribution functions to their one-dimensional marginal distribution functions. Alternatively, they are defined as distribution functions whose one-dimensional margins are uniform (Nelsen 2006). Copulas are used as multivariate modeling tools when the multivariate dependence is of interest and the usual multivariate normality is in question (Yan 2006). One of their primary application is in simulation and Monte Carlo studies. In particular, samples are generated from a specified joint distribution, and then used to study mathematical models of real-world systems, or for statistical studies. Statistical studies include the comparison of a new statistical method with competitors, robustness properties, or the agreement of asymptotic with small sample results (Nelsen 2006). Central to the theory of copulas is the Sklar’s theorem, which elucidates the role that copulas play in the relationship between multivariate distribution functions and their univariate margins (Nelsen 2006). Sklar’s theorem is as follows.

**Sklar’s theorem.** Let $X$ and $Y$ be random variables with distribution functions $F$ and $G$, respectively, and joint distribution function $H$. Then there exists a copula $C$ such that

$$H(X, Y) = C(F(X), G(Y)) \quad (6.1.15)$$

holds. If $F$ and $G$ are continuous, $C$ is unique. Otherwise, $C$ is uniquely determined on $\text{Ran} F \times \text{Ran} G$, where $\text{Ran}$ is the range of the distribution function.

Most of the usefulness of copulas derives from the fact that, for strictly monotone transformations of the random variables, copulas are either invariant or change in predictable ways. As reported in Table 6.5, various families and subfamilies of copulas exist. Attention is here restricted to the elliptical family, and in particular to Normal copulas. The reason for this is that, differently from other families, elliptical copulas allow both positive and
negative association even when the copula dimension is higher than two (Yan 2006). Furthermore, Normal copulas are of easy implementation and their dependence structure is fully determined by the correlation matrix. Let $F$ be the multivariate cumulative distribution function (CDF) of an elliptical distribution, and let $F_i$, $i = 1, \ldots, p$, be the CDF of the $i$th margin. Then, the elliptical copula determined by $F$ is

$$C(u_1, \ldots, u_p) = F[F_1^{-1}(u_1), \ldots, F_p^{-1}(u_p)] \quad i = 1, \ldots, p$$

(6.1.16)

where $F_i^{-1}$ is the inverse (or quantile function) of $F_i$. The density of an elliptical copula is obtained by differentiating equation (6.1.16), and is given by

$$c(u_1, \ldots, u_p) = \frac{f[F_1^{-1}(u_1), \ldots, F_p^{-1}(u_p)]}{\prod_{i=1}^{p} f_i[F_i^{-1}(u_i)]},$$

(6.1.17)

where $f$ is the joint probability density function (PDF) of the elliptical distribution, and $f_i$ are marginal density functions. The density of a Normal copula with dispersion matrix $\Sigma$ is

$$c(u_1, \ldots, u_p|\Sigma) = |\Sigma|^{-1/2} \exp\left\{ \frac{1}{2} c^T (I_p - \Sigma) c \right\}$$

(6.1.18)

where $c = (q_1, \ldots, q_p)^T$ with $q_i = \Phi^{-1}(u_i)$, and $\Phi$ is the CDF of $\mathcal{N}(0, 1)$ (Yan 2006). Unstructured dependence of the form

$$\begin{pmatrix}
1 & \rho_1 & \rho_2 \\
\rho_1 & 1 & \rho_3 \\
\rho_2 & \rho_3 & 1
\end{pmatrix}$$

(6.1.19)

is here considered. To fit a copula-based model, both copula parameters and marginal parameters need to be estimated. Let $(x_{n1}, \ldots, x_{np})^T$, $n = 1, \ldots, N$, be $N$ independent realizations from a multivariate distribution. Suppose that the multivariate distribution is specified by $p$ margins with CDF $F_i$ and PDF $f_i$, and a copula with density $c$. Let $\beta$ be

---

**Table 6.5: Families and subfamilies of copulas implemented in R.**

<table>
<thead>
<tr>
<th>Families</th>
<th>Archimedean</th>
<th>Elliptical</th>
<th>Extreme value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subfamilies</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clayton</td>
<td>Normal</td>
<td>Galambos</td>
<td></td>
</tr>
<tr>
<td>Gumbel</td>
<td>t</td>
<td>Husler Reiss</td>
<td></td>
</tr>
<tr>
<td>Frank</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ali-Mikhail-Haq</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The vector of marginal parameters and $\alpha$ be the vector of copula parameters (association parameters). The loglikelihood function is

$$l(\beta, \alpha) = \sum_{n=1}^{N} \log c(F_1(x_{n1}; \beta), \ldots, F_p(x_{np}; \beta); \alpha) + \sum_{n=1}^{N} \sum_{i=1}^{P} \log f_i(x_{ij}; \beta).$$ (6.1.20)

Marginal parameters $\beta$ and copula parameters $\alpha$ are estimated using a two-stage method known as inference functions for margins (IFM). With this approach, marginal parameters $\hat{\beta}_{IFM}$ are first estimated by

$$\hat{\beta}_{IFM} = \arg \max_{\beta} \sum_{n=1}^{N} \sum_{i=1}^{P} \log f_i(x_{ni}; \beta).$$ (6.1.21)

Subsequently, given $\hat{\beta}_{IFM}$, the association parameters $\alpha$ are estimated by

$$\hat{\alpha}_{IFM} = \arg \max_{\alpha} \sum_{n=1}^{N} \log c(F_1(x_{n1}; \hat{\beta}_{IFM}), \ldots, F_p(x_{np}; \hat{\beta}_{IFM}); \alpha).$$ (6.1.22)

The best-fitting marginals, the estimated parameters, and the Kolmogorov-Smirnov test results for the applicative data are summarized in Table 6.6. Tables 6.7 and 6.8 report the measured correlation matrix and the estimated association parameters, respectively. Figures 6.16 and 6.17 depict the MARS residuals and an example of data generated from the fitted copula-based model. Error terms with a structure different from the standard normal multivariate distribution are generated using random draws from a Normal copula with unstructured correlation matrix (6.1.19) and the marginal distributions reported in Table 6.9. The R package copula is used to both fit the model and simulate copula-based errors (Yan 2006).
Table 6.7: Correlation matrix of the MARS model residuals.

<table>
<thead>
<tr>
<th></th>
<th>$\epsilon_{V_1}$</th>
<th>$\epsilon_{I_{\text{max}}}$</th>
<th>$\epsilon_{I_{\text{back}}}$</th>
<th>$\epsilon_{V_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon_{V_1}$</td>
<td>1</td>
<td>-0.07</td>
<td>-0.03</td>
<td>0.18</td>
</tr>
<tr>
<td>$\epsilon_{I_{\text{max}}}$</td>
<td>-0.07</td>
<td>1.00</td>
<td>0.09</td>
<td>-0.10</td>
</tr>
<tr>
<td>$\epsilon_{I_{\text{back}}}$</td>
<td>-0.03</td>
<td>0.09</td>
<td>1.00</td>
<td>-0.30</td>
</tr>
<tr>
<td>$\epsilon_{V_2}$</td>
<td>0.18</td>
<td>-0.10</td>
<td>-0.30</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 6.8: Normal copula parameter values estimated by maximum likelihood.

<table>
<thead>
<tr>
<th></th>
<th>$\epsilon_{V_1}$</th>
<th>$\epsilon_{I_{\text{max}}}$</th>
<th>$\epsilon_{I_{\text{back}}}$</th>
<th>$\epsilon_{V_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon_{V_1}$</td>
<td>1</td>
<td>-0.05</td>
<td>-0.07</td>
<td>0.18</td>
</tr>
<tr>
<td>$\epsilon_{I_{\text{max}}}$</td>
<td>-0.05</td>
<td>1.00</td>
<td>0.10</td>
<td>-0.14</td>
</tr>
<tr>
<td>$\epsilon_{I_{\text{back}}}$</td>
<td>-0.07</td>
<td>0.10</td>
<td>1.00</td>
<td>-0.32</td>
</tr>
<tr>
<td>$\epsilon_{V_2}$</td>
<td>0.18</td>
<td>-0.14</td>
<td>-0.32</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Figure 6.16: Graphical representation of the residuals obtained by fitting a multiresponse MARS model to the measured data.
**Figure 6.17:** Graphical representation of the errors simulated using the Normal copula fitted to the MARS residuals.

**Table 6.9:** One-dimensional distributions used to generate the copula-based error structure. 

<table>
<thead>
<tr>
<th>Marginal distribution</th>
<th>Parameter values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{N}(\mu, \sigma) )</td>
<td>( \mu = 0, \sigma = 0.9 )</td>
</tr>
<tr>
<td>( t(df) )</td>
<td>( df = (3, 10) )</td>
</tr>
<tr>
<td>( \mathcal{E}(\mu_p, \sigma_p, p) )</td>
<td>( \mu_p = 0, \sigma_p = 0.5, p = (1, 1.5) )</td>
</tr>
</tbody>
</table>
6.1.4 Instances generation design

All instances are generated using three sample sizes \( N = \{80, 160, 240\} \), three values for the number of predictors \( p = \{3, 5, 8\} \) and two values for the number of responses \( q = \{4, 6\} \). The number of levels per predictor and the number of constrained input variables are defined according to the value of the number of predictors. More precisely, if \( p = 3 \) ten levels per predictor and two constrained inputs are generated. If \( p = 5 \), eight levels per predictor and three constrained inputs are generated. If \( p = 8 \), four levels per predictor and four constrained inputs are generated. Within the first approach, the predictor data matrix \( X \) is simulated as described in section 6.1.1 and two models, the first- and second-order models in equations (6.1.2) and (6.1.3), are used to simulate the functions \( f(X) \). Within the second approach, \( X \) is simulated as described in section 6.1.2. Finally, the term \( \epsilon \) is generated as described in section 6.1.3. If draws from a multivariate normal distribution are used, then

\[
\epsilon \sim N_q(0, \Sigma) \tag{6.1.23}
\]

with \( \Sigma \) given by either equation (6.1.13) or equation (6.1.14). If draws from a copula are used, then

\[
\epsilon \sim C(F_1, \ldots, F_q) \tag{6.1.24}
\]

where \( C \) is a Normal copula with unstructured dependence (equation (6.1.19)) and \( F_j \) are the employed marginal CDFs summarized in Table 6.9.

Let the variables of the simulation study be: type of functional structure generating approach (first and second approach), sample size \( N \), number of predictors \( p \), number of responses \( q \), type of employed model in the first approach (first- and second-order model), type of sampling method in the second approach (random and nonrandom), and error structure. Each variable has a certain number of levels (e.g., \( N \) has three levels corresponding to 80, 160 and 240 simulated data). A complete factorial design is used to set up the simulation study and a total of \( 432 = 2 \times 3 \times 3 \times 2 \times 2 \times 2 \times 3 \) instances are generated.
Table 6.10: *Uniresponse modeling techniques investigated in the simulation study.*

<table>
<thead>
<tr>
<th>Uniresponse modeling techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear regression (LR)</td>
</tr>
<tr>
<td>Polynomial regression (PL)</td>
</tr>
<tr>
<td>Ridge regression (RR)</td>
</tr>
<tr>
<td>Partial Least Squares (PLS)</td>
</tr>
<tr>
<td>Multivariate Adaptive Regression Splines (MARS)</td>
</tr>
<tr>
<td>Neural Network (NN)</td>
</tr>
<tr>
<td>Mixed model-based penalized splines (MMPS)</td>
</tr>
</tbody>
</table>

Table 6.11: *Multivariate modeling techniques investigated in the simulation study.*

<table>
<thead>
<tr>
<th>Multivariate modelling techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multivariate ridge regression (MRR)</td>
</tr>
<tr>
<td>Two-block PLSR (PLSR2)</td>
</tr>
<tr>
<td>Multivariate Adaptive Regression Splines (MMARS)</td>
</tr>
<tr>
<td>Multivariate neural network (MNN)</td>
</tr>
<tr>
<td>Reduced rank regression (RRR)</td>
</tr>
<tr>
<td>Curds and Whey (C&amp;W)</td>
</tr>
<tr>
<td>Curds and Whey Ridge Regression (C&amp;W-RR)</td>
</tr>
</tbody>
</table>

6.2 **The algorithm and performance spaces**

A proper algorithm space is composed by a large number of diverse models. Interest is here on separate (uniresponse) and multivariate multiple regression models, where the term multiple refers to the presence of multiple predictors. Whereas separate multiple models are characterized by a quite high diversity, advances in the multivariate field concern mainly linear models and, as clearly described in Breiman & Friedman (1997), the majority have roots in a common technique, namely CCA. Exceptions are the multivariate MARS model and the multivariate neural network. Tables 6.10 and 6.11 list the uniresponse and multivariate models considered for the simulation study. The listed models are described in Chapters 3 and 5. To evaluate the models performance, their mean squared error of prediction (MSEP) is estimated by leave-one-out cross-validation. A measure of the models predictive accuracy is therefore obtained.
6.3 The feature space

The greatest challenge in the meta-learning approach is the derivation of suitable metrics as features to characterize the datasets. Features must be chosen so that the varying complexities of the problem instances are exposed, any known structural properties of the problems are captured, and any known advantages and limitations of the different algorithms are related to features (Smith-Miles 2008). Therefore, the choice of features depends very much on the problem domain, as well as on the chosen models. This section describes the metrics used to characterize the problems, which have been simulated employing the procedures of Section 6.1. The identified metrics are easily calculated, are likely to correlate with models performances and aim at exposing the strengths and weaknesses of the different models. In particular, the use of computationally inexpensive metrics are of interest because sophisticated measurements would require a large amount of CPU time, that otherwise could have been allocated to a sophisticated learner of equal or even better complexity (Pfahringer et al. 2000). New metrics have been devised to characterize the instances of the investigated multiresponse regression domain. The newly developed measures are considered along with the main metrics previously developed to cope with uniresponse regression problems (Smith-Miles 2008, Kopf et al. 2000, Brazdil Pavel 1998, Soares et al. 2004, Amasyali & Ersoy 2009). Some of them have been excluded for this study because of their non immediate relationship with the theory of the investigated models. The metrics considered are here divided into four groups and summarized in four different tables (Tables 6.12-6.15). They are simple measures, statistical measures, information theoretical measures, and model-based measures. A second approach, known as landmarking, is employed for a more direct characterization of the instances. The identified metrics and landmarking are described further below.

Metrics

The simple measures are summarized in Table 6.12. They describe the properties of the problem and capture basic concepts such as size of the dataset and presence of collinearity. To measure the degree of collinearity of a problem, the condition number is used. The condition number is the most popular measure of ill-conditioned of a square matrix $M$ (Izenman 2008), and is given by the ratio of the largest to the smallest nonzero singular
Table 6.12: Simple measures to characterize the properties of the regression problems simulated.

<table>
<thead>
<tr>
<th>Simple measures</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of observations</td>
<td>$N$</td>
</tr>
<tr>
<td>Number of predictors</td>
<td>$p$</td>
</tr>
<tr>
<td>Number of responses</td>
<td>$q$</td>
</tr>
<tr>
<td>Ratio between number of observations and number of predictors</td>
<td>$r = N/p$</td>
</tr>
<tr>
<td>Predictors’ condition number</td>
<td>$k(X^TX)$</td>
</tr>
<tr>
<td>Responses’ condition number</td>
<td>$k(Y^TY)$</td>
</tr>
</tbody>
</table>

value of $M$:

$$k(M) = \frac{\sigma_1}{\sigma_r}.$$  \hspace{1cm} (6.3.1)

Here, $\sigma_i$ is the square-root of the $i$th largest eigenvalue of the matrix $M$ and $M$ is a $r$-dimensional square matrix. Clearly, it always holds that $k \geq 1$. If $M$ is an orthogonal matrix, all singular values are unity, and so $k = 1$. Large values of $k(M)$ indicate that its singular values are widely spread out and so the matrix $M$ is said to be ill-conditioning. Conversely, if $k(M)$ is small, $M$ is said to be well-conditioned. Interest is here on both the predictors’ and responses’ condition number, thus the matrices $M = X^TX$ and $M = Y^TY$ are considered. Before using equation (6.3.1), each variable may be scaled to have equal length. For example, the $i$th predictor $X_i$ should be replaced by $X_i/s_i$, where $s_i$ is the sample standard deviation of the $i$th variable and $i = 1, \ldots, p$ (Izenman 2008). Alternative measures of ill-conditioning are the collinearity indices

$$k_k(M) = \sqrt{VIF_k} \quad k = 1, \ldots, r,$$  \hspace{1cm} (6.3.2)

where $VIF_k = (1 - R_k^2)^{-1}$ is the $k$th variance inflation factor. If, for example, $M = (X^TX)$, $R_k^2$ is the squared multiple correlation coefficient of the $k$th column of $X$ on the other $p - 1$ columns of $X$, for $k = 1, \ldots, p$. Large values of $VIF_k$ (typically, $VIF_k > 10$) imply that $R_k^2$ is close to unity, which in turn suggests near collinearity may be present. The collinearity indices have values at least one and are invariant under scale changes of the columns of $X$ (Izenman 2008). Because a unique value is produced and due to its larger popularity, the condition number has been chosen as the measure of problem ill-conditioning.
The statistical measures are summarized in Table 6.13. They describe the properties of both the variables and the observations in the dataset. They include basic summary statistics, distance-based measures, and metrics based on data projection methods (e.g., PCA and CCA). Basic summary statistics can be simple functions such as average, minimum and maximum. However, because these metrics are calculated for each variable in the dataset and because datasets with different dimensionalities are studied, the use of the aforementioned simple metrics would result in a different amount of features per dataset. Indeed, to model the functional relationship between instance features and models performances a meta-dataset is generated. The rows of the meta-dataset identify the instances (meta-observations) and the columns identify the features (meta-variables). Such a dataset constitutes the predictors' matrix, while the response matrix is composed of instances in the rows and models performances in the columns. A predictors' matrix with different number of variables per observation can be a problem for regression and classification algorithms. To cope with this difficulty, features can be derived from histograms. Amasyali & Ersoy (2009) built meta-features using histogram's bin and frequency values, as well as the shape of the histograms. A constant number of histogram's bin values has been used. Instead, mean, standard deviation and quartiles are used here to characterize the histograms of the metrics summarized in Table 6.13. Correlation-based measures in Table 6.13 aim to detect linear dependencies within the predictors, within the responses, and also between predictors and responses. Because perfectly correlated variables are redundant, highly correlated variables are not expected to generate additional information (Guyon & Elisseeff 2003). Therefore, measuring the proportion of highly correlated variables, can give an idea of the amount of redundant information in the problem. The tests for association between paired samples, are carried out using one of Pearson's product moment correlation coefficient, Kendall’s tau or Spearman’s rho, in order to identify highly correlated variables. When possible, metrics assuming values in the interval $[0,1]$ are generated to simplify their comparison. Distance-based metrics are employed to study the distance between either observations or variables of a problem. They employ dissimilarity measures such as the Euclidean distance and a correlation-based metric. The Euclidean distance can be used to measure the distance between either two observations or two
variables in a matrix $A$. It is defined as

$$d_E(x_i, x_j) = [(x_i - x_j)^T(x_i - x_j)]^{1/2} = \left[ \sum_{k=1}^{r} (x_{ik} - x_{jk})^2 \right]^{1/2} \quad (6.3.3)$$

where $x_i = (x_{i1}, \ldots, x_{ir})^T$ and $x_j = (x_{j1}, \ldots, x_{jr})^T$ are two points in $\mathbb{R}^r$. The distances between the observations are measured to give an indication of the problem difficulty in terms of sparsity of the observed points. Instead, the distances between the variables suggest their degree of dissimilarity (closeness); the smaller the distances the closer the variables. The correlation-based dissimilarity, measures the distance between two variables $X_i$ and $X_j$, and is given by

$$d_{cor}(x_i, x_j) = 1 - \rho_{ij} \quad (6.3.4)$$

where $-1 \leq \rho_{ij} \leq 1$ is the correlation between the pair of variables. A relatively large absolute value of $\rho_{ij}$ suggests the variables are ‘close’ to each other, whereas a small correlation ($\rho_{ij} \approx 0$) suggests the variables are ‘far away’ from each other (Izenman 2008). The dissimilarity measures in equations (6.3.3) and (6.3.4) are all computed using standardized data. Let $D_E(A)$ be the matrix of the Euclidean distances between the observations in $A$, and let $D_E(A^T)$ be the matrix of the Euclidean distances between the variables in $A$. Also, let $D_{cor}(A^T)$ be the matrix of correlation-based distances between the variables in $A$. Simple statistical measures such as mean, standard deviation and quartiles can then be calculated on the following metrics:

$$\frac{D_E(A)}{\max(D_E(A))}, \quad \frac{D_E(A^T)}{\max(D_E(A^T))} \quad \text{and} \quad D_{cor}(A) \quad (6.3.5)$$

A statistical metric has been developed to estimate the amount by which the training error underestimate the true error. The training error is the average loss over the training sample and is given by:

$$\overline{err} = \frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2. \quad (6.3.6)$$

Because the same data is being used to fit the model and assess its error, the quantity $\overline{err}$ in equation (6.3.6) will be an overly optimistic estimate of the true error. Therefore, the training error is not a good estimate of the test error, as it does not properly account for the model complexity (Hastie et al. 2009). Indeed, the training error consistently
decreases with the model complexity, typically dropping to zero if we increase the model complexity enough. However, a model with zero training error is overfit to the training data and will typically generalize poorly (Hastie et al. 2009). It is shown that the average optimism over multiple training sets is given by:

$$\omega = \frac{2}{N} \sum_{n=1}^{N} \text{Cov}(\hat{y}_n, y_n)$$  \hspace{1cm} (6.3.7)

where Cov indicates the covariance. Thus the amount by which $\bar{\text{err}}$ underestimate the true error depends on how strongly $y_n$ affects its own prediction. The harder we fit the data, the greater Cov($\hat{y}_n, y_n$) will be, thereby increasing the optimism. The formula in equation (6.3.7) can be used to obtain a meta-feature which measures the optimism obtained when fitting a simple linear model. A high value of $\omega$ indicates the necessity of hardly correct the estimated training error. Because multiple responses are investigated, multiple covariance values are obtained. Therefore, various metrics are used, including the number of statistically significant covariances Cov($\hat{y}_n, y_n$), as well as mean, standard deviation and quartiles of:

$$\text{Cov}(\hat{Y}, Y) = \frac{E(\text{Cov}(\hat{Y}, Y))}{\max \text{Cov}(\hat{Y}, Y)}$$  \hspace{1cm} (6.3.8)

can be used. The class of statistical measures considers also metrics derived from low-dimensional data projection methods, such as principal component analysis (PCA) and canonical correlation analysis (CCA) described in chapter 5. PCA aims to maximize the variance of the projection of a multivariate data matrix (Hardle & Simar 2007). Because PCA is not scale invariant, it is reasonable to standardize the measurement units. A measure of how well the first $q$ principal components (PCs) explain the total variation is given by the relative proportion:

$$I_q = \frac{\sum_{j=1}^{q} \lambda_j}{\sum_{j=1}^{p} \lambda_j}.$$  \hspace{1cm} (6.3.9)

Furthermore, PCA can be used to provide an idea of the amount of shrinkage produced by RR. Indeed, RR shrinks low-variance directions more than it does high-variance directions (Hastie et al. 2009). As a consequence, a greater amount of shrinkage is applied to the PCs
with smaller variance. The motivation behind this affirmation is as follows. Let

\[ X = UDV^T \]  

be the singular value decomposition of the centered input matrix \( X \). Here \( U \) and \( V \) are \( n \times p \) and \( p \times p \) orthogonal matrices, with the columns of \( U \) spanning the column space of \( X \), and the columns of \( V \) spanning the row space. \( D \) is a \( p \times p \) diagonal matrix, with diagonal entries \( d_1 \geq d_2 \geq \ldots \geq d_p \geq 0 \), called the singular values of \( X \). If one or more values of \( d_j = 0 \), \( X \) is singular. Using the SVD the ridge solutions can be written in the following form:

\[ \hat{X}_{\lambda} = \sum_{i=1}^{p} \frac{u_i d_i}{d_i^2 + \lambda} u_i^T y, \]  

(6.3.11)

where the \( u_i \) are the columns of \( U \). Since \( \lambda \geq 0 \), \( d_i^2/(d_i^2 + \lambda) \leq 1 \). Like linear regression, ridge regression computes the coordinates of \( y \) with respect to the orthonormal basis \( U \). It then shrinks these coordinates by the factors \( d_i^2/(d_i^2 + \lambda) \). A greater amount of shrinkage is therefore applied to the coordinates of basis vectors with smaller \( d_i^2 \) (Hastie et al. 2009). Similarly, a small value of \( q \) in equation (6.3.9) indicates a large number of low-variance directions, and so uniresponse RR is expected to generate a large amount of shrinkage. CCA aims to maximize the association (measured by correlation) between the low-dimensional projections of two data sets. As shown in Breiman & Friedman (1997), it constitutes the foundations of most multivariate techniques. Mean, standard deviation and quartiles of the canonical correlations and of the linear combination coefficients are expected to correlate with the performance of some multivariate multiple regression techniques.

The information theory measures are summarized in Table 6.14, and they include entropy and mutual information. The entropy of a discrete random variable \( Y \) was introduced by Shannon (1948) and is as follows

\[ H(Y) = -\sum_{n=1}^{N} p_Y(y_n) \log p_Y(y_n). \]  

(6.3.12)
Table 6.13: Statistical measures which have been selected to characterize the properties of the variables in the regression problems simulated. \(sc(M)\) indicates the number of correlations of a matrix \(M\), that are significant according to statistical test of association between paired samples. \(p\) denotes the number of predictors and \(q\) the number of responses. \(r\) indicates the rank of the predictors covariance matrix \(X^T X\), while \(s\) indicates the rank of the responses covariance matrix \(Y^T Y\).

<table>
<thead>
<tr>
<th>Statistical measures</th>
<th>(X_i)</th>
<th>(Y_j)</th>
<th>(\text{cor}(X))</th>
<th>(\text{cor}(Y))</th>
<th>(\text{cor}(X, Y))</th>
<th>(s(X))</th>
<th>(s(Y))</th>
<th>(k(X))</th>
<th>(k(Y))</th>
<th>(D_{\text{E}}(X))</th>
<th>(D_{\text{E}}(Y))</th>
<th>(D_{\text{C}}(X^T))</th>
<th>(D_{\text{C}}(Y^T))</th>
<th>(\text{Cov}(\hat{Y}, Y))</th>
<th>(\text{cc}(X, Y))</th>
<th>(\lambda_1)</th>
<th>(\lambda_i)</th>
<th>(\lambda_1)</th>
<th>(\lambda_i)</th>
<th>(\text{sw}(Y))</th>
<th>(\text{sw}(Y))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single predictors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single responses</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correlation of X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correlation of Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correlation of X and Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predictors’ skewness values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responses’ skewness values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predictors’ kurtosis values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responses’ kurtosis values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normalized Euclidean distances of observations in X</td>
<td>(D_{\text{E}}(X))</td>
<td>(\max D_{\text{E}}(X))</td>
<td>(\max D_{\text{E}}(X^T))</td>
<td>(\max D_{\text{E}}(Y))</td>
<td>(\max D_{\text{E}}(Y^T))</td>
<td>(\text{Cov}(\hat{Y}, Y))</td>
<td>(\text{cc}(X, Y))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normalized Euclidean distances of variables in X</td>
<td>(D_{\text{E}}(X))</td>
<td>(\max D_{\text{E}}(X))</td>
<td>(\max D_{\text{E}}(X^T))</td>
<td>(\max D_{\text{E}}(Y))</td>
<td>(\max D_{\text{E}}(Y^T))</td>
<td>(\text{Cov}(\hat{Y}, Y))</td>
<td>(\text{cc}(X, Y))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normalized Euclidean distances of observations in Y</td>
<td>(D_{\text{E}}(Y))</td>
<td>(\max D_{\text{E}}(Y))</td>
<td>(\max D_{\text{E}}(Y^T))</td>
<td>(\max D_{\text{E}}(Y))</td>
<td>(\max D_{\text{E}}(Y^T))</td>
<td>(\text{Cov}(\hat{Y}, Y))</td>
<td>(\text{cc}(X, Y))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correlation-based distances of X</td>
<td>(D_{\text{C}}(X^T))</td>
<td>(\frac{\text{Cov}(\hat{Y}, Y)}{\max \text{Cov}(\hat{Y}, Y)})</td>
<td>(\text{cc}(X, Y))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correlation-based distances of Y</td>
<td>(D_{\text{C}}(Y^T))</td>
<td>(\frac{\text{Cov}(\hat{Y}, Y)}{\max \text{Cov}(\hat{Y}, Y)})</td>
<td>(\text{cc}(X, Y))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normalized covariance values (\text{Cov}(\hat{Y}, Y))</td>
<td>(\max \text{Cov}(\hat{Y}, Y))</td>
<td>(\text{Cov}(\hat{Y}, Y))</td>
<td>(\text{cc}(X, Y))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canonical correlations</td>
<td>(\text{cc}(X, Y))</td>
<td>(\text{cc}(X, Y))</td>
<td>(\text{cc}(X, Y))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coefficients of the X linear combinations</td>
<td>(\lambda_1)</td>
<td>(\lambda_i)</td>
<td>(\lambda_1)</td>
<td>(\lambda_i)</td>
<td>(\text{cc}(X, Y))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coefficients of the Y linear combinations</td>
<td>(\lambda_1)</td>
<td>(\lambda_i)</td>
<td>(\lambda_1)</td>
<td>(\lambda_i)</td>
<td>(\text{cc}(X, Y))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Other statistical measures

<table>
<thead>
<tr>
<th>Ratio of (sc(X)) and (p)</th>
<th>(sc(X)/p)</th>
<th>(sc(X)/p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio of (sc(Y)) and (q)</td>
<td>(sc(Y)/q)</td>
<td>(sc(Y)/q)</td>
</tr>
<tr>
<td>Ratio of (sc(X, Y)) and (p \cdot q)</td>
<td>(sc(X, Y)/(p \cdot q))</td>
<td>(sc(X, Y)/(p \cdot q))</td>
</tr>
<tr>
<td>Relative importance of the largest eigenvalue for X</td>
<td>(I(X)<em>1 = \sum</em>{i=1}^{\lambda_1} \lambda_i)</td>
<td>(I(Y)<em>1 = \sum</em>{i=1}^{\lambda_1} \lambda_i)</td>
</tr>
<tr>
<td>Relative importance of the largest eigenvalue for Y</td>
<td>(I(X)<em>1 = \sum</em>{i=1}^{\lambda_1} \lambda_i)</td>
<td>(I(Y)<em>1 = \sum</em>{i=1}^{\lambda_1} \lambda_i)</td>
</tr>
<tr>
<td>% of PCs for X to explain the 95% of the variation</td>
<td>(v) % s.t. (I(X)_v = 95%)</td>
<td>(v) % s.t. (I(Y)_v = 95%)</td>
</tr>
<tr>
<td>% of PCs for Y to explain the 95% of the variation</td>
<td>(u) % s.t. (I(Y)_u = 95%)</td>
<td>(u) % s.t. (I(Y)_u = 95%)</td>
</tr>
<tr>
<td>Shapiro-Wilk test of Normality for Y</td>
<td>(\text{sw}(Y))</td>
<td>(\text{sw}(Y))</td>
</tr>
<tr>
<td>Shapiro-Wilk test of Normality for (E) (after linear fitting)</td>
<td>(\text{sw}(Y))</td>
<td>(\text{sw}(Y))</td>
</tr>
</tbody>
</table>
Table 6.14: Information theoretical measures which have been identified to capture the randomness of the regression problems which have been simulated. Here, $i = 1, \ldots, p$ and $j = 1, \ldots, q$.

<table>
<thead>
<tr>
<th>Information theoretical measures</th>
<th>$\mathcal{H}(X_i)$</th>
<th>$\mathcal{H}(Y_j)$</th>
<th>$\mathcal{H}(X)$</th>
<th>$\mathcal{H}(Y)$</th>
<th>$I(X_i)$</th>
<th>$I(Y_j)$</th>
<th>$I(X)$</th>
<th>$I(Y)$</th>
</tr>
</thead>
</table>

If $Y$ is a continuous random variable with probability density function $p_Y(y)$, then the entropy of $Y$ is defined by

$$
\mathcal{H}(Y) = - \int p_Y(y) \log p_Y(y) \, dy
$$

and is referred to as differential or continuous entropy (Lazo & Rathie 1978). The entropy gives a notion of how much information is contained in $Y$ and is largest when $Y$ has greatest variance (e.g., when $Y$ is most unpredictable). The largest value of $\mathcal{H}(Y)$ occurs when $Y$ has a Gaussian distribution, while small values of $\mathcal{H}(Y)$ occur when the distribution of $Y$ is concentrated on specific values (Izenman 2008). The mutual information is a measure of dependence between the components of the random vector $Y$, and is defined as

$$
I(Y) = \sum_{i=1}^{p} \mathcal{H}(Y_i) - \mathcal{H}(Y).
$$

The quantity $I(Y)$ in equation (6.3.14) is called the Kullback-Leibler distance between the density $p_Y(y)$ and its independence version $\prod_{i=1}^{p} p_{Y_i}(y_i)$, where $p_{Y_i}(y_i)$ is the marginal density of $Y_i$ (Hastie et al. 2009). Information theory measures can capture the randomness of an instance. Since an instance that appears quite random has few internal structures that can be exploited by a model in finding a solution, the randomness of an instance correlates with complexity (Smith-Miles 2008).
**Table 6.15**: Model-based measures to include the structural characteristics of decision tree models for each regression problem simulated.

<table>
<thead>
<tr>
<th>Measures based on decision tree model</th>
<th>( T_{\text{nodes}} )</th>
<th>( T_{\text{leaves}} )</th>
<th>( T_{\text{splits}} )</th>
<th>( T_{\text{size}} )</th>
<th>( T_{\text{depth}} )</th>
<th>( T_\epsilon )</th>
<th>( T_{X\text{nodes}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of leaves</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of splits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tree size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Depth of the tree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tree prediction error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of variables with multiple nodes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Model-based measures* are typically obtained from decision tree models. Multivariate regression trees that allow for multiple responses are considered here, and the developed metrics are summarized in Table 6.15. The R package `mvpart` is used (Therneau & Atkinson 2007).

**Landmarking**

Landmarking was developed as an alternative to generating computationally expensive features and has become a key direction in meta-learning research (Smith-Miles 2008). With this approach, the performance of simple and efficient learners (models) are calculated on a dataset with the purpose of determining the location of such a problem in the space of all problems (Kopf et al. 2000, Pfahringer et al. 2000). This is achieved by measuring directly the performance of simple and efficient learning algorithms (models) on the specific problem so as to identify areas of the learning space on which the learner performs specially well (Pfahringer et al. 2000). Differently from previously described features, landmarking tries to characterize a domain by directly relating the performance of some learners, the landmarkers, to the performance of some other algorithm (Pfahringer et al. 2000). The basic idea is to predict the models performance relative to the performance of simpler and faster learners (Smith-Miles 2008). Indeed, the performance of simple learners on a special problem might give an indication on the performance of other learners (Kopf et al. 2000). A simple and fast learner in regression is the linear model. With landmarking, the \( R^2 \) of the model, its MSEP and statistical measures on its residuals
Table 6.16: Landmarking to generate computationally expensive features and evaluate the performance of simple and fast learners (models). Here, $j = 1, \ldots, q$.

<table>
<thead>
<tr>
<th>Landmarking - linear model</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model coefficient of determination</td>
<td>$R^2(\text{lm})$</td>
</tr>
<tr>
<td>Model predictive accuracy</td>
<td>$\text{MSEP}(\text{lm})$</td>
</tr>
<tr>
<td>Mean, standard deviation and quartiles of the elements of $\epsilon$</td>
<td>$\epsilon_j$</td>
</tr>
</tbody>
</table>

can be used to predict the performance of more complex models. The performance and statistical measures employed here for landmarking, are summarized in Table 6.16.

6.4 Concluding remarks: ongoing and future research

After the seminal work by Wolpert & Macready (1997) describing the so-called no free lunch theorems, it is now generally accepted that no single learning algorithm can dominate some other algorithm over all possible learning problems (Pfahringer et al. 2000). Of interest is therefore the selection of the algorithm that is more likely to perform best for a given problem (algorithm selection problem). As recognized by Rice (1976), algorithm and model selection are basically the same concepts. Therefore, in this context, the terms ‘algorithm’ and ‘model’ have the same meaning. Model selection is typically performed experimentally, that is evaluating the performance of a large number of models on the given problem and selecting the best candidate according to a predefined statistical criteria (e.g., AIC, BIC, cross-validation) (Brazdil Pavel 1998, Hastie et al. 2009). In the real-world applications, the approach of simply comparing the applicable models is just not possible. Due to resource limitations, only a handful of models can ever be applied to a specific problem. Furthermore, usually several different representations are reasonable for a given application, which adds a further dimension to this already vast search space for the optimal model (Pfahringer et al. 2000). A somewhat more sophisticated approach is to describe different classes of problems and search for correlations between these different classes and the different optimal models (Pfahringer et al. 2000). This approach is the one adopted by the meta-learning framework which has been described in this chapter. The developed meta-learning framework aims to acquire some rules, to be used as a guidance when faced with a new application. It consists of building meta-rules...
matching domain characteristics and model performance to discover the conditions under which a model outperforms others. Such a matching can be achieved using clustering, classification or regression models including decision trees, as well as linear, piece-wise linear, or polynomial models, and neural networks (Smith-Miles 2008, Gama & Bradzil 1995). The meta-learning framework developed in this chapter has laid the theoretical foundation for a large computational study. Such a computational study is part of the ongoing research, and is expected to provide meta-rules of the following type:

- $C&W$ recommended if $(q < 10)$ and $(E(Cov(\hat{Y}, Y)) \geq 0.6)$

- $\text{MSEP}(\text{RRR}) = 5.7 + 0.72 \cdot R^2 - 3 \cdot k(X_3)$

where, $q$ is the number of responses, $R^2$ is the coefficient of determination of a simple linear model, and $k(X_3)$ is the condition number of the third predictor. Successful meta-rules are expected to capture regularities of the encountered situations that allow to relate a model with those domains in which it performs well (Brazdil Pavel 1998). Such meta-rules should provide a knowledge, which is useful to select the model that is more likely to outperform others and to simultaneously gain understanding about why a specific model is more suitable to a problem with certain features. The purpose of future research is to seek insight, explanation, and understanding of the obtained empirical rules. This examination will be made from a theoretical perspective with a view to confirm the sense of the rules, as well as generating insights into models behaviour. Therefore, existing models can knowledgeably be adapted, or new models developed, to extend the classes of problems on which we can expect models to perform well (Smith-Miles 2008).
Conclusions

The work in this thesis dealt with the experimental design and optimization of high dimensional systems with multiple responses. The aim was to support the practitioner in facing the challenging aspects of this type of system, which is very common in real-world applications, by developing an approach capable of planning the experiments with a view to optimize the system responses. Because the same purpose is pursued by the response surface methodology (RSM), this technique was also studied. In particular, nonparametric and semiparametric RSM (NPRSM) and design and analysis of computer experiments (DACE) were considered because they were more suitable to the high dimensionality and nonlinearity of the systems of interest. Attention was restricted to one-stage methods because of their ability to cope with highly deceptive problems.

In this thesis, a one-stage distribution-free approach, named evolutionary model-based multiresponse approach (EMMA), was developed. It employed an initial random design, which was subsequently augmented by identifying additional design points under the guidance of a particle swarm optimization (PSO) algorithm with time-varying parameters. At each step of the procedure, a multiresponse multivariate adaptive regression splines (MARS) model was fitted to the collected data and used to select a target. The target is a set of response values obtained by separately optimizing the fitted MARS functions. A distance-based metric was then employed to minimize the distance between the values of the measured responses and the identified target. The advantages of the proposed approach derived from its one-stage distribution-free nature, its capability of automatically selecting the target (which is typically unknown) and its ability to converge to the true global optimum, as demonstrated by the empirical study on a set of standard benchmark functions. When applied to the chemical problem that motivated the research, EMMA
identified an experimental point (chemical composition) resulting in a significantly better output (functionalized coating) compared to the best coating found using the methodology typically employed by the scientists. Of interest was also the comparison of the developed approach with other methodologies (e.g., NPRSM, DACE, evolutive optimization procedures). However, it was difficult to find published works that tested the procedure on standard benchmark functions and reported the results in terms of the number of function evaluations (required experiments). This is an area we need to further address.

To improve the one-stage approach, the modeling step was investigated and a meta-learning framework for model selection was developed. A collection of multiresponse modeling techniques were identified to evaluate their performance on a set of problems and to relate such performances to the characteristics of the studied problems. Simulated data were generated because a large number of problems (instances), with different degrees of complexities, were required and not enough multiresponse real datasets were available. Two procedures were developed to simulate the datasets. Their objective is twofold. First, generate instances with various levels of complexity and, second, produce problems that share similarities with the observed chemical data (data collected applying the EMMA to the chemical study). Both standard and non-standard error structures were generated using the multivariate normal distribution and a copula-based technique, respectively. A difficulty encountered in simulating the datasets derived from the different magnitude of the signal term, $f(X)$, and the error term, $\epsilon$. In particular, error values too high with respect to the signal values, resulted in datasets that were too noisy because the signal was ‘hidden’ by the noise. In contrast, error values too small with respect to the signal values resulted in datasets with an insufficient level of noise. To overcome this drawback, the signal to noise ratio was fixed and kept constant. Finally, a set of metrics was developed to measure the features of the simulated instances. The metrics were identified so as to satisfy the following criteria: easy to calculate, likely to correlate with models performances, and able to expose the strengths and weaknesses of the different models. Therefore, the foundations of a large computational study were laid.

Implementation of the meta-learning framework is part of the ongoing research. The final output will consist of meta-rules relating problem characteristics and model performances. Successful meta-rules are expected to capture regularities of the encountered situations
and to provide some guidance when faced with a new application. The purpose of future research is to seek insight, explanation, and understanding of the meta-rules with the aim to generate insights into each model's behaviour and eventually adapt existing models to improve their performance in certain classes of problems. Another future direction of interest is the study of Pareto-based multiresponse optimization procedures to be integrated with EMMA. This would allow the identification of a set of solutions, for which no response can be improved without detracting from at least one other response. Therefore, according to the relevance of the investigated responses, the practitioner would have the possibility to choose, a posteriori, between equivalent optimal solutions.

In conclusion, in many applicative fields (e.g., science and engineering) practitioners often have to face systems characterized by challenging aspects such as high dimensionality of the input space, multiple responses, constraints on the input variables, unknown functional relationship, time and budget limitations. This creates the necessity of developing experimental design techniques to guide the practitioner and gain knowledge about the system with the final aim to optimize it. Moreover, the use of model-based optimization techniques speeds up the search procedure, thus reducing the number of experiments required. The results presented in the thesis contribute to illustrate these aspects, both from an empirical and applied point-of-view.
Bibliography


URL: [http://dx.doi.org/10.1109/TEVC.2007.896686](http://dx.doi.org/10.1109/TEVC.2007.896686)


**URL:** http://CRAN.R-project.org/package=earth


**URL:** http://www.R-project.org


URL: [http://CRAN.R-project.org/package=AlgDesign](http://CRAN.R-project.org/package=AlgDesign)


Wold, H. (1966), ‘Soft modeling by latent variables; the nonlinear iterative partial least squares approach’, *Perspectives in Probability and Statistics. Papers in Honour of M. S.*


**URL:** [http://dx.doi.org/10.1109/TSMCC.2005.855506](http://dx.doi.org/10.1109/TSMCC.2005.855506)