DATA MINING METHODOLOGIES FOR SUPPORTING ENGINEERS DURING SYSTEM IDENTIFICATION

THÈSE N^O 4056 (2008)

PRÉSENTÉE LE 15 MAI 2008

À LA FACULTÉ DE L'ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT
LABORATOIRE D'INFORMATIQUE ET DE MÉCANIQUE APPLIQUÉES À LA CONSTRUCTION
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Sandro SAITTA

ingénieur informaticien diplômé EPF
de nationalité suisse et originaire de Bavois (VD)

acceptée sur proposition du jury:

Prof. B. Moret, président du jury
Prof. I. Smith, Dr B. Raphael, directeurs de thèse
Prof. B. Faltings, rapporteur
Prof. P. Struss, rapporteur
Dr E. Viennet, rapporteur

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2008

برای رعناگل زندگی من

Acknowledgments

This work was funded by the Swiss National Science Foundation under grant #200020-109257.

My first acknowledgment goes to my co-advisor, Prof. Ian Smith, who was present during my PhD for guiding my work. He also was a valuable person for explaining to me crucial aspects of thesis work. Benny Raphael, my other co-advisor and an Assistant Prof. at the National University of Singapore, followed my thesis from the very beginning to the end. Working in an autonomous manner during four years is not always straightforward. When I faced an obstacle, Benny was always there to help me. I will always remember a sentence he wrote to me: "*Remember, you can make something good come out of everything if you do it in the right spirit*". Prakash Kripakaran, a Post doc. researcher at EPFL, is also the kind of person that you only meet once in your life. I think my work would never have reached the present state without his help. I had a lot of fruitful discussions with him regarding difficult issues in my research. Instead of only giving a simple answer, Prakash always suggested a new way to face a particular problem. François Fleuret, a researcher at IDIAP, is an expert in machine learning. I had a lot of general discussions with him about data mining and it was really good food for thought. He is also the first person who made me understand what research really is about. I also thank examiners for their time and interest in my work: Prof. Boi Faltings (EPFL), Prof. Younès Bennani (Université Paris 13) and Prof. Peter Struss (TUM).

I would also like to thank the following people: Marco Viviani for the nice time we spent in the office; Suraj Ravindran for the good time playing table-tennis and discussing various subjects; Bernard Adam for good time we spent in conferences and skiing; Mylène Devaux for many interesting discussions, Bernd Domer for his relevant advice; Landolf Barbarigos for his good mood and his interesting questions; Sylvain Demierre for discussions on Sci-Fi; Alain Herzog for taking me as a model for his pictures; Sahar Hosseinian for her help. I would also like to thank my colleagues or former colleagues for their advice, proofreading, comments or some help on my work: Hugo Pelletier, Youssef Belmouden, Yvan Robert-Nicoud, Pierino Lestuzzi, Charles Gilliard, Jean-Louis Guignard, Patrice Gallay, Christian Greifenhagen, Francine Laferrière, Cendrine Pons and Ozlem Tayfur.

Let me finally acknowledge my family and all my friends for supporting me and showing interest in my work. Although the question "*how is your thesis going?*" seems usual, it is always a pleasure to answer by explaining your ongoing projects. I thank my mother Dominique and my father Luigi for believing in me. I have kept the most important for the end: thanks to Rana, the flower of my life, for her support and listening.

Abstract

Data alone are worth almost nothing. While data collection is increasing exponentially worldwide, a clear distinction between retrieving data and obtaining knowledge has to be made. Data are retrieved while measuring phenomena or gathering facts. Knowledge refers to data patterns and trends that are useful for decision making. Data interpretation creates a challenge that is particularly present in system identification, where thousands of models may explain a given set of measurements. Manually interpreting such data is not reliable. One solution is to use data mining. This thesis thus proposes an integration of techniques from data mining, a field of research where the aim is to find knowledge from data, into an existing multiple-model system identification methodology.

It is shown that, within a framework for decision support, data mining techniques constitute a valuable tool for engineers performing system identification. For example, clustering techniques group similar models together in order to guide subsequent decisions since they might indicate possible states of a structure. A main issue concerns the number of clusters, which, usually, is unknown.

For determining the correct number of clusters in data and estimating the quality of a clustering algorithm, a score function is proposed. The score function is a reliable index for estimating the number of clusters in a given data set, thus increasing understanding of results. Furthermore, useful information for engineers who perform system identification is achieved through the use of feature selection techniques. They allow selection of relevant parameters that explain candidate models. The core algorithm is a feature selection strategy based on global search.

In addition to providing information about the candidate model space, data mining is found to be a valuable tool for supporting decisions related to subsequent sensor placement. When integrated into a methodology for iterative sensor placement, clustering is found to provide useful support through providing a rational basis for decisions related to subsequent sensor placement on existing structures. Greedy and global search strategies should be selected according to the context. Experiments show that whereas global search is more efficient for initial sensor placement, a greedy strategy is more suitable for iterative sensor placement.

Keywords: data mining, machine learning, correlation, PCA, clustering, K-means, cluster validity, feature selection, PGSL, SVM, system identification, decision support, sensor placement, measurement system design.

Résumé

Les données seules n'ont presque aucune valeur. Alors que la récolte de données augmente de manière exponentielle à travers le monde, il est important de bien distinguer la récolte de données de l'obtention de la connaissance. Les données sont récoltées en mesurant certains phénomènes ou en rassemblant des faits. La connaissance, quant à elle, se réfère aux tendances présentes dans les données qui permettent de prendre certaines décisions.

L'interprétation des données est un problème important dans le cadre de l'identification de systèmes, un domaine qui consiste à trouver l'état d'une structure (représenté par des modèles) à partir de mesures. En effet, lors de l'identification de systèmes, plusieurs milliers de modèles peuvent expliquer un ensemble de mesures. Il n'est pas possible d'interpréter manuellement ces modèles. Une solution est d'utiliser le *data mining*, un domaine de recherche qui a pour but de trouver de la connaissance à partir des données. Cette thèse propose ainsi d'intégrer des techniques de *data mining* dans une méthodologie existante d'identification de systèmes.

Cette thèse montre que l'aide à la décision, fournie par l'utilisation des techniques de *data mining*, est précieuse pour les ingénieurs en charge de l'identification de systèmes. Par exemple, le *clustering* permet de grouper les modèles similaires (représentant les différents états possibles d'une structure) pour aider dans la prise de décisions. Le problème principal concerne le nombre de *clusters* (groupes) qui est inconnu.

Pour estimer le nombre de *clusters* dans un ensemble de modèles, et juger de la qualité de l'algorithme de *clustering*, un indice de validité est proposé. Cet indice de validité permet d'estimer de manière fiable le nombre de *clusters* dans un ensemble de données. Dans le cas de l'identification de systèmes, cet indice aide les ingénieurs à comprendre les résultats d'un algorithme de *clustering*. De plus, une technique de sélection de paramètres (des modèles), combinant deux algorithmes existant, est développée. La technique proposée permet de sélectionner un faible nombre de paramètres qui différencient les modèles représentant correctement la structure des autres modèles.

Les techniques de *data mining* sont aussi de précieux outils pour la prise de décisions concernant l'ajout de capteurs. Lorsqu'il est integré dans une methodologie de placement de capteurs, le *clustering* se trouve être un outil capable d'aider les ingénieurs à ajouter des capteurs supplémentaires sur une structure existante. Les expériences effectuées montrent que le choix de la stratégie de placement de capteurs dépend de l'utilisation du système de mesure.

Mots-clés: fouille de donnèes, apprentissage automatique, corrélation, analyse en composantes principales, groupement de données, sélection de paramètres, identification de système, aide à la décision, placement de capteurs.

Notation

$cov(x, y)$	Covariance between variable x and y
$corr(x, y)$	Correlation between variable x and y
$\overline{x}$	Mean of vector x
s_{ij}	Covariance between p_i and p_j
p_i	Parameter i
A^T	Transpose of matrix A
$d(x_i, x_j)$	Euclidean distance between x_i and x_j
ε	Error which is calculated as the difference between predictions γ_i and measurements m_i
τ	Threshold value evaluated from measurement and modeling errors in the identification process
m_i	Measurement at i
γ_i	Prediction at i
e_{meas}	Measurement error (difference between real and measured quantities in a single measurement)
e_{mod}	Modeling error (difference between the prediction of a given model and that of the model that accurately represents the real behavior)
e_1	Error due to the discrepancy between the behavior of the mathematical model and that of the real structure
e_2	Error due to the numerical computation of the solution of the partial differential equations representing the mathematical model
e_3	Error due to the assumptions that are made during the simulation of the numerical model
d	Dimensionality of a data set (also number of parameters)
n	Number of points
n_i	Number of points in cluster c_i
z_i	Centroid of cluster c_i
k	Number of clusters
α	Empirical value for unique cluster detection
$O(n)$	"Big O" notation for a complexity of the order of n
μ	Mean
σ	Standard deviation
bcd	Between class distance
wcd	Within class distance
$K(x_i, x_j)$	Kernel function between x_i and x_j

λ_i	i-th Lagrange multiplier
C	SVM tuning parameter representing the penalty of misclassifying training examples
P_i	i-th probability
$H(X)$	Entropy of variable X

Contents

Chapter

1

Introduction

"*The capacity of digital data storage worldwide has doubled every nine months for at least a decade, at twice the rate predicted by Moore's Law for the growth of computing power during the same period.*" (Fayyad and Uthurusamy, 2002)

Overview

This chapter introduces the context of the thesis. It briefly describes related topics such as data mining, system identification and sensor placement. The last section presents research questions as well as the research methodology for achieving the objectives of this thesis.

1.1 Context

Data alone is worth almost nothing. While data is increasing exponentially, people in some fields are "starving" for knowledge. In spite of this, the gap between data and knowledge may be huge. These days, the meaning of the word data is often confused with knowledge. Knowledge is obtained through the understanding of data. The amazing increase in data worldwide brings several challenges. The more the amount of data, the more difficult it is to understand. It is sometimes assumed that the increase of knowledge is proportional to the increase of data. The reason for such an assertion might be the lack of appreciation of the difference between obtaining and understanding data.

Increase of data is a challenge that is particularly present in engineering. The number of sensors is increasing while costs are decreasing. In many domains, engineers are saturated with data of many types. A good example of such a task is model-based diagnosis (de Kleer and Williams, 1987) and system identification (Ljung, 1999). Recently, a new methodology (Robert-

Nicoud, 2003) has been developed in which system identification is treated as a constraint satisfaction problem (CSP) instead of the more traditional optimization problem. This approach results in a set of several candidate models instead of a single model.

When there are many models, engineers need sophisticated tools to interpret them. Data mining (Tan et al., 2006) may provide help. Data mining techniques are used for the task of identifying characteristics of candidate models. Better system identification is possible by integrating data mining into the overall process. No work has been done on *mining* models. More specifically, data mining techniques have never been used for identifying characteristics of candidate models that explain observations (Chapter 2 provides more details). The present work is an attempt to fill this knowledge gap by developing an overall methodology for multiple-model system identification that integrates data mining to provide support for engineers.

1.2 Data Mining

Data mining techniques are becoming important in the context of the increasing trend in data worldwide as explained in Section 1.1. There are more and more sensors capturing changes in our environment and our infrastructure. Therefore, a growing challenge involves determining the meaning of data. As written in Piatetsky-Shapiro (2007), "*[...] as long as the world keeps producing data of all kinds [...] at an ever increasing rate, the demand for data mining will continue to grow.*"

Data mining is a field which is concerned with understanding data. In other words, the aim is to look for patterns in data (Pal and Mitra, 2004). As this pattern may be very difficult to find, it is sometimes compared to gold mining in rivers (Figure 1.1); gravel represents the enormous amount of data and gold nuggets are the hidden patterns to find.

Although civil engineers were among the first of all traditional engineering disciplines to use the power of computers five decades ago, they are now lagging behind other professions in the use of advanced techniques such as data mining. Indeed, data mining techniques have proven their efficiency in domains such as handwritten digit recognition, image and speech recognition, DNA sequences, financial time series and web mining. Although data mining has been used in engineering, most of this work takes advantage of the predictive abilities of data mining methods. Very little work applies data mining techniques to tasks such as describing the structure of data. Known to the author, there is no attempt to apply data mining to models in system identification. This work is thus a new application for data mining.

Figure 1.1: Data mining can be compared to gold mining in rivers.

1.3 System Identification

Several years after construction, structures may no longer fulfill their intended functions. As written in Levy and Salvadori (2002), "*It is the destiny of the man-made environment to vanish [...]*". People outside of civil engineering domains have the misconception that civil engineers know exactly how structures behave in service. The complexity of both the structures and the materials involved make the understanding of exact structural behavior impossible. One way to learn about the state of the structure, before it collapses or as frequently happens, it reaches a stage where repair costs increase by orders of magnitude, is through diagnosis. When the goal of diagnosis is to determine models that reasonably explain measured responses, the approach is commonly known as system identification. Although system identification is closely related to diagnosis, the focus of this work is on helping engineers identify the system, not diagnose it. The aim is not to propose a way to repair the system as it is the case in diagnosis, rather to find the state of the system (even if it is not damaged) in order to improve management of artifacts that are expected to last more than one hundred years.

The goal of system identification is to determine the state of a system and values of system parameters through comparisons of predicted with observed responses. Traditionally, this is treated as an optimization problem in which the best combination of values of model parameters are selected such that differences between model predictions and measurements are minimal. Recent work has brought out the different types of errors that can occur in system identification processes (Robert-Nicoud, 2003). These errors make optimization in system identification unreliable since the global optimum may not correspond to the true state of the system due to compensating modeling and measurement errors. In such situations, treating the task as a constraint satisfaction problem (CSP) is more appropriate (see Section 2.5). It is noted that recent work proposes a distributed version of the constraint programming approach (Faltings, 2006).

Since measurements are indirect, the use of models is necessary. Even though a design model may be the most appropriate for designing and analyzing the structure prior to construction, it often cannot be used for system identification. This is usually because design models are conservative. On the other hand, diagnosis models have to be as accurate as possible in order to avoid wrong diagnoses. The current work is a combination of model based reasoning concepts from computer science (de Kleer and Williams, 1987) and traditional model updating techniques used in engineering (Ljung, 1999). A correct understanding of the output using such techniques is an important challenge.

Difficulties associated with system identification are that since many model predictions might match observations with certain limits, the best matching model may not be the correct model. In this work, the reliability of identification is defined as the probability that the candidate model(s) obtained through system identification corresponds to reality. Reliability is poor when many models predict the similar responses at measured locations. Factors that affect the reliability of system identification have been studied in previous research (Robert-Nicoud et al., 2004). The present work is an extension of this research and uses data mining techniques for a better estimation of the reliability of identification.

1.4 Sensor Placement

A basic assumption of system identification is that there is a set of sensors measuring an effect. There are thousands of ways to measure physical phenomena in structures and many new technologies are emerging. Although their development has been the result of significant scientific effort, decisions related to the choice of measurement technology, specifications of performance and positioning of measurement locations are often not based on systematic and rational methodologies. While use of engineering experience and judgment may often result in measurement systems that provide useful results, a poorly designed measurement system can waste time and money.

When placing sensors on a structure, the analogy with medical diagnosis is relevant. People usually go to the doctor for a diagnosis of their conditions. They want to know what is wrong. For that, the doctor measures physiological parameters such as temperature and pulse rate. They try to infer causes from what is measured. The way doctors conduct the measurements is iterative. Through asking patients questions, they formulate possibilities. From these possibilities they measure temperature, for example. They iteratively measure symptoms and improve upon possibilities so that they are most likely to match real causes of problems. This procedure is very similar to sensor placement. Engineers place sensors, record measurements, formulate possibilities and then place more sensors if needed. This is done iteratively until they think they have enough information to identify the state of the system. However, the current lack of a systematic methodology for placing sensors on structures means that sensor placement tasks are usually very subjective.

1.5 Objectives

The primary goal of this work is to investigate the application of recent advances in data mining to system identification. To fulfill this objective, a methodology that uses data mining methods to support complex diagnostic tasks in engineering is proposed. Data mining techniques are used for better understanding of candidate models. The applicability of data mining techniques is further investigated for supporting iterative sensor placement in system identification.

This work is summarized with the following research questions:

To what extent can data mining techniques support engineers during system identification tasks? How can these techniques be improved to enhance the reliability of system identification?

These research goals are translated into the following objectives:

1. **Provide support for system identification through data mining**
   Data mining techniques are used to identify characteristics of candidate models that explain observations. Aspects involved are to i) estimate the number of groups among candidate models, ii) group them into clusters and iii) propose a methodology to select relevant parameters that explain candidate models.

2. **Design and implement a decision support system**
   Information obtained from objective 1 is used to integrate data mining into an overall methodology for system identification. Research issues include i) developing a clustering strategy to estimate possible states of a structure, ii) proposing an efficient way to display multi-dimensional clusters and iii) integrating input from engineers within the iterative methodology.

3. **Improve measurement system design using data mining techniques**
   Comparison of strategies for initial sensor placement are carried out. In addition to providing information for the engineer, data mining methods are used for iterative configuration of measurement systems. Tasks involved in achieving this objective are to i) employ a global search algorithm for initial sensor placement, ii) compare its performance to an existing greedy strategy, iii) support additional sensor placement through data mining and iv) propose an efficient stopping criterion for the methodology.

4. **Validate the methodology on a real engineering example**
   To demonstrate its capabilities on a real structure, the methodology is applied to the Schwandbach Bridge, a famous bridge in Switzerland with historical importance.

Figure 1.2 shows the research methodology used to achieve these objectives. It can be used as a guide for the reader. Research activities, validation data and parameters are described throughout the thesis. A final discussion on this schema is given in Section 7.2.

The structure of this thesis is as follows. The next chapter reviews the relevant literature and highlights supporting research as well as areas needing further work. Chapter 3 presents a new cluster validity index. Chapter 4 introduces a new feature selection algorithm. The system identification methodology integrating data mining techniques used in this work is presented in Chapter 5. Sensor placement strategies are presented in Chapter 6. Finally, discussion of conclusions and future challenges are described in the last chapter.

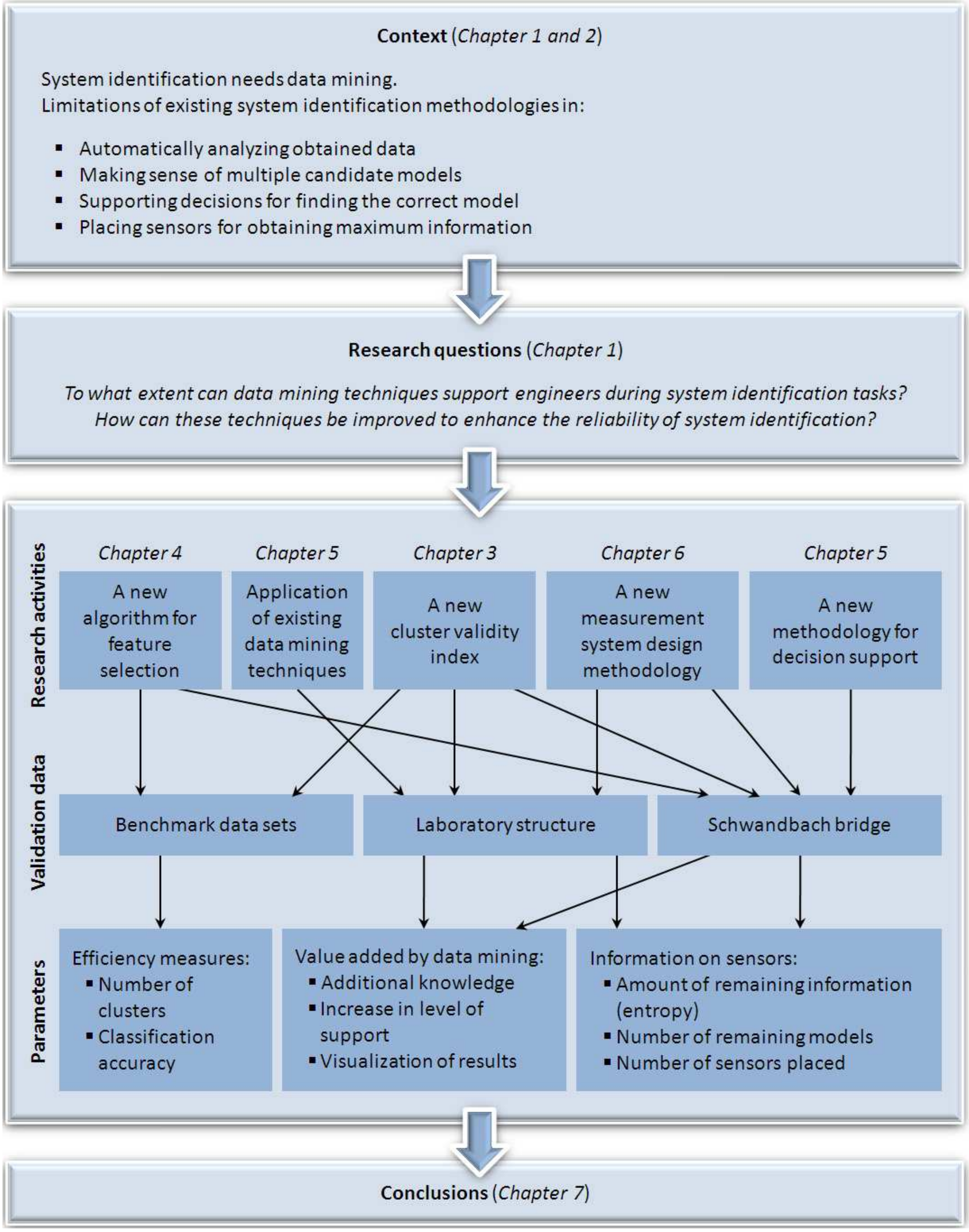

Context (Chapter 1 and 2)
System identification needs data mining.
Limitations of existing system identification methodologies in:
Automatically analyzing obtained data
Making sense of multiple candidate models
Supporting decisions for finding the correct model
Placing sensors for obtaining maximum information
Research questions (Chapter 1)
To what extent can data mining techniques support engineers during system identification tasks?
How can these techniques be improved to enhance the reliability of system identification?
Research activities
Chapter 4
A new algorithm for feature selection
Chapter 5
Application of existing data mining techniques
Chapter 3
A new cluster validity index
Chapter 6
A new measurement system design methodology
Chapter 5
A new methodology for decision support
Validation data
Benchmark data sets
Laboratory structure
Schwandbach bridge
Parameters
Efficiency measures:
Number of clusters
Classification accuracy
Value added by data mining:
Additional knowledge
Increase in level of support
Visualization of results
Information on sensors:
Amount of remaining information (entropy)
Number of remaining models
Number of sensors placed
Conclusions (Chapter 7)

Figure 1.2: Research methodology for this thesis.

Chapter

2

Literature Review

“*Computers have promised us a fountain of wisdom but delivered a flood of data.*” (Piatetsky-Shapiro, 1991)

Overview

In this chapter, the literature is reviewed to identify strengths and weaknesses in existing research. Strengths provide foundations for this thesis. Weaknesses, including lack of work, help establish the originality of the research objectives of this thesis. Topics that are studied include data mining, clustering, feature selection, system identification and sensor placement. Since data mining is the main area of research of this thesis, definitions of important terms are provided.

2.1 From Data to Knowledge

As written in Dietterich (2003), “*Machine learning is the study of methods for programming computers to learn.*” Machine learning (Langley, 1996; Flach, 2001) is an active artificial intelligence domain. In a similar way as statistics, the objective of machine learning is to find relationships in data. The term learning is difficult to define precisely (Figure 2.1). Many researchers have assigned different meanings to the term. One definition comes from Witten and Frank (2005): “*To get knowledge of by study, experience, or being taught*”. The term learning can be defined as in Ackoff (1989): “*Learning takes place when one’s efficiency increases over time or trial*”. There are several types of machine learning algorithms (Mitchell, 1997). Their usefulness depends on characteristics of the application.

Machine learning is a well established field since it has been successfully applied within a range of domains such as market analysis (Ari, 2004), classifying DNA sequences (Simek et al.,

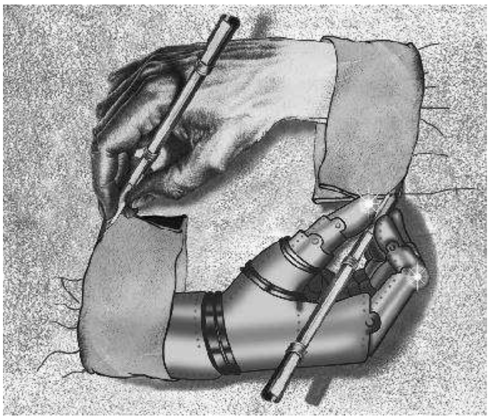


Figure 2.1: The concept of machine learning. The human teaches the machine (i.e. the computer) how to learn (M.C. Escher, Drawing Hands, 1948).

2004), handwriting recognition (Bahlmann et al., 2002), image recognition (Wilking and Roefer, 2004) and text categorization (Sebastiani, 2002). For a longer list of successful applications, see Cristianini and Shawe-Taylor (2000). Flach (2001) proposes a personal review of important books in machine learning. A comprehensive current state of the machine learning field is given by Mitchell (2006).

The more data there are, the more difficult it is to analyze them and obtain useful knowledge (Kantardzic and Zurada, 2005). As noted in Lavrac et al. (2004), data mining is different from machine learning in terms of the main objectives of the whole data analysis process. The greatest difference is that machine learning is more concerned with algorithms that find patterns, whereas data mining focuses on the knowledge extraction process. As written in Dietterich (2003), while machine learning is concerned with accuracy and effectiveness, data mining seeks to find understandable patterns. The term *data mining* refers to mining knowledge from large amounts of data (Han and Kamber, 2001). It is situated at the intersection of statistics, machine learning and databases (see Figure 2.2). The general aim of data mining is to apply machine learning algorithms to usually large data sets. In the real world, where data mining results are used, data are often incomplete, noisy and larger than machine learning data sets (Frawley et al., 1992).

According to Pal and Jain (2005), the origins of data mining date back to 1989, when the IJCAI[1] Workshop on knowledge discovery in databases (KDD) (Piatetsky-Shapiro, 1991) took place. In statistics, the traditional paradigm is to first establish hypotheses on the data and then test them. When data sets contain hundreds of attributes and more, this process

[1] International Joint Conference on Artificial Intelligence.

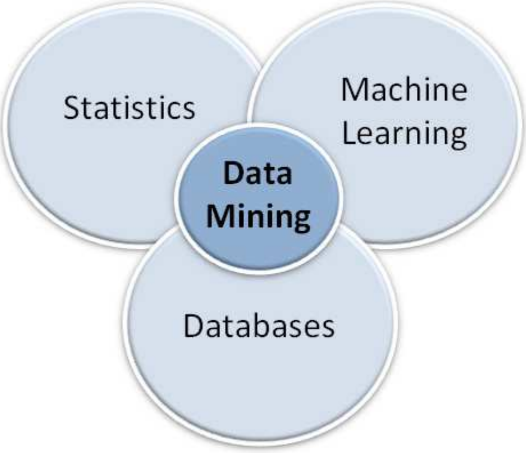


Figure 2.2: Data mining is at the intersection of statistics, machine learning and databases. Picture adapted from Tan et al. (2006).

becomes difficult and time consuming. With the increasing ability of computers to store and process data in a reasonable time, approaches have evolved to test-and-hypothesize (Carbone, 2000). Data mining has emerged both from the machine learning and database communities in the 1990's. The original KDD conferences initiated many early data mining ideas. However, the use of the term *data mining*, is much older. In medicine, Harris (1984) uses sequential multiple logistic regression to determine the risks associated with angiocardiography. In this work, the term data mining is used as a synonym for data exploration. Lovell (1983) wrote a paper entitled "Data Mining" where these words are defined as a "*a research paradigm that masquerade under a variety of aliases*". In econometrics[2], the term data mining has a somewhat negative connotation (Smyth, 2000). The earliest paper found by the author that mentions data mining is by Jorgenson et al. (1970). Data mining is defined as the process of "*consideration of a wide range of alternatives and selection of the one that fits best*". It is only in the early 1990's that the term data mining was adopted by computer scientists with its current meaning. A brief history of data mining is given in Smyth (2000).

Data mining methods can be grouped in three categories: supervised learning, reinforcement learning and unsupervised learning. Supervised learning can be seen as learning with a teacher that gives feedback for the learning task. This feedback is represented by a training set and consists of examples with both input and output values. It is opposed to the test set, which is the final set one want to test and that consists only of input values (the output is predicted).

[2] The field of developing and applying statistical methods in economics.

Patterns in data can be automatically identified, validated on existing data and then used for predictions with new data (Witten and Frank, 2005). In reinforcement learning, training data contain partial information about the output. In unsupervised learning, no feedback is given to the learning algorithm (i.e. no teacher). Particularities of this category are that trends are directly inferred from the data set, thus no output is known for a given data set.

Several recent textbooks (Hand et al., 2001; Webb, 2002; Tan et al., 2006) cover the data mining research area. Data mining is usually applied to tasks such as recognition of images (Wilking and Roefer, 2004), characters (Sempere and Lopez, 2003) and speech (Zhou et al., 2005). Data mining has also been successfully applied in domains such as crime pattern detection (Nath, 2006), gene classification (Yuan et al., 2003), email classification (Aery and Chakravarthy, 2005) and collaborative filtering (Candillier et al., 2007). Several other applications of data mining are briefly described in Langley and Simon (1995). Examples of application domains are given in Frawley et al. (1992). Limitations are also under study, for example in the domain of counterterrorim (Jonas and Haper, 2006). Valuable introductory texts about data mining are Fayyad and Uthurusamy (2002), Witten and Frank (2005) and Tan et al. (2006). A framework for fast application of data mining has been proposed by Menzies and Hu (2003). Known to the author, data mining has never been applied as a knowledge extraction process for the task of system identification.

Data mining is a very active research domain. This is evident from the recent creation of the ACM[3] Transactions on knowledge discovery from data (2007) and Statistical Analysis and Data Mining (2008) even though several journals already exist in this field. Kriegel et al. (2007) give a view of the future of data mining by presenting incoming challenges in this field. Nowadays, data mining research is mostly influenced by practical problems coming from industry sectors (Perner, 2006). As written in Langley and Simon (1995), "*The ultimate test of machine learning is its ability to produce systems that are used regularly in industry [...]*". This work is an example of successful application of data mining methods in engineering (objective 4, Section 1.5).

2.2 Introduction to Existing Data Mining Techniques

2.2.1 Pearson's Correlation

Visualization in a multi-dimensional space is not feasible for full-scale engineering tasks, when the number of parameters p is more than 3. The simplest visualization strategy is to plot a graph for each possible combination of parameters. For each parameter pair (p_i, p_j) there is a two dimensional plot. Drawbacks are that this requires human intervention and that the

[3] Association for computing machinery.

number of graphs increases with the square of the number of parameters p, thereby resulting in a complexity of $O(p^2)$. A numerical method to search for these relationships is to use the correlation measurement (Edwards, 1984). Correlation is a measure of linear association between two random variables. It is derived from the covariance measure and is given by:

$$\mathrm{corr}(x, y) = \frac{\mathrm{cov}(x, y)}{\sqrt{\mathrm{var}(x) \cdot \mathrm{var}(y)}} \tag{2.1}$$

where *cov* is the covariance and *var* the variances of the specified variables. The correlation between two variables x and y corresponds to the link between them and can be written:

$$\mathrm{corr}(x, y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\left[\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2\right]^{1/2}} \tag{2.2}$$

where n is the number of samples for which variables x and y are compared. The correlation varies between -1 and 1. These bounds are reached when the association between x and y is perfectly linear. If the correlation is zero, it means that the covariance is zero (Equation 2.1). If this is the case, the two variables are assumed to be independent. Results of the application of correlation to system identification are given in Section 5.3.4.

2.2.2 Principal Component Analysis (PCA)

As explained in Section 2.2.1, displaying multidimensional spaces is not straightforward. One strategy to improve feasibility is to reduce the dimensionality of the space. One of the most popular methods for reducing dimensionality is principal components analysis (PCA) (Smith, 2002; Davies and Fearn, 2005). PCA generates a new set of variables - called principal components - that are linear combinations of the initial variables. The goal of PCA is to find a system of principal components that are sorted in a manner that the first components can explain most of the data.

In order for the data to be statistically comparable, they are first standardized to have a zero mean and unit standard deviation. Otherwise it is unreliable since PCA is concerned with explaining relative variations in parameter values. For each variable to have zero mean and unit standard deviation, data are transformed according to the expression $(x_i - \mu_i)/\sigma_i$ where μ_i is the mean and σ_i the standard deviation of x_i.

The starting point for using PCA is the correlation matrix of the data. As data are already standardized, the covariance matrix is used. In this section, the term parameter refers to the parameter in it's standardized form. To obtain the principal components, the covariance matrix S is first constructed as follows:

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1p} \\ s_{21} & s_{22} & \dots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{p1} & s_{p2} & \dots & s_{pp} \end{bmatrix} \tag{2.3}$$

where s_{ij} is the covariance between the parameter p_i and p_j. The formula of the covariance corresponds to the numerator of equation (2.2). Note that the special cases s_{kk} are equal to the variance of k. The PCA method is based on the fact that the covariance matrix S can be written as:

$$\mathbf{S} = \mathbf{VLV^T} \tag{2.4}$$

where L is a diagonal matrix containing the eigenvalues of S and the column of V contains the eigenvectors of S (for more details see Jackson (1991)). The principal components, which are linear combination of the initial variables, correspond to the eigenvectors of S and can be represented as an orthogonal basis for the new space of the data. The principal components are sorted in decreasing order according to how well they represent the variability of the data. Each sample is transformed to a new dimensional space defined by selected principal components. The main practical goal is usually to reduce the number of dimensions by choosing only the first two or three principal components. Thus, the initial data are represented by a linear combination of the initial parameters in a new and lower dimensional space. Results of the application of PCA to system identification are given in Section 5.3.5.

2.3 Clustering and Cluster Validity

2.3.1 Clustering Techniques

One of the best known examples of unsupervised learning is clustering (Jain and Dubes, 1988; Xu and Wunsch, 2005; Tan et al., 2006). The goal of clustering is to group data points that are similar according to a chosen similarity metric (Euclidean distance is commonly used). Clustering can also be used in combination with other techniques such as genetic algorithms (Korkmaz et al., 2006). Clustering techniques have been applied in domains such as text mining (SanJuan and Ibekwe-SanJuan, 2006), intrusion detection (Liu et al., 2004), DNA micro-arrays (Garatti et al., 2007) and information exploration (Hearst, 2006). In these fields, as in many others, the number of clusters is usually not known in advance.

Clustering techniques that are proposed in the literature, although considerable (Jain et al., 2004), can be divided into four main categories (Halkidi et al., 2001): partitional clustering (for example, K-means), hierarchical clustering (for example, BIRCH), density-based clustering (for example, DBSCAN) and grid-based clustering (for example, STING). Although the mixture of Gaussian approach can be mentioned, its computational complexity is too high to be used in practice and approximation procedures are often needed (Cheung, 2005). Clustering is known as a form of unsupervised learning, as well as numerical taxonomy and partitioning (Theodoridis and Koutroumbas, 1999).

One of the most popular techniques for clustering is K-means (McQueen, 1967; Jain and Dubes, 1988). Reasons for the popularity of this technique include the absence of drawbacks of other types (Halkidi et al., 2001). For example, hierarchical clustering has a higher complexity and density-based clustering algorithms often require tuning non-intuitive parameters. Finally, density-based clustering algorithms do not always give clusters of good quality. Advantages of K-means include computational efficiency and easy interpretation of results. K-means is certainly the most widely used clustering algorithm in practice (Berkhin, 2002). K-means clustering has been successfully applied in domains such as relational databases (Ordonez, 2006), gene expression data (Chan et al., 2006) and decision support (Packhama et al., 2005).

The drawbacks of K-means include random choice of centroid locations at the start of the algorithm, treatment of variables as numbers and the unknown number of clusters k. Impact of the first drawback can be assessed through multiple runs or specific initialization methods (Bradley and Fayyad, 1998). However, Pena et al. (1999) have reported that specific initialization methods are not better than random centroids. The paper by Huang (1998) contains a possible solution to the second drawback through the use of a matching dissimilarity measure to handle categorical parameters. Concerning the third point, the number of clusters is an input parameter that is fixed *a priori* in the standard K-means algorithm. One way to address this challenge is through the use of cluster validity indices. As many other data mining algorithms, K-means has reduced reliability when treating high-dimensional data because data sets are nearly always too sparse. This is because the use of the Euclidean distance becomes meaningless in high-dimensional sparse spaces (François, 2007). A solution involves combining K-means with feature extraction methods such as principal component analysis (PCA) (Ding and He, 2004) and self-organizing maps (SOM) (Vesanto and Alhoniemi, 2000).

For the purpose of grouping data, the K-means algorithm is used with adaptations (see below). K-means evolves k crisp and hyper-spheroidal clusters in order to minimize their intra-cluster distances, shown as the metric J in Equation 2.5:

$$J = \sum_{j=1}^{k} \sum_{x_i \in c_j} d(x_i, z_j)^2 \tag{2.5}$$

where k is the number of clusters, x_i the i-th data point and z_j the centroid of cluster c_j. The k starting centroids are chosen randomly among all data points. The data set is then partitioned according to the minimum squared distance. The cluster centers are iteratively updated by computing the mean of the points belonging to the clusters. The process of partitioning and updating is repeated until a stopping criterion is reached. This happens when either the cluster centers or the value of the metric J in Equation 2.5 do not change over two consecutive iterations.

As stated in above, K-means has three main drawbacks. Two of them are taken into account. First, to control the randomness of K-means, it is launched $t = 20$ times from k_{min} to k_{max} clusters. A new validity index is used to estimate the number of clusters, which is an input to K-means. Thus, at each iteration, an index value is computed. The minimum or maximum - depending on the index - is chosen to be the most suitable number of clusters.

2.3.2 Cluster Validity

When performing clustering tasks, results should be treated with caution. Indeed, as noted in Jain et al. (1999), clustering is a difficult subjective task. An impossibility theorem for clustering has even been proposed. In Kleinberg (2002) it is shown that there is no clustering function that satisfies a set of three properties (scale-invariance, richness and consistency). This theorem can be relaxed for practical use of clustering algorithms. As written in Maulik and Bandyopadhyay (2002), two important issues in clustering are i) determination of the number of clusters present in the data and ii) evaluating how good is the clustering itself. These two issues motivate research in the field of cluster validation. Validity indices are also useful for estimating the quality of clusters. An example is given in Famili et al. (2004). Other important challenges in clustering are fixing initial conditions (Salem and Nandi, 2005) and treating high dimensional data sets (Li et al., 2003). Many cluster validation techniques are available (Bezdek and Pal, 1998; Fraley and Raftery, 1998; Halkidi et al., 2001, 2002a,b). This evaluation can be used to determine the most reliable number of clusters in a data set. Several indices have been proposed in the literature (Bezdek and Pal, 1998; Halkidi et al., 2001; Wu and Chow, 2004; Kim and Ramakrishna, 2005; Yang et al., 2006). These indices were evaluated through plotting them to determine the number of clusters visually. Most of them have been compared with known results (Kim and Ramakrishna, 2005; Brun et al., 2007). Selected validity indices are briefly described below.

The Hubert statistic assesses how well the data fit a proposed crisp structure. The concept behind the Hubert statistic is the correlation measure. Since calculation of the original index is computationally expensive, a modified index was proposed. In the modified Hubert statistic (Theodoridis and Koutroumbas, 1999), a *knee* on the plot indicates a possible value for the number of clusters. Finding this knee is somewhat subjective. The Dunn index (Dunn, 1974) combines dissimilarity between clusters and their diameters to estimate the most reliable number of clusters. The Dunn index is computationally expensive ($O(n^2)$) and sensitive to noise (Halkidi et al., 2001). An index based on a ratio of between and within scatter cluster matrices is proposed by Calinski and Harabasz (1974). It is ranked among the best in Milligan and Cooper (1985). The concepts of dispersion of a cluster and dissimilarity between clusters are used to compute the Davies-Bouldin index (Davies and Bouldin, 1979) which has recently been reported to be among the best (Kim and Ramakrishna, 2005). The Silhouette index (Kaufman and Rousseeuw, 1990) uses average dissimilarity between points to show the structure of the data and consequently, its possible clusters. As stated in Bolshakova and Azuaje (2003), the Silhouette index is most suitable for estimating the first choice or the best partition.

The index proposed by Halkidi et al. (2000) is based on average scattering for clusters and total separation between clusters. This index has to be tuned with a parameter that may vary the clustering results for small number of clusters. The Maulik-Bandyopadhyay index (Maulik and Bandyopadhyay, 2002) is related to the Dunn index and involves the tuning of a parameter. Finally, the Geometric index (Lam and Yan, 2005) has been developed for handling clusters of different densities and close clusters as well. A particular feature of this index is the use of the eigen-axes lengths as a way of measuring the intra-cluster distance.

Much work has been done on stability-based methods for assessing cluster validity, where the main idea is to cluster subsample of the original data set (Ben-Hur et al., 2002; Lange et al., 2004; Greene and Cunningham, 2006). A relational visual cluster validity method is given in Ding and Harrison (2007). However, these methods require a visual inspection of the results or are computationally expensive or both. In Tibshirani and Walther (2005) the clustering problem is viewed as a supervised one. They use the self-defined prediction strength to estimate the number of clusters in a data set. Their method is highly complex and involves both hierarchical and K-means clustering.

Most of these indices require the specification of at least two clusters. Although not often studied by the data mining community, the single cluster case is important and is likely to happen in practice. Most cluster validity methods have the drawback that they are undefined for a single cluster Gordon (1996). Several other validity indices exist in the literature (Kothari and Pitts, 1999; Pelleg and Moore, 2000; Lam and Yan, 2005). Some are computationally expensive (i.e. higher than $O(n)$) (Halkidi et al., 2001) while others are unable to discover the real number of clusters in all data sets (Kim and Ramakrishna, 2005). In Kyrgyzov et al. (2007), kernel minimum description length (KMDL) is used for cluster validity. However, their method requires the tuning of an hyper-parameter. Although probability density functions (PDF) are used in Sakai et al. (2007), their technique relies on a visual inspection of results. The gap statistic (Tibshirani et al., 2000) compares the change of within cluster dispersion to an appropriate null distribution. Although this index is defined for $k = 1$, the paper only focuses on well separated clusters. The new validity index proposed in this work helps overcome such limitations (objective 1, Section 1.5).

2.4 Feature Selection

2.4.1 Wrapper Approach

Feature selection (Liu and Motoda, 1998; Cakmakov and Bennani, 2002; Guyon et al., 2006) is a method used to reduce the number of features before applying a data mining algorithm. Irrelevant features may have negative effects on a prediction task. Moreover, the computational complexity of a classification algorithm may suffer from the *curse of dimensionality* caused by several features. When a data set has too many irrelevant variables and only a few examples, overfitting is likely to occur (François, 2007). In addition, data are usually better characterized using fewer variables (Cheng et al., 2007). As written in Cakmakov and Bennani (2002), "*Identifying the common "core" characteristics of a set of objects that are representative of their class is of enormous use in focusing the attention of a person or computer program*". However, in practice several features are used to describe an effect. As written in Ackoff (1989), "*the less a phenomenon is understood, the more variables are required to explain it*". Feature selection has been applied in fields such as multimedia database search (Evgeniou et al., 2003), image classification (Fleuret, 2004) and biometric recognition (Kumar and Zhang, 2005). A comprehensive introduction to feature selection can be found in Guyon and Elisseeff (2003). It is noted that feature selection can also be unsupervised (He et al., 2005; Guérif and Bennani, 2007).

Feature selection techniques (Dash and Liu, 1997) can be divided into three main categories (Tan et al., 2006): embedded approaches (feature selection is part of the classification algorithm, i.e. decision tree), filter approaches (features are selected before the classification algorithm is used) and wrapper approaches (the classification algorithm is used as a black box to find the best subset of attributes). Due to its very definition, embedded approaches are limited since they only suit a particular classification algorithm. As noted in Molina et al. (2002), a relevant feature is not necessarily relevant for a given classification algorithm. Filter methods, however, make the assumption that the feature selection process is independent of the classification step. The work done by Kohavi and Sommerfield (1995) recommends replacement of the filter approach by wrappers. This usually provides better results, the price being higher computational complexity (Weston et al., 2001). Although already known in statistics and pattern recognition (Blum and

Langley, 1997), wrappers are new in the data mining community.

Since wrapper techniques treat the classification algorithm as a black box, any search strategy can be used in combination (Kohavi and John, 1998). This makes wrapper approaches universal. The accuracy of the classification algorithm may be used as the objective function of the search strategy. As with any classification algorithm, wrapper feature selection techniques face the overfitting problem that may happen while training. One way to reduce the overfitting problem is to use a k-fold cross-validation strategy (Stone, 1974; Bradley and Fayyad, 1998; Hsu et al., 2003). The training set is randomly divided into k different folds. Each fold is held out in turn while the $k-1$ remaining are used to train the classification algorithm. The classification error rate is calculated on the validation fold. The classification algorithm is thus executed k times on different training sets. Finally, the average of these k error rates is an approximation of the generalization ability of the classification algorithm. Correlation coefficient and mutual information are cheaper to evaluate than cross-validation. However, when the number of features is high, they become difficult to evaluate (François, 2007). As mentioned by Kohavi and John (1998), a wrapper approach such as the one used here need: i) a state space, ii) an initial state, iii) a termination condition and iv) a search engine (see Section 4.3 for details).

Several search algorithms that select a subset of m features out of d exist in the literature (Jain et al., 2000; Reunanen, 2006). Although exhaustive search is guaranteed to find the optimal feature subset, it is not feasible when d is not small. The branch and bound procedure proposed by Narendra and Fukunaga (1977) explores only a part of all possible feature combinations. It is guaranteed to find the optimal feature subset in less time than needed for exhaustive search. Its main drawback concerns the monotony assumption of the feature selection objective function. That is, if a variable is added to the feature set, it should never decrease the value of the objective function, which is not always the case (Zongker and Jain, 1996).

Individual ranking procedures are often called naive methods. The idea is to individually rank each feature at a time, according to its prediction power. This technique is valid only if every feature is independent, which is usually not the case in practice. Sequential forward selection (SFS) starts with a single feature and iteratively adds a feature to increase the classification criteria (Jain et al., 2000). Caruana and Freitag (1994) examine five hill-climbing procedures for feature selection. The main limitation of all these methods is that they are greedy strategies. Ideally, all possible subset of features should be considered. Combination of features can provide relevant information that is not carried separately by these features. The best example is the simple XOR problem, where two features taken separately may be irrelevant, while taken together are very useful. In this case, a sequential search technique will never find this relevant combination since none of the two variables are selected separately.

Sequential backward selection (SBS) starts with all features and iteratively remove a single feature to increase the classification accuracy (Jain et al., 2000). Although combination of features are taken into account with this technique, a high number of computations are necessary since it starts with the set of all features. This may not be feasible for very high dimensional data set. The *plus l take away r* method combines the former two techniques by first applying SFS for l features and then SBS for r features. Although this method seems promising, a value for l and r have to be given by the user. Sequential forward floating search (SFFS) and sequential backward floating search (SBFS) are generalization of the two former methods (Pudil et al., 1994). In this case, l and r are determined automatically. However, these techniques are still suboptimal procedures (Webb, 2002). Bins and Draper (2001) propose a feature selection algorithm for data sets with very large number of features. However, their algorithm is complex since it involves three different steps. Guérif and Bennani (2006) and Li et al. (2008) propose to combine feature selection and clustering. Finally, Tenenhaus et al. (2007) is a good example of a wrapper approach for feature selection.

2.4.2 Stochastic Methods

As written in Cakmakov and Bennani (2002), there are mainly two reasons to use stochastic methods for feature selection. First, to avoid getting stuck in local minima and second to have more chance to capture feature dependences. Using stochastic methods is intuitive since the feature selection problem is exponential (Oh et al., 2004). An advantage of stochastic methods is the avoidance of the monotonicity assumption made by sequential methods (Yang and Honavar, 1998). In Loughrey and Cunningham (2005), a search strategy using simulated annealing (SA) is used for feature selection. Lin et al. (2006) propose to combine SA with support vector machine (SVM) for feature selection and hyper-parameter optimization. Several studies have also been carried out using genetic algorithms (GA) for feature selection (Vafaie and Imam, 1994; Yang and Honavar, 1998; Raymer et al., 2000; Huang and Liu, 2006). Hybrid GA procedures have been proposed as well (Oh et al., 2004; Huang et al., 2007). Both SA and GA have a wide range of hyper-parameters to tune before obtaining convincing results (Liu and Chian, 1997; Kudo and Sklansky, 2000; Oh et al., 2004). For SA, they are annealing schedule, number of loops, initial temperature and transition rate. For GA, they are population size, crossover rate, mutation rate and number of generations. If the tuning is not done correctly, this leads to poor results.

PGSL is a direct search algorithm that employs global sampling to find the minimum of a user defined objective function. PGSL has been successfully applied to optimization problems involving highly non-linear objective functions containing a large number of local minima (Raphael and Smith, 2003a). It has proven its efficiency for structural control (Domer et al., 2003), system identification (Robert-Nicoud et al., 2005a), configuration (Svanerudh et al., 2002) and leak detection (Raphael and Smith, 2005). PGSL has advantages over SA and GA regarding hyper-parameter tuning (Domer et al., 2003). Only three parameters are fixed using a simple guideline proposed in Raphael and Smith (2003a). Moreover, PGSL gives competitive results when comparing to SA and GA (Raphael and Smith, 2003a; Domer et al., 2003; Raphael and Smith, 2005). However, no work has been done on using PGSL for feature selection (objective 1, Section 1.5).

As explained above, wrapper feature selection combines a search strategy with a classification algorithm. Among classification techniques, kernel methods (Boser et al., 1992; Cristianini and Shawe-Taylor, 2000) have potential because they can detect more general relationships. Kernel methods map data into a (generally) higher dimensional vector space in order to detect structure in the data more easily. Kernel methods are generally well known thanks to the increasing popularity of support vector machines (SVM) (Boser et al., 1992; Vapnik, 1995). SVM is a kernel-based technique that can be used with such inner product information. SVM have been successfully applied in domains such as text classification (Zhuang et al., 2005), face identification (Fernandez and Viennet, 1999) and classification of gene expression data (Yuan et al., 2003). SVM hyper-parameters can be found through grid search (Soares et al., 2004). Chapelle et al. (2002) propose a gradient descent algorithm. In Luxburg et al. (2004), data compression is used on the training labels for hyper-parameter selection. Although a hybrid Monte Carlo technique is proposed in Gold and Sollich (2002), it is computationally expensive. As proposed in Fröhlich et al. (2003) for GA, in the approach suggested in this thesis, selection of SVM hyper-parameters is done throughout the feature selection process using PGSL. Finally, it has been observed that SVM can suffer from irrelevant features (Barzilay and Brailovsky, 1999; Weston et al., 2001; Rakotomamonjy, 2003).

As noted in François (2007), classification techniques such as ANN and SVM using a Gaussian kernel consider each feature to have equal importance. For this reason, SVM may perform badly when there are many irrelevant features (Weston et al., 2001). The literature contains several applications of SVM to feature selection as well as feature selection for SVM (Hermes and Buhmann, 2000). In Evgeniou et al. (2003), a variable is important if, when removed, the separating boundary varies the most. This method is however applicable only when features are independent. Guyon et al. (2002a,b) propose an algorithm based on SVM and recursive feature elimination (RFE). Liu and Zheng (2006) use SVM for wrapper feature selection. Both strategies use a greedy algorithm, iteratively adding/removing features. In Rakotomamonjy (2003), although SVM is used for feature selection, the feature selection and learning process are distinct. A sparse linear SVM algorithm that inherently performs variable selection is used in Bi et al. (2003). Chen and Lin (2006) and Wu and Li (2006) also use SVM for feature selection. These works either combines SVM with a filter approach or use particularities of SVM, avoiding its applicability with other classification algorithms. No work has been done on combining PGSL and SVM for wrapper feature selection (objective 1, Section 1.5).

2.5 System Identification

2.5.1 Introduction

A systematic approach to interpretation of measurement data employs methodologies developed in the field of system identification (Ljung, 1999). System identification involves determining the state of a system and values of system parameters through comparisons of predicted and observed responses. When the forms of relationships between observable quantities and system parameters are known, regression techniques are useful for identifying system parameters. However, these techniques are rarely applicable to structural engineering because closed form relationships between system parameters and responses are often not available. Most structures are analyzed using numerical methods. Strategies that compute the values of finite element model parameters through matching predicted responses with measured values are called finite element model updating or model calibration methods. Fully instantiated models that have (continuous) values for all parameters are obtained through these procedures.

A survey of model updating procedures is given in Robert-Nicoud et al. (2005c). Many previous studies propose methods for modifying stiffness coefficients that predict dynamic properties of structures, for example Friswell and Mottershead (1995). Such work is often aimed at supporting evaluation of earthquake damaged structures (Chaudhary et al., 2000). Proposals for interpretation of static measurements are few and they involved minimizing the difference between measured and analytical quantities from a given finite element model (Liu and Chian, 1997; Reich and Park, 2001). The number of unknown variables is fixed. Models that have varying numbers of degrees of freedom and consequently, different sets of variables are not accommodated in such approaches. Note that model free monitoring exist (Posenato et al., 2006; Lubasch et al., 2006). In the latter, however, the study is limited to load identification.

In conventional system identification, a suitable model is identified by matching measurement data with model predictions. Model calibration involves minimization of the difference between predictions and measurement data through identification of good values of model parameters. This strategy is based on the assumption that the model that best fits the observations is the most reliable model. This assumption is flawed; there are several factors that could cause the best fit to be the wrong model.

Errors influence the reliability of system identification. Various types of errors may compensate each other such that bad model predictions match measured values. The following definitions are used in this description: measurement error (e_{meas}) is the difference between real and measured quantities in a single measurement. Modeling error (e_{mod}) is the difference between the prediction of a given model and that of the model that accurately represents the real behavior. Modeling errors have three principal sources e_1, e_2 and e_3 (Raphael and Smith, 2003b). Source e_1 is the error due to the discrepancy between the behavior of the mathematical model and that of the real structure. Source e_2 is introduced during the numerical computation of the solution of the partial differential equations representing the mathematical model. Source e_3 is the error due to the assumptions that are made during the simulation of the numerical model. Typical assumptions are related to the choice of boundary conditions and model parameters such as material properties, for example E and I. All these errors as well as the abductive aspect of the system identification task justify the use of a multiple model approach since many models may have equal validity under these conditions.

2.5.2 Multiple Models

Compositional modeling is a framework for constructing adequate device models by composing model fragments selected from a model fragment library (Falkenhainer and Forbus, 1991). Model fragments partially describe components and physical phenomena. A complete model is created by combining a set of fragments that are compatible. For modeling the behavior of structures, fragments represent support conditions, material properties, geometric properties, nodes, number of elements and loading. Assumptions are explicitly represented in model fragments so that the model composition module generates only valid models that are compatible with the assumptions chosen by users. Model composition makes it possible to search for models containing varying numbers of degrees of freedom. There is no need to formulate an optimization problem in which the number of variables is fixed a-priori. Models are automatically generated by combining model fragments and are analyzed by the finite element method in order to compare their predictions with measurements.

In model-based diagnosis (de Kleer and Williams, 1987), a library of models can be used to perform the diagnosis of a system (Struss, 2007). The core objective of model-based diagnosis is to find candidate diagnoses that explain observations (de Kleer, 2006). Whereas in system identification the aim is to find the state of the system (whether there is a fault or not), model-based diagnosis objective is to diagnose the system, i.e. find the problem (Balakrishnan and Honavar, 1998). However, diagnosis goes beyond the task of finding the problem. As written in Struss (2007), "*Diagnosis is only relevant if it supports a decision [...]*". Thus, the final aim of diagnosis is not only to identify the problem, but also to find a possible remedy. Examples of remedy are replacement of components, reconfiguration, etc. Examples of applications of model-based diagnosis are automotive industry (Struss and Price, 2004), autonomous mobile robots (Steinbauer and Wotawa, 2005) and software debugging (Köb and Wotawa, 2004).

A possible way to perform fault diagnosis is through parameter estimation (Isermann, 1993). Using static and dynamic process models as well as measurements, relationships and redundancies are used to detect faults. A methodology based on parameter estimation is used to generate analytical symptoms. Examples can be found in industrial robots (Freyermuth, 1991) and grinding machines (Janik and Fuchs, 1991). The notion of analytical redundancy due to measurements is used as well. For more details, see Section 7.4.6. In Krysander and Nyberg (2002), consistency relations with highest diagnosis capability are used. For this, they use the structural information to find sub-models which are then derived in consistency relations. Finally, Darwiche (2000) studies the application of model-based diagnosis under real-world constraints. System identification is a complex process and can, for example, be supported by qualitative reasoning-based approaches (Travé-Massuyès et al., 2003). However, in the latter, a single model is iteratively updated. The paper by Addanki et al. (1991) introduces the concept of graphs of models. However, in their work, models are generated manually and they work with only one model at a time.

Traditionally, system identification is treated as an optimization problem in which the difference between model predictions and measurements is minimized. Values of model parameters for which model responses best match measured data are determined by this approach. As explained in Section 2.5.1, this approach is not reliable because different types of modeling and measurement errors are present (Banan et al., 1994; Sanayei et al., 1997; Catbas et al., 2007). Moreover, they can compensate each other such that the global minimum indicates models that are far away from predictions of the model representing the correct state of the system (Robert-Nicoud et al., 2005c). Therefore, instead of optimizing one model, a set of candidate models is identified, such that their prediction errors lie below a certain threshold value.

A model is defined in Robert-Nicoud et al. (2005c) (and in this thesis) as a distinct set of values for a set of parameters. The threshold is computed using an estimate of the upper bound of errors due to modeling assumptions (e_{mod}) as well as measurements (e_{meas}). The set of candidate models is iteratively filtered using subsequent measurements for system identification. This approach could generate either a unique model for the structure or a set of models which are equally capable of representing the structure. This depends on parameters chosen for the identification problem and errors.

Modeling assumptions define the parameters for the identification problem. The set of model parameters may consist of quantities such as elastic modulus, connection stiffness and moment of inertia. Each set of values for the model parameters corresponds to a model of the structure. An objective function is used to evaluate the quality of candidate models. The objective function E is defined as follows:

$$E = \begin{cases} \varepsilon & \text{if } \varepsilon > \tau \\ 0 & \text{if } \varepsilon \leq \tau \end{cases} \quad \text{with} \quad \varepsilon = \sqrt{\sum (m_i - \gamma_i)^2} \tag{2.6}$$

ε is the error which is calculated as the difference between predictions γ_i and measurements m_i. τ is a threshold value evaluated from measurement and modeling errors in the identification process. The set of models that have $E = 0$ form the set of candidate models for the structure.

An important aspect of the methodology is the use of a stochastic global search and optimization algorithm for the selection of a population of candidate models whose predictions are close to measurements (Robert-Nicoud et al., 2000). Mathematical optimization techniques that make use of derivatives and sensitivity equations are not used because search is performed among sets of model classes that contain varying numbers of parameters and multiple local minima have been observed in the search space. A stochastic global search algorithm called PGSL (Raphael and Smith, 2003a) is used to minimize the cost function that evaluates the difference between measurements and model predictions. It has been empirically observed that the number of evaluations of the objective function required by PGSL for finding the global minimum does not increase as rapidly as other search techniques such as genetic algorithms (Raphael and Smith, 2003a).

It was shown earlier that the location of the global minimum shifts in the presence of modeling and measurement errors. Therefore, an accurate computation of the global minimum is not necessary and in fact, could result in solutions that are far from reality. Hence the search task is reformulated such that any solution whose objective function value lies below a threshold is considered to be acceptable. A population of solutions is obtained by repeating search several times. No work has been done on automatic analysis of these models. Data mining methods can be used to better understand these models (objective 2, Section 1.5).

2.5.3 Data Mining for System Identification

Data mining has already been used in engineering (Grossman et al., 2001; Melhem and Cheng, 2003; Alonso et al., 2004). Examples of applications include oil production prediction (Nguyen and Chan, 1999), joint damage assessment (Yun et al., 2001), traffic pattern recognition (Yan et al., 2005) and composite joint behavior (Shirazi Kia et al., 2005). Soibelman and Kim (2002) study data preparation, which is a crucial step before knowledge extraction. Machine learning methods such as decision trees and neural networks have also been integrated in diagnosis systems (Balakrishnan and Honavar, 1998). In Chantler et al. (1998), the use of machine learning techniques is mentioned and their high training set size requirement noted. In Portinale et al. (2004), case-based reasoning (CBR) is integrated in the context of diagnosis. However, these studies focus on the prediction abilities of such algorithms.

Some papers describing the use of machine learning applied to system identification have already been published. However, they concern dynamic systems (Abad et al., 2002), consistency based diagnosis (Alonso et al., 2004), automatic defect classification (McNamara et al., 2004) and automated repair (Saunders et al., 2000). In Cattan and Mohammadi (1997), neural networks are used to predict expert subjective ratings. The work by Chen et al. (2005a) focuses on the data quality aspect of civil infrastructure data. Soft computing techniques have been applied to infrastructure management (Flintsch and Chen, 2004). However, most of the soft computing techniques use neural networks and thus are not meant for knowledge extraction.

Data mining is an active field in manufacturing as well (Harding et al., 2006), for example in decision support. Koonce et al. (1997) use data mining, through decision trees, on industrial data. Kusiak (2002) generate rules for a manufacturing problem. Lee and Park (2003) employ data mining in the semiconductor manufacturing environment. Schnalzer et al. (2006) identify bridge performance patterns using hierarchical clustering. However, they did not discuss the number of clusters issue. No work has been done on applying data mining to models in system identification (objective 2, Section 1.5).

A clustering technique such as K-means (Webb, 2002) can be useful. Even though clustering is often proposed for various applications by the data mining community, it is not straightforward and there are many open research issues (see Section 2.3.2). While there are well accepted methods for evaluating predictive models such as cross-validation (Webb, 2002), clustering of possible models has not been investigated and quantitative methods are not straightforward for evaluating this task. The criterion for assessing the capability of algorithms is subjective and dependent on the final goal of the knowledge discovery task.

Pearson's correlation is a measure of the relationship between two variables (Edwards, 1984). Studies using correlation include natural hazards prevention (Pulinets et al., 2004) and cache replacement policies (Ari, 2004). To the knowledge of the author, correlation has never been applied to multiple model system identification. Principal component analysis (PCA) (Jolliffe, 2002) generates a small set of principal components (linear combinations of variables) that explain most of the variability of the data. It has been successfully applied to dimension reduction in financial time series (Lendasse et al., 2001) and micro-arrays data (Lexin and Hongzhe, 2004). Hybrid data mining methods are proposed in the literature (Pan et al., 2005; Xu et al., 2005). Most work combines data mining methods for better predictions. For example, Ding and He (2004) propose a combination of PCA and K-means to improve the prediction accuracy. However, the visualization improvement is not taken into account. Self-organizing maps (SOM) (Guérif and Bennani, 2007; Cabanes and Bennani, 2008) can also be used in combination with K-means to reduce the dimensionality. In this case SOM transform the data usually in 2D. SOM has several tuning parameters to fix such as the neighborhood function, the grid type and the learning rate. Finally, it is noted that displaying multidimensional clusters of models has never been studied up to now (objective 2, Section 1.5).

2.6 Sensor Placement

2.6.1 Examples in Engineering

The use of sensors for structural health monitoring has been increasing exponentially. Large numbers of sensors lead to enormous amounts of data (Brownjohn, 2007). Often, data are either redundant or meaningless, thereby complicating data management. It is thus important to select and place sensors so that maximum useful information is obtained. This process, which stands upstream from data interpretation, is known as sensor placement.

Sensors are increasingly used worldwide for tasks such as model-based diagnosis (Struss, 2006) and automatic control (Culler and Hong, 2004). The field of sensor configuration has emerged recently and research concerning sensor networks is now emerging in parallel. Examples of the interest in this field are the special issue of Communications of the ACM on wireless sensor networks in 2004 and the publication of a new journal, ACM Transactions on Sensor Networks, in 2005. Moreover, research evolves in managing these sensor networks mainly to satisfy growing user needs (Mullen et al., 2006). Work on sensors is also carried out in areas such as multi-sensor management (Xiong and Svensson, 2002). Reliability (Bagajewicz and Sanchez, 2000) and uncertainty (Guratzsch and Mahadevan, 2006) are currently studied in system identification as well as decision support systems (Sanchez-Marre et al., 2006).

One of the most concerned fields is civil engineering where measurements are rarely direct. Models are needed to relate measurements to causal information. Applications areas in this field include fault detection (Worden and Burrows, 2001), water networks (Robert-Nicoud et al., 2005c) and health monitoring (Meo and Zumpano, 2005). Installation of sensors and measurement campaigns are time-consuming tasks. This motivates the use of a framework for automating the sensor placement process. Li et al. (2006) use norm based techniques to place sensors. Parker et al. (2006) propose experimental validation of their genetic algorithm strategy for sensor placement. In Schulte et al. (2006), a forward-backward selection algorithm is envisaged for optimal sensor placement. Minimization of an information entropy criterion is used in Papadimitriou et al. (2000) and Pareto optimal concept in Papadimitriou (2005). All of these studies involved structural dynamics contents and have yet to be evaluated with static measurement data.

2.6.2 Sensor Placement for System Identification

One of the most important reasons for making measurements is system identification (Ljung, 1999), where the idea is to understand the behavior of a structure. In this case, the challenge is to determine the true state of the structure according to measurements. System identification can be model-based (see Section 2.5.2). Configuring a measurement system that best separates candidate models is necessary for effective system identification. Different methods can be used to measure the separation between predictions. For example, Robert-Nicoud et al. (2005b) uses the notion of entropy. Variance is compared to entropy for the measure of model separability and the entropy is found to be better. The expression used to calculate entropy is the Shannon's entropy function (Shannon and Weaver, 1949). This expression comes from the field of information theory and it formulates the disorder within a set. In our case, a set is an ensemble of model predictions for a system identification task. The disorder, and therefore entropy, is at maximum when model predictions show wide dispersion. At the best measurement locations, model predictions should have maximum variation (Robert-Nicoud et al., 2005b).

Sensor placement is also an active area of research in fault diagnosis (Raghuraj et al., 1999; Commault et al., 2006; Frisk and Krysander, 2007). A description of an approach for identifying required measurements for performing diagnosis can be found in de Kleer and Williams (1987). They use entropy as a measure of probabilities of candidate models in order to identify measurements to be taken. However, their approach is part of a diagnosis methodology and requires measurements from previous sensors in order to locate the next sensor. Raghuraj et al. (1999) use a graph-based formulation for the sensor placement problem. In their work, however, they do not find all minimal sensor sets that could diagnose the system. In Travé-Massuyès et al. (2006) sensor placement is based on analytical redundancy relations (ARR). However, their strategy has a high complexity due to the numbers of ARR. In Frisk and Krysander (2007), the sensor placement problem is formulated as a graph theoretical problem. A fault f is detected if there is an observation consistent with fault mode f and not consistent with no-fault mode. Sensors are placed so that faults in different components are the best isolated. Theoretical proofs are given in Krysander and Frisk (2007). These techniques are based on redundancy and therefore, the number of sensors needed to diagnose the fault may be important.

Within the context of system identification, Robert-Nicoud et al. (2005b) place each sensor according to a greedy algorithm. In greedy algorithms, strategies that accept a less attractive alternative for a better overall solution do not exist. While finding an answer, the best immediate, or local, solution is always selected. Although finding the overall, or globally, optimal solution for some optimization problems, greedy algorithms may find non optimal solutions for other problems.

Unlike greedy algorithms, global search algorithms are intended to find the best solution among all possible. Most popular global search algorithms are simulated annealing (SA) (Kirkpatrick et al., 1983) and genetic algorithms (GAs) (Holland, 1975; Goldberg, 1989). More recent proposals include the repeated weighted boosting search (Chen et al., 2005b). The probabilistic global search lausanne (PGSL) (see Section 2.5.2) is used. Only three search parameters need to be tuned and performance has been shown to be as good or better than SA and GA (Raphael and Smith, 2003a). Up to now, PGSL has never been used for sensor placement. In addition, no comparison has been made between greedy and global search for sensor placement (objective 3, Section 1.5).

Data mining has already been used for sensor data management as well as for interpretation in order to place sensors (Mani, 2003; Ailamaki et al., 2003; Faschingbauer and Scherer, 2007). However, clustering is not used in these examples. Combination of both sensors and clustering concepts has been found in the literature in several fields of research. In Banta and Abidi (1996), a sensor placement strategy using clustering is used for acquisition of 3D models. The aim, however, is to obtain closely packed clusters. In Younis et al. (2003), clustering is used to partition sensor networks into clusters. Ideas of cluster compactness and separation are not used. In Mukherjee and Memik (2006), the position of sensor is determined by sensor centers. In this particular application, the number of clusters is known in advance. Kumar (2003) uses clustering as a view for connected information space containing sensors. Although clusters of sensors are studied, no work related to sensor placement is done. No work directly related to using clustering for improving sensor placement has been found (objective 3, Section 1.5).

The use of data mining in addition to the iterative aspect of the methodology makes the integration of engineers in the process important. Firstly, data mining results have to be visualized. Visualization is important for helping engineers make decisions. The objective is to represent results obtained with data mining methods to engineers accurately and simply in order to carry out system identification. Secondly, engineers need feedback from the system. That is to say, the system must show the results in an easily understandable manner. This way, engineers can understand them and consequently draw conclusions. Secondly, there is a need for engineers to interact with the system in order to improve the system identification task (objective 2, Section 1.5). The motivations for these needs, as mentioned in Stalker and Smith (2002), are that usually engineers work with incomplete knowledge and problems that are context dependent.

2.7 Definitions

As written in Section 2.1, the goal of data mining is to transform data into knowledge. To ensure clarity, terms such as data, information and knowledge are defined in this section. In this work, data mining is applied in civil engineering. Computer scientists (e.g. the author) and civil engineers (users) have different terminologies. Certain words are specific to one field and unknown to researchers in the other domain. The situation can even be worse with partially known terminology: the same word can have two different meanings if it is read by a computer scientist or a civil engineer. These issues are discussed in Section 2.7.2. This thesis also contains several abbreviations. They are explained in Chapter 8.1.

2.7.1 Data, Information, Knowledge and Wisdom

The aim of data mining is to draw understandable knowledge from raw data. Behind these notions of data and knowledge, a more complex hierarchy exists. This hierarchy originates independently from knowledge management, design and information science (Sharma, 2005). In knowledge management, the data information knowledge wisdom (DIKW) hierarchy or pyramid has been initiated by Cleveland (1982), Zeleny (1987) and Ackoff (1989) separately. Figure 2.3 shows the pyramid representing the DIKW hierarchy.

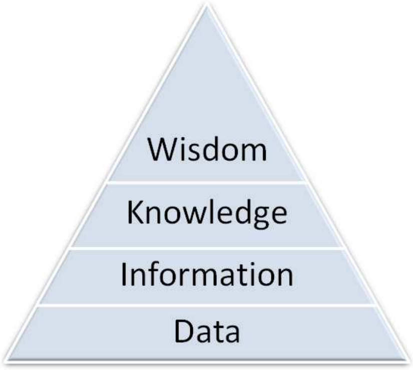


Figure 2.3: Pyramid of the data information knowledge wisdom (DIKW) hierarchy.

Zeleny (1987) translates the different parts of the DIKW hierarchy respectively by *know-nothing*, *know-what*, *know-how* and *know-why*. Ackoff (1989) proposes comprehensive definitions for such terms (Bellinger et al., 2005). He writes that "*[data] are products of observation*". It simply exist and has no significance. *Information*, which is inferred from data, answers questions such as *who*, *what*, *where*, *when* and *how many*. It consists of data linked together by relational connections. *Knowledge* is know-how and is acquired through learning. Knowledge is a useful collection of information. Ackoff (1989) proposes an additional layer named *Understanding*. It represents the why and allows to synthesize new knowledge from previous one. Finally, *Wisdom* is the ability to evaluate any choice. As written in Bellinger et al. (2005), "*it asks questions* *to which there is no (easily-achievable) answer*". More details about these definitions are given in Ackoff (1989). In the design domain, the DIKW hierarchy is found in Cooley (1987). In information science, Cleveland (1982) mentions a hierarchy for information, knowledge and wisdom. Figure 2.4 is an illustration of a part of this hierarchy.

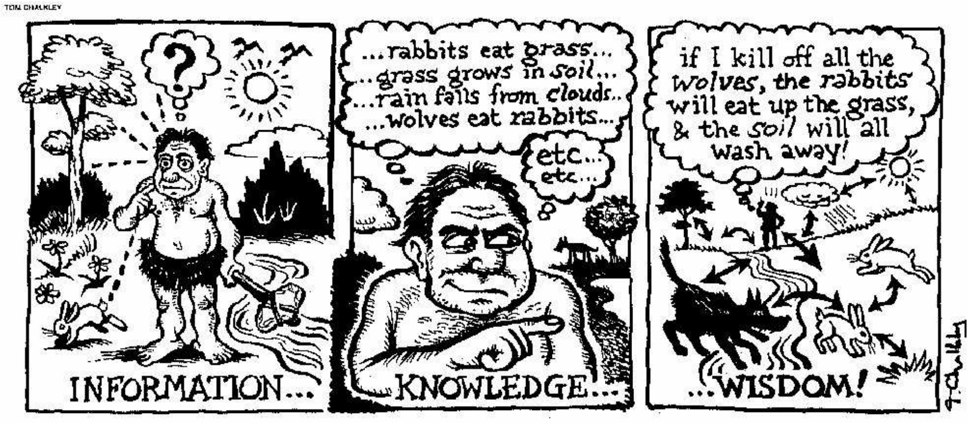


Figure 2.4: Illustration of information, knowledge and wisdom. Originally published in THE FUTURIST (1992). Used with permission from the World Future Society, 7910 Woodmont Avenue, Suite 450, Bethesda, Maryland 20814. Telephone:301-656-8274; www.wfs.org.

According to Cleveland (1982), this hierarchy is mentioned for the first time by T.S. Eliot - a poet - in 1934. The following quote is taken from Eliot (1934):

> "*Where is the Life we have lost in living?*
> *Where is the wisdom we have lost in knowledge?*
> *Where is the knowledge we have lost in information?*"

It is noted that other hierarchies exist. For example, Raphael and Smith (2003b) propose a simpler ontology. First, the notion of wisdom is not included. Second, information is defined to include data and knowledge. With this representation, knowledge is a type of information, along with data. Knowledge is defined as data that are linked together. Tuomi (1999) proposes an inverted hierarchy, where data emerge only after information is obtained, which is the case after obtaining knowledge. The author is also aware of the fact that definitions of these terms can vary (Zins, 2007). This thesis is written with a computer science point of view. The author agrees on the DIKW hierarchy and belongs to the people thinking that, for example, *Information Science* should be named *Knowledge Science* (Zins, 2006). Therefore, concepts such as data and knowledge are used with their meaning coming from the DIKW hierarchy. For more details, a comprehensive study of the meaning of these terms is done through the comparison of 130 definitions from the information science point of view (Zins, 2007). Spiegler (2003) has written a precise description of data, information and knowledge and especially the connection between them.

In mathematics and computer science, several expressions are used to define the *knowledge discovery* process. In Frawley et al. (1992), it is defined as the "*nontrivial extraction of implicit, previously unknown, and potentially useful information from data*". The difference between data mining and knowledge discovery is subtle. Data mining can be seen as the cause (what is done) and knowledge discovery the effect (what is obtained) (Pal and Jain, 2005). A common term used in the literature is *machine learning.* Distinctions between *machine learning* and *data mining* are given in Section 2.1. Another expression often used is *pattern recognition.* Whereas with pattern recognition the goal is usually known in advance (the question to be answered is known), this is not the case with data mining (Pal and Jain, 2005). In data mining applications, data are collected and interesting knowledge is thus inferred. In the remainder of this thesis, the expression data mining is used.

2.7.2 Computer Scientists and Civil Engineers

From the data mining point of view, several engineering applications exist. Grossman et al. (2001) describe a range of application of data mining to solve common engineering issues. Langley and Simon (1995) briefly describe applications of data mining for tasks such as diagnosis, monitoring, forecasting and automation. From the civil engineering viewpoint, data mining is an emerging field. Several papers using neural networks for civil engineering applications have been published (Abudayyeh et al., 2006). A comprehensive study about application of data mining to engineering problems has been done (Reich and Barai, 1999). Although most of the terminology is distinct for these two fields, some words are used for different purposes. This is the case with the word *model* which has a different meaning for civil engineers who perform system identification from the meaning used by those in data mining.

In system identification, the term model often denotes a representation of a physical entity that is used for simulation and analysis, for example, using a finite element software. A model is defined as a set of values for given parameters. These parameters represent the system one attempts to identify. A model thus contains values of system parameters (Robert-Nicoud et al., 2005a). Examples of system identification models are given in Robert-Nicoud et al. (2005a). More details about models can be found in Section 2.5. In data mining, algorithms usually build models from input data sets. This model is then used for predictive or descriptive purposes. Decision trees, neural networks and support vector machines are examples of such models. As written in Tan et al. (2006), a "*model generated by a learning algorithm should both fit the input* *data well and correctly predict the class label of records it has never seen before*". A model can be seen as an expert (Witten and Frank, 2005). Thus, a *data mining model* in computer science, has a completely different meaning from a *system identification model* in civil engineering. In the remainder of this thesis, and since data mining will be integrated in a system identification methodology, the word model refers to the system identification model.

2.8 Summary

Previous sections have shown strengths and weaknesses in the research field related to this work. Regarding strengths, one can mention K-means, one of the most popular techniques for clustering (Jain and Dubes, 1988). To estimate the correct number of clusters, several indices have been proposed (Bezdek and Pal, 1998; Halkidi et al., 2001; Wu and Chow, 2004; Kim and Ramakrishna, 2005; Yang et al., 2006). Indices such as Davies-Bouldin (Davies and Bouldin, 1979) and Calinski-Harabasz (Calinski and Harabasz, 1974) have recently been reported to be among the best (Maulik and Bandyopadhyay, 2002; Kim and Ramakrishna, 2005). When selecting relevant features, Kohavi and Sommerfield (1995) recommends to use wrapper approaches since they usually provide better results. Support vector machines (SVM) have been successfully used for wrapper feature selection (Liu and Zheng, 2006). Wrapper techniques using probabilistic approaches such as simulated annealing and genetic algorithms have been found to be the most efficient (Oh et al., 2004; Huang et al., 2007). Probabilistic global search lausanne (PGSL) is a global search algorithm that is competitive with genetic algorithms (Raphael and Smith, 2003a). Robert-Nicoud (2003) proposed a methodology for generating multiple models for the system identification task. Robert-Nicoud et al. (2005b) and Papadimitriou et al. (2000) used Shanon entropy in a greedy algorithm for initial sensor placement.

Studies of previous work indicated several weaknesses. Thesis objectives (Section 1.5) are indicated in parentheses:

- A multiple model system identification approach has been developed. However, it is limited in interpreting models that are identified. Engineers have to examine candidate models manually (objective 2).

- Although often used in engineering, data mining has never been applied to mine model parameters in system identification. Most literature mentions the use of data mining for predictive purposes rather than for descriptive goals (objective 2).

- In clustering, the number of clusters is usually not known in advance. Although several cluster validity methods exist, they only work for certain data sets and are usually undefined for a single cluster (objective 1).

- Wrapper feature selection approaches are the most effective, for example when combining a sequential search technique with support vector machines (SVM). However, sequential search requires the assumption of monotonicity[4]. Other proposals, such as genetic algorithm based approaches, need complex tuning of search parameters (objective 1).

- Most sensor placement strategies involve structural dynamics contexts. They have yet to be evaluated with static measurement data. Furthermore, complete iterative methodologies for sensor placement, integrating data mining as a decision support tool, and involving engineers are rare (objective 3).

- Up to now, PGSL has not been used for sensor placement. In addition, no comparison has been made between a greedy algorithm and global search (PGSL) for initial sensor placement (objective 3).

- Most data mining studies focus on artificial benchmarks or, in the best case, on laboratory tests. There are very few examples of successful applications of data mining on real engineering examples (objective 4).

The above shortcomings are addressed in this thesis. Chapter 3 presents a new index for finding the correct number of clusters in a data set. A new feature selection algorithm combining SVM and PGSL is proposed in Chapter 4. Chapter 5 shows how to integrate data mining in an overall methodology for system identification. A comparison between greedy algorithm and global search (PGSL) is done in Chapter 6. A methodology for iterative sensor placement using clustering is also introduced. Finally, conclusions and future work are given in the last chapter.

[4] This assumption means that adding new features never degrades the performance.

Chapter

3 A New Cluster Validity Index

"*While it is entertaining to find patterns in clouds, it is pointless and perhaps embarrassing to find clusters in noise.*" Tan et al. (2006)

Overview

This chapter presents the first contribution of this research, a new validity index for clustering. This new index provides reliable estimates of the number of clusters in a data set. First, existing selected validity indices are described for comparison with the proposed index. The new index is then introduced and tested on benchmark data sets. Finally, limitations of the new index are evaluated.

3.1 Existing Indices

Since it is not feasible to test every existing index, six validity indices that are suitable for hard partitional clustering are used to compare results with those of the new validity index. These indices serve as a basis for evaluating results from the proposed index on benchmark data sets. Notation for these indices have been adapted to provide a coherent basis. The metric used on the standardized data set is the Euclidean distance $d(x, y)$. The Euclidean distance is chosen since it is easily understood by non-specialists.

Dunn index: One of the oldest and most cited indices is proposed by Dunn (1974). The Dunn index (DU) identifies clusters which are well separated and compact. The goal is therefore to maximize the inter-cluster distance while minimizing the intra-cluster distance. The Dunn index for k clusters is defined by Equation 3.1:

$$DU_k = \min_{i=1,\ldots,k} \left\{ \min_{j=1+1,\ldots,k} \left(\frac{diss(c_i, c_j)}{\max_{m=1,\ldots,k} diam(c_m)} \right) \right\} \tag{3.1}$$

where $diss(c_i, c_j) = \min_{x \in c_i, y \in c_j} d(x, y)$ is the dissimilarity between clusters c_i and c_j and $diam(c_m) = \max_{x,y \in c_m} d(x, y)$ is the intra-cluster function (or diameter) of the cluster. If Dunn index is large, it means that compact and well separated clusters exist. Therefore, the maximum is observed for k equal to the most probable number of clusters in the data set.

Calinski-Harabasz index: This index (Calinski and Harabasz, 1974) is based on a ratio of between cluster scatter matrix ($BCSM$) and within cluster scatter matrix ($WCSM$). The Calinski-Harabasz index (CH) is defined as follows:

$$CH_k = \frac{BCSM}{k-1} \cdot \frac{n-k}{WCSM} \tag{3.2}$$

where n is the total number of points and k the number of clusters. The $BCSM$ is based on the distance between clusters and is defined in Equation 3.3:

$$BCSM = \sum_{i=1}^{k} n_i \cdot d(z_i, z_{tot})^2 \tag{3.3}$$

where z_i is the center of cluster c_i and n_i, the number of points in c_i. The $WCSM$ is given in Equation 3.4:

$$WCSM = \sum_{i=1}^{k} \sum_{x \in c_i} d(x, z_i)^2 \tag{3.4}$$

where x is a data point belonging to cluster c_i. To obtain well separated and compact clusters, $BCSM$ is maximized and $WCSM$ minimized. Therefore, the maximum value for CH indicates a suitable partition for the data set.

Davies-Bouldin index: Similar to the Dunn index, Davies-Bouldin index (Davies and Bouldin, 1979) identifies clusters which are far from each other and compact. The Davies-Bouldin index (DB) is defined according to Equation 3.5:

$$DB_k = \frac{1}{k} \sum_{i=1}^{k} \max_{j=1,\ldots,k, i \neq j} \left\{ \frac{diam(c_i) + diam(c_j)}{d(z_i, z_j)} \right\} \tag{3.5}$$

where in this case, the diameter of a cluster is defined as in Equation 3.6:

$$diam(c_i) = \sqrt{\frac{1}{n_i} \sum_{x \in c_i} d(x, z_i)^2} \tag{3.6}$$

with n_i the number of points and z_i the centroid of cluster c_i. Since the objective is to obtain clusters with minimum intra-cluster distances, small values for DB are interesting. Therefore, this index is minimized when looking for the best number of clusters.

Silhouette index: The silhouette statistic (Kaufman and Rousseeuw, 1990) is another well known way of estimating the number of groups in a data set. The silhouette index (SI) computes for each point a width depending on its membership in any cluster. This silhouette width is then an average over all observations. This leads to Equation 3.7:

$$SI_k = \frac{1}{n}\sum_{i=1}^{n}\frac{(b_i - a_i)}{\max(a_i, b_i)} \tag{3.7}$$

where n is the total number of points, a_i is the average distance between point i and all other points in its own cluster and b_i is the minimum of the average dissimilarities between i and points in other clusters. Finally, the partition with the highest SI is taken to be optimal.

Maulik-Bandyopadhyay index: A more recently developed index is named the I index (Maulik and Bandyopadhyay, 2002). For consistency with other indices it is renamed MB. This index, which is a combination of three terms, is given through Equation 3.8:

$$MB_k = \left(\frac{1}{k}\cdot\frac{E_1}{E_k}\cdot D_k\right)^p \tag{3.8}$$

where the intra-cluster distance is defined by $E_k = \sum_{i=1}^{k}\sum_{x\in c_i} d(x, z_i)$, E_1 being the value of E_k for $k = 1$ and the inter-cluster distance by $D_k = \max_{i,j=1}^{k} d(z_i, z_j)$. As before, z_i is the center of cluster c_i. The correct number of clusters is estimated by maximizing Equation 3.8. According to Maulik and Bandyopadhyay (2002), p is chosen to be two.

Geometric index: The last index used for comparison is the Geometric index (Lam and Yan, 2005). One of its advantages is its ability to accommodate data with clusters of different densities as well as clusters that overlap. The geometric index (GE) is defined by Equation 3.9:

$$GE_k = \max_{1\leq r\leq k}\frac{\left(2\sum_{j=1}^{d}\sqrt{\lambda_{jr}}\right)^2}{\min_{1\leq q\leq k, r\neq q} d(z_r, z_q)} \tag{3.9}$$

where d is the dimensionality of the data and λ_{jr} is the eigenvalue of the covariance matrix from the data. While the numerator is the squared eigen-axis length, the denominator represents the inter-cluster distance. The optimal solution is found by minimizing the index over the number of clusters.

3.2 Score Function

A typical goal of clustering is to maximize the inter-cluster distance (separability) while minimizing the intra-cluster distance (compactness) over iterations. The index developed in this work - called the score function (SF) - is based on these two concepts. This section gives details related to the way the SF has been developed and the ideas that have lead to its development. The following definitions are used. Firstly, the Euclidean distance is used to measure to what degree two data points are separated. Secondly, the size of the i-th cluster, n_i, is given by the number of points it contains.

Two concepts used in the proposed index are the “between class distance” (*bcd*), representing the separability of clusters, and the “within class distance” (*wcd*) representing the compactness of clusters. Three approaches are commonly used to measure the distance between two clusters: single linkage, complete linkage and comparison of centroids. DU is based on single linkage and has a complexity of $O(n^2)$. Although SI does not fit well into these three categories, its computational complexity is the same as the first two. DB, MB and GE compare centroids. CH follows the third approach since the distances of centroids from the overall mean of the data are determined. The main advantage of using the distance from the overall mean of the data is that the minimum and maximum are not used when comparing centroids. The minimum and maximum are sensitive to outliers. In this work, the score function uses the third approach since the first two have high computational costs (Halkidi et al., 2001). The *bcd* is given by Equation 3.10:

$$bcd = \frac{1}{nk}\sum_{i=1}^{k} d(z_i, z_{tot})^2 \cdot n_i \tag{3.10}$$

where n is the total number of data points, k is the number of clusters, z_i its centroid of the current cluster and z_{tot} the centroid of all the data points. The main quantity in the *bcd* is the distance between z_i and z_{tot}, $d(z_i, z_{tot})$. As in the CH index, each distance is weighted by the cluster size n_i to limit the influence of outliers. This has the effect to reduce the sensitivity to noise. Like all other tested indices, n is used to avoid the sensitivity of *bcd* to the total number of points. Finally, the value of k in the denominator is used to penalize the addition of new clusters. Thus, *bcd* is reduced as k increases. In this way, the limit of one point per cluster is avoided. The *wcd* is given by Equation 3.11:

$$wcd = \frac{1}{k}\sum_{i=1}^{k} \sqrt{\frac{1}{n_i}\sum_{x \in c_i} d(x, z_i)^2} \tag{3.11}$$

Computing values for *wcd* involves determining the distance between each point and the centroid of its cluster. Again, n_i is used for taking into account the size of clusters. The mean is taken over the k clusters. A graphical representation of distances used in both Equation 3.10 and 3.11 can be found in Figure 3.1.

With Equations 3.10 and 3.11, *bcd* and *wcd* are independent of the number of data points. The main idea, as stated in the beginning of this section, is to maximize Equation 3.10 while

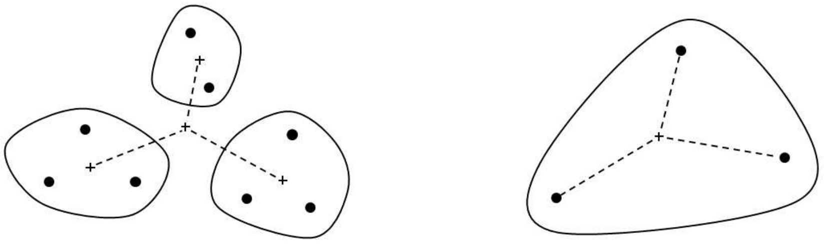


Figure 3.1: Graphical representations of *bcd* (left) and *wcd* (right).

minimizing Equation 3.11. Therefore, compact and well separated clusters are aimed at. This can be done by maximizing the ratio of *bcd* and *wcd*, noted as the index, as shown in Equation 3.12:

$$Index = \frac{bcd}{wcd} \tag{3.12}$$

Equation 3.12 has two difficulties. The first difficulty occurs when the clusters are perfect. Here, the value of the index in Equation 3.11 is zero and the ratio of Equation 3.12 is indeterminate. Therefore, the ratio cannot be used in this form in the case of perfect clusters. The second difficulty occurs when there is only one cluster in the data. In this case, Equation 3.10 is zero and thus the ratio of Equation 3.12 is zero. This is not desirable since it means that the single cluster case is not comparable with other cases. A possible solution to these difficulties involves the use of the exponential notation. Consequently, a new value for the index is proposed:

$$Index = \frac{e^{bcd}}{e^{wcd}} = e^{bcd-wcd} \tag{3.13}$$

A third difficulty is related to bounds. All other tested indices have no bounds. It is thus difficult to appreciate the results of such indices. Since the "distance" to either perfect clusters or no cluster at all is not known. The upper bound allows the examination of how close the current clusters are to the perfect cluster case. The bounds for the index in Equation 3.13 are $]0, \infty[$. It is also desirable to avoid very large numbers for computational reasons. Again, exponential notation is used. Avoiding all of these difficulties leads to the formula for the score function (SF), defined by Equation 3.14:

$$SF = 1 - \frac{1}{e^{e^{bcd-wcd}}} \tag{3.14}$$

Thus, Equation 3.14 is maximized to obtain the most reliable number of clusters. The score function is now bounded by]0,1[and deals with the perfect cluster case and the single cluster case. The strength of the SF originates in part from the fact that it is built on ideas from several indices. Since it is not based on minimum/maximum values, it is not influenced by outliers. The size of clusters is taken into account in both *bcd* and *wcd*. The comparison of centroids is used in the place of single or complete linkage. This avoids the computational complexity. The number of clusters k is used to penalize the addition of clusters. Finally, the exponential notation is used to both accommodate single and perfect cluster cases and to define bounds. As can be seen through Equations 3.10 and 3.11, computational complexity is linear. If n is the number of data points, the proposed score function has a complexity of $O(n)$. Tests that have been conducted with benchmark problems indicate that this function provides good results. This is the subject of the next section.

3.3 Results

3.3.1 Number of clusters

In this subsection, there are two goals. The first goal is to test the score function on benchmark data sets. The second goal is to compare results between indices. k_{min} and k_{max} are taken to be respectively 2 and 10. If not explicitly stated, data sets used in this section are composed of 1000 points in two dimensions. As written in Section 2.3.1, one drawback of K-means is the random choice of centroid locations. To limit the impact of the problem, K-means is run several times. This number is a compromise between better results and computational time needed. After experimental tests, this value is chosen to be 20. Therefore, the best K-means result over 20 runs is taken. More details are given in Section 5.2.2.

Example 1: In the first data set, *Unbalanced*, three clusters of different compactness are present (see Figure 3.2a). Clusters of varying densities is an important issue (Chou et al., 2004). Table 3.1 shows that, unlike other indices, Dunn is not able to correctly estimate the number of clusters (three). This is due to the definition of the Dunn index. The diameter, for example, can be affected by outliers since it is not based on a mean value.

Example 2: The second data set, *Overlapped*, consists of three clusters. Two of these clusters overlap (see Figure 3.2b). This data set is important since the ability to deal with overlapping clusters is one of the best ways to compare indices (Bouguessa et al., 2006). Table 3.2 shows the results for this data set. GE overestimates the number of clusters. A weakness of GE is to be based on the minimum distance between two clusters. This leads to problems when dealing with overlapping clusters. DU, DB and SI identify the two overlapping clusters as one cluster. This is due to their dependence on a minimum or maximum value. This is not the case with CH, MB and SF which correctly estimate the three clusters.

Example 3: This data set, named *Noisy*, contains seven clusters with an additional noise. It

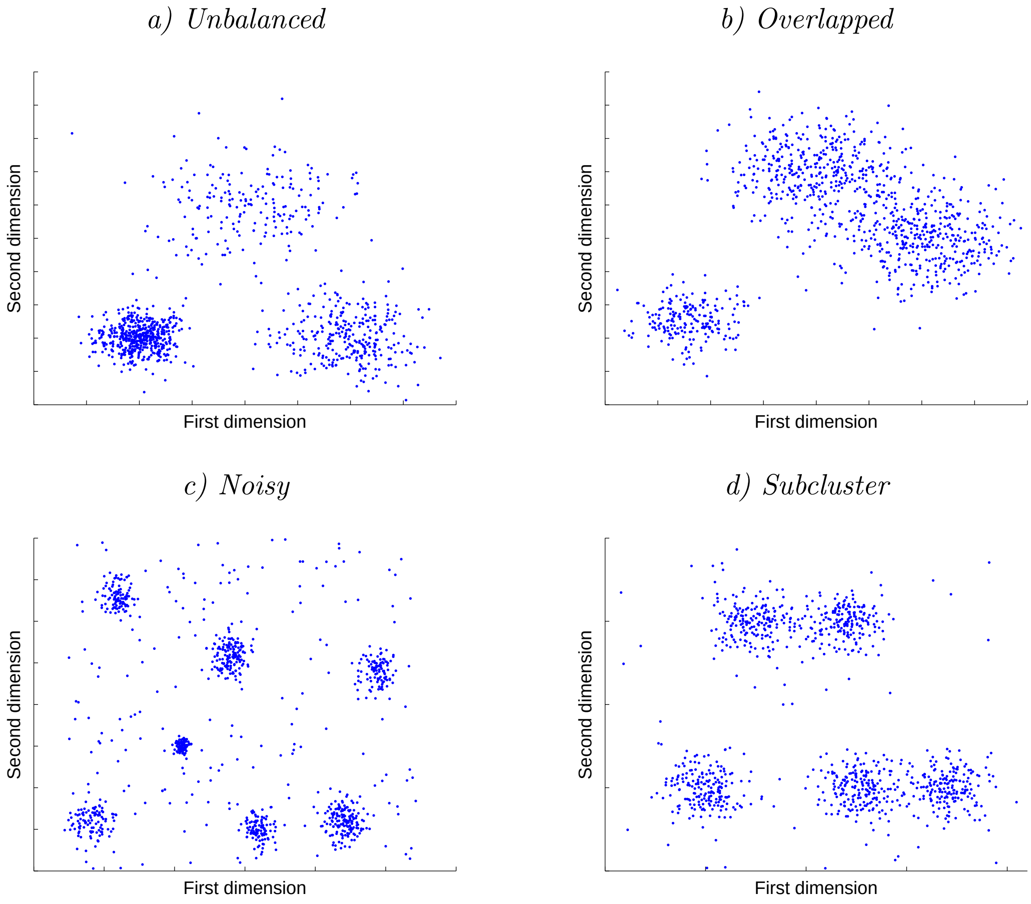


Figure 3.2: Four artificial data sets, *Unbalanced*, *Overlapped*, *Noisy* and *Subcluster*.

k	2	3	4	5	6	7	8	9	10
DU	**0.056**	0.036	0.022	0.014	0.017	0.008	0.009	0.007	0.010
CH	950.6	**3453.3**	2725.0	2455.3	2111.1	2214.6	1961.3	2160.6	2107.9
DB	0.800	**0.457**	0.697	0.688	0.784	0.762	0.852	0.846	0.819
SI	0.682	**0.893**	0.819	0.716	0.714	0.728	0.593	0.521	0.565
MB	2.746	**7.600**	6.245	5.106	4.986	4.236	3.819	3.971	3.618
GE	3.257	**1.720**	1.842	1.876	1.939	1.931	2.043	2.107	2.212
SF	0.489	**0.648**	0.627	0.617	0.603	0.595	0.593	0.584	0.584

Table 3.1: Results of the seven validity indices on the *Unbalanced* data set (example 1). The best result on 20 runs is taken. The data set is shown in Figure 3.2a. Bold numbers show maximum values for all indices except DB and GE, where minimum values are desired. This indication is used for Tables 3.1 to 3.6. The correct number of clusters is $k = 3$.

k	2	3	4	5	6	7	8	9	10
DU	**0.091**	0.011	0.012	0.016	0.016	0.017	0.021	0.014	0.019
CH	1346.2	**2497.6**	2154.2	1996.0	1941.3	1887.3	1834.7	1772.8	1744.2
DB	**0.543**	0.562	0.653	0.809	0.784	0.766	0.767	0.753	0.731
SI	**0.779**	0.771	0.672	0.611	0.582	0.583	0.589	0.597	0.592
MB	4.426	**5.520**	4.646	3.800	3.208	2.827	2.592	2.374	2.061
GE	2.719	1.885	2.046	2.010	1.885	1.745	1.676	1.705	**1.625**
SF	0.577	**0.636**	0.612	0.593	0.588	0.582	0.579	0.577	0.576

Table 3.2: Results of the seven validity indices on the *Overlapped* data set (example 2). The data set is shown in Figure 3.2b. The correct number of clusters is $k = 3$.

can be seen in Figure 3.2c. It is rarely the case that clusters appear clearly in real situations. The data are often noisy and some indices are sensitive to noise as pointed out in Halkidi et al. (2001). Table 3.3 contains the results for this specific data set. It can be seen that DU, CH, DB, SI and MB overestimate the correct number of clusters. Presence of noise is too strong for these indices to correctly estimate the number of clusters. Only GE and SF are able to determine the seven clusters.

k	2	3	4	5	6	7	8	9	10
DU	0.038	0.038	0.064	0.070	0.077	0.077	0.067	0.075	**0.081**
CH	769.3	1018.6	1476.1	1722.4	2174.7	2849.9	3136.7	3201.4	**3294.3**
DB	1.108	0.700	0.608	0.500	0.457	0.465	**0.440**	0.481	0.486
SI	0.580	0.636	0.740	0.768	0.803	0.829	0.843	0.852	**0.860**
MB	1.640	1.744	3.069	3.845	5.457	7.370	**7.937**	7.136	6.148
GE	4.215	2.717	1.860	1.613	1.079	**0.952**	1.191	1.534	1.507
SF	0.419	0.513	0.567	0.590	0.604	**0.612**	0.605	0.601	0.601

Table 3.3: Results of the seven validity indices on the *Noisy* data set (example 3). The data set is shown in Figure 3.2c. The correct number of clusters is $k = 7$.

Example 4: The following data set, named *Subcluster*, contains five clusters, with two “pairs”. It is visible in Figure 3.2d. It can happen in real-life that data sets contain clusters which are closely grouped together. Existing indices developed for hard clustering may not be able to deal with such situations. Table 3.4 presents the results for this data set. More details about sub-cluster hierarchies can be found in Section 3.3.4.

k	2	3	4	5	6	7	8	9	10
DU	0.059	**0.069**	0.020	0.017	0.016	0.014	0.014	0.014	0.015
CH	979.3	2431.0	2647.7	**3774.1**	3351.0	3045.1	2833.7	2636.2	2550.7
DB	0.907	0.489	**0.467**	0.469	0.579	0.683	0.714	0.750	0.792
SI	0.657	**0.841**	0.821	0.810	0.735	0.729	0.677	0.635	0.661
MB	1.890	9.523	16.206	43.550	**54.825**	36.058	49.388	43.522	41.192
GE	3.793	1.235	1.147	**1.122**	1.510	1.435	1.525	1.570	1.658
SF	0.480	0.636	0.638	**0.641**	0.627	0.618	0.613	0.606	0.601

Table 3.4: Results of the seven validity indices on the *Subcluster* data set (example 4). The data set is shown in Figure 3.2d. The correct number of clusters is $k = 5$.

Example 5: The next data set, named *Wine*, is a real-life data set (Merz and Murphy, 1996). It is made of 178 points in 13 dimensions. *Wine* contains 3 clusters. Results of the seven indices are given in Table 3.5. Here, CH, DB, SI and SF are able to discover the three clusters. While MB underestimates the number of clusters, DU and GE over-estimate the correct value.

k	2	3	4	5	6	7	8	9	10
DU	0.160	0.232	0.232	0.210	0.190	0.235	0.212	**0.239**	0.234
CH	69.52	**70.94**	56.20	47.17	42.23	38.26	36.26	34.33	32.73
DB	1.505	**1.257**	1.501	1.481	1.402	1.421	1.307	1.423	1.425
SI	0.426	**0.451**	0.418	0.407	0.390	0.368	0.313	0.348	0.353
MB	**5.689**	5.391	3.546	3.445	2.682	2.008	1.893	1.733	1.380
GE	97.747	99.209	104.685	101.154	108.083	97.892	93.336	**86.958**	91.108
SF	0.269	**0.385**	0.314	0.324	0.253	0.240	0.231	0.233	0.242

Table 3.5: Results of the seven validity indices on the *Wine* data set (example 5). The data set is made of 178 points in a 13 dimensions. The correct number of clusters is $k = 3$.

Example 6: In this last example, the *Cancer* data set is used (Merz and Murphy, 1996). It contains 569 points in 30 dimensions. *Cancer* is composed of 2 clusters and is a good example of a problem in a relatively high dimensional space. Results are presented in Table 3.6. Three indices, CH, SI and SF, are able to deal with these two clusters represented in 30 dimensional space. DU, DB, MB and GE are not able to catch the trend due to either the cluster shapes or the high dimensionality of the data.

k	2	3	4	5	6	7	8	9	10
DU	0.076	0.078	0.075	0.078	0.072	0.064	0.072	**0.079**	0.067
CH	**267.7**	197.1	159.0	140.4	128.8	118.6	109.7	103.3	98.1
DB	1.444	1.461	1.502	1.432	1.534	**1.391**	1.418	1.408	1.457
SI	**0.519**	0.492	0.441	0.427	0.279	0.257	0.259	0.244	0.228
MB	16.202	11.433	13.890	10.265	**26.346**	20.834	14.002	5.697	12.279
GE	2.599	2.497	2.426	2.558	2.946	2.546	2.231	2.273	**2.215**
SF	**0.657**	0.446	0.340	0.238	0.216	0.160	0.149	0.137	0.124

Table 3.6: Results of the seven validity indices on the *Cancer* data set (example 6). The data set is made by 569 points represented in 30 dimensions. The correct number of clusters is $k = 2$.

Table 3.7 summarizes the results of the application of the seven indices to four artificial and two real-life data sets. SF is the only index performing well on all data sets. The closest index, in term of good results, is CH. This is due to the similarity of the two equations. Both CH and SF takes into account the number and size of clusters. Among all, CH and SF are the only two indices to be based on a comparison of cluster centroid (z_i) with overall centroid (z_{tot}).

In our experiments, SF correctly identified the number of clusters in all six data sets. The SF successfully processes the standard case with clusters of different size and compactness (*Unbalanced*), overlapped clusters (*Overlapped*), clusters with noise (*Noisy*), groups of clusters (*Subcluster*) and multidimensional data (*Wine* and *Cancer*).

Data Sets	**DU**	**CH**	**DB**	**SI**	**MB**	**GE**	**SF**
Unbalanced	2(X)	3(O)	3(O)	3(O)	3(O)	3(O)	3(O)
Overlapped	2(X)	3(O)	2(X)	2(X)	3(O)	10(X)	3(O)
Noisy	10(X)	10(X)	8(X)	10(X)	8(X)	7(O)	7(O)
Subcluster	3(X)	5(O)	4(X)	3(X)	6(X)	5(O)	5(O)
Wine	9(X)	3(O)	3(O)	3(O)	2(X)	9(X)	3(O)
Cancer	9(X)	2(O)	7(X)	2(O)	6(X)	10(X)	2(O)

Table 3.7: Estimated number of clusters for six data sets and seven validity indices. Notation (O) and (X) respectively indicates when the correct number of clusters has been found or not.

To test the score function more completely, several other aspects are evaluated. For example, properties of perfect clusters and sub-clusters are challenges. The single cluster case has to be considered as well. Although not commonly studied in the literature, it may often happen in practice. Recent research studies by others on clustering validity indices, have been limited to cluster data from 2 to k_{max} clusters. Finally, a comparative study of all indices is done.

3.3.2 Perfect Clusters

The SF upper bound indicates the perfect cluster case. Proximity to this bound (1.0) is a measure of closeness of data sets to perfect clusters. The next two data sets are used to test how the SF deals with perfect clusters. The data sets *Perfect3* and *Perfect5* are made of 1000 points in 2D and contain three and five clusters respectively which are nearly perfect (i.e. with a very high compactness). Both data sets as well as their score function curve are given in Figure 3.3.

The correct number of clusters is identified in both situations. An interesting observation is related to the maximum value for the SF. In the first case (0.854), the maximum is higher than in the second one (0.772). This is due to the dependence of the SF on the number of clusters k. This can be seen in Equations 3.10 and 3.11. More details of the influence of k can be found in

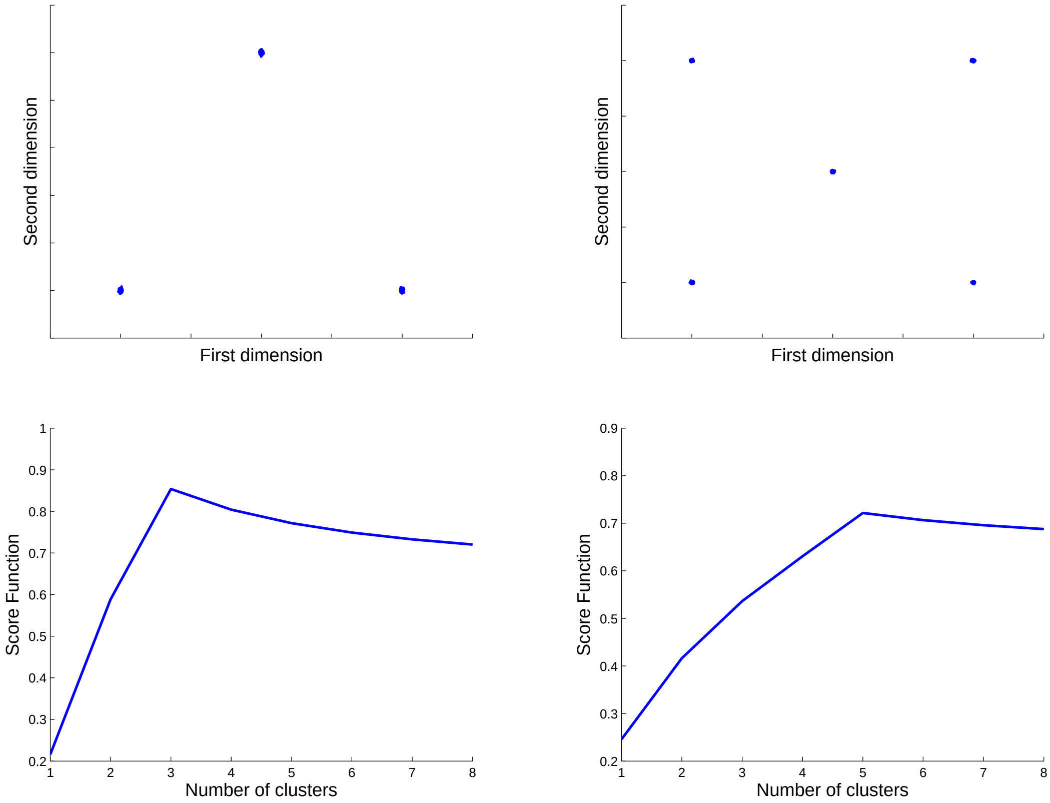


Figure 3.3: Comparison between two data sets: *Perfect3* and *Perfect5* containing respectively three and five clusters (top figures). Their respective score function curves is given as well (bottom figures).

Section 3.4.1. Finally, the SF gives an idea of how good clusters are through the proximity of the value of the index to its upper bound of unity.

3.3.3 Single Cluster

Before attempting to identify a single cluster, the definition of a cluster is clarified. Several definitions exist in the literature. A possible definition is given in Ling (1972). It states that a cluster is considered to be "real" if it is significantly compact or isolated or both at the same time. Concepts of compactness and isolation are based on two parameters that define internal properties of a cluster. The main drawback of such definitions is that they are often too restrictive; few data sets satisfy such criteria. Another way of testing for the existence of a single cluster is the null hypothesis (Engelman and Hartigan, 1969). However, this test is usually carried on univariate data.

An objective of the index, SF, is to accommodate the single cluster case. This case is not usually treated by other indices. In this subsection, k_{min} and k_{max} are taken to be respectively 1 and 8. Plot of SF with respect to the number of clusters provide indications related to how the single cluster case can be identified. Firstly, two situations may occur. Either the number of clusters is clearly located with a global maximum (Figure 3.4, left) or the SF has no clear global maximum (Figure 3.4, right).

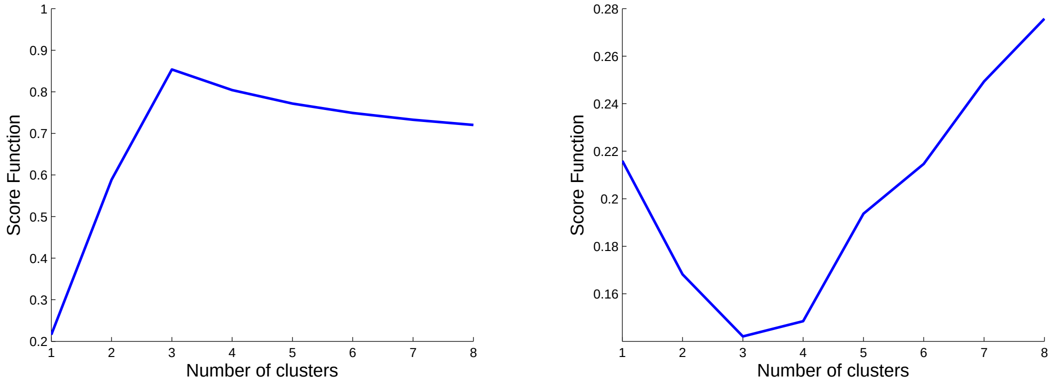


Figure 3.4: Difference of the SF trend with a data set containing three clusters (left) and single cluster (right).

Since in the first situation, the number of clusters is identifiable, the challenge lies in the second situation. In this case, there are two possibilities. They are: i) data forms a single cluster and ii) the correct number of clusters is higher than k_{max}.

In this paper, an empirical equation is proposed to distinguish between these two cases. For this purpose, three new data sets are introduced: *Single*, which contains 1000 points in 2D representing a single and spherical cluster, *SingleN* is the same cluster as *Single* plus added noise and *Single30* is a single cluster in a 30 dimensional space. It has been observed that in the single cluster cases, the value of the SF when $k = 2$, denoted as SF_2 is closer to the value for $k = 1$ (SF_1) than in other data sets. Therefore, the ratio between SF_1 and SF_2 is used as an indicator of single cluster as shown in Equation 3.15.

$$\frac{SF_1}{SF_2} \geq \alpha \tag{3.15}$$

where SF_1 and SF_2 are respectively the value for SF when $k = 1$ and $k = 2$. Results of this indicator on artificial and real-life benchmark data sets are given in Table 3.8.

Data sets	**Indicator**	**Data sets**	**Indicator**
Unbalanced	0.44	*SingleN*	**1.28**
Overlapped	0.37	*Single30*	**0.60**
Noisy	0.52	*Wine*	0.10
Subcluster	0.45	*Cancer*	0.01
Single	**0.61**		

Table 3.8: Results of the indicator (SF_1/SF_2) for nine benchmark data sets. Bold numbers indicate the single cluster cases.

According to Table 3.8, it is empirically stated that the data set is likely to contain a single cluster if Equation 3.15 is satisfied with $\alpha \cong 0.6$. Only three data sets containing a single cluster satisfy the condition in Equation 3.15.

3.3.4 Sub-clusters

Another case is the sub-cluster situation. This occurs when existing clusters can be seen as a cluster hierarchy. If this information can be captured by the validity index, more information about the structure of the data can be given to the user. The data set *Subcluster* in Figure 3.2d is an example of this situation. The index SF is compared with the previously mentioned indices on this topic. Figure 3.5 shows the evolution of each validity index with respect to the number of clusters.

In Figure 3.5, MB is not able to find the correct number of clusters (neither the sub-clusters, nor the overall clusters). In the case of DU, only the overall three clusters are detected. The reason is related to the distance measured between two clusters. Dunn uses the minimum

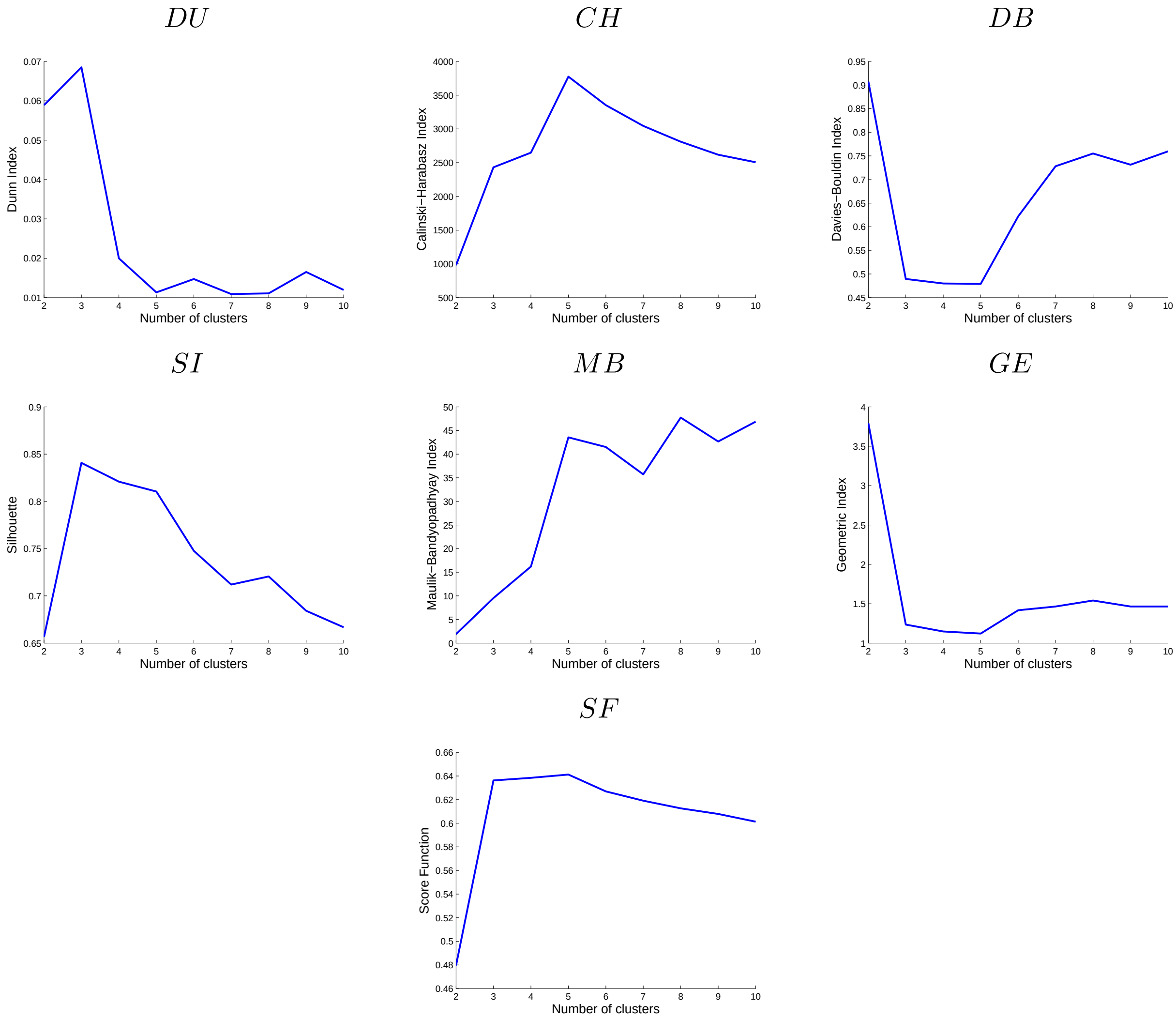


Figure 3.5: Comparison of DU, CH, DB, SI, MB, GE and SF for the sub-cluster case of Figure 3.2d. DB and GE must be minimized.

between points in two different clusters c_i and c_j. This strategy is limited in the case of the *Subcluster* data set since clusters overlap. With SI, although the sub-cluster hierarchy is visible, the recommended number of clusters is three. Finally, the indices that are able to find five clusters and show a peak at three clusters are CH, DB, GE and SF.

3.3.5 Comparative Study

All of these indices are different. Distinguishing aspects are their definition, their optimization strategy (minimum/maximum), their complexity or their definition with specific numbers of clusters such as $k = 1$. An index may have a tuning parameter to fix. This is the case of the MB index. The computational complexity is important. Although data sets tested here are small, other real-life examples may have tens or hundreds of thousands of points. In these cases, a validity index with a linear complexity is preferred over polynomial complexity. Since none of the other indices are bounded, the perfect cluster case is difficult to identify. When a value is obtained for a given index, it is usually difficult, or impossible, to know the proximity of the data set in relation to the perfect cluster situation. Since the single cluster case is usually not taken into consideration when developing indices, most of them are not defined for such a situation. This is the case for DU, DB, SI, MB and GE. All of these indices somehow involve the distance between two different clusters. In a single cluster case there is no such value. Although this problem does not appear for CH, the denominator of Equation 3.2 prevents the single cluster situation. Table 3.9 contains a summary of the important properties of the seven validity indices.

Properties	**DU**	**CH**	**DB**	**SI**	**MB**	**GE**	**SF**
On $k = 2..n$	max	max	min	max	max	min	max
Tuning parameters	no	no	no	no	yes	no	no
Complexity	O(n^2)	O(n)	O(n)	O(n^2)	O(n)	O(n)	O(n)
Bounds	]0,∞[	]0,∞[	]0,∞[	]-∞,∞[	]0,∞[	]0,∞[	]0,1[
Single cluster	no	no	no	no	no	no	emp.
Sub-clusters	no	yes	yes	no	no	yes	yes

Table 3.9: Properties of the seven compared validity indices. The single cluster line states whether the single cluster case is handled empirically (*emp.*) or not (*no*). The sub-clusters line shows *yes* for indices that are shown empirically to find sub-clusters and *no* otherwise. The tuning parameter for MB is p in Equation 3.9.

Except for indices DB and GE, which have to be minimized, all indices have to be maximized on $k = 2..n$. Only SF can be maximized on $k = 1..n$ due to its definition. The standard computational complexity is O(n), with n being the number of points, except for DU and SI ($O(n^2)$). This is due to the way these two indices calculate the distance between clusters. MB is the only index with a tuning parameter (p in Equation 3.9). This value is usually chosen to be two in the literature (Maulik and Bandyopadhyay, 2002; Kim et al., 2004). Concerning the bounds, the SF is the only index that has a lower and upper bound. This is a strong advantage with regards to other indices since it increases the usefulness of the value. SF is also the only index to be defined for the single cluster case ($k = 1$). Finally, only CH, DB, GE and SF reveal sub-clusters in data.

To summarize, the main drawbacks of the Dunn index are its computational load and its sensitivity to noise. It is useful for identifying clean clusters in data sets containing no more than hundreds of points. Although the Davies-Bouldin index gives good results for distinct groups, it is not designed to accommodate overlapping clusters. The Silhouette index is only suitable for estimating the first choice and therefore, it should not be applied to data sets with sub-clusters. The Maulik-Bandyopadhyay index has the particularity of being dependent on a user specified parameter. The Maulik-Bandyopadhyay and Geometric indices have been found to give bad results on multidimensional data sets. Although closely related to SF, CH has no upper bound and is not defined for $k = 1$. To conclude, the score function (SF) is competitive with existing validity indices on multidimensional and noisy data sets. It can handle perfect, single and sub-cluster cases. Finally, the SF is computationally efficient. It is in $O(n)$ where n is the number of data points.

3.4 Limitations

Since the SF depends on two exponentials, its evolution when the number of clusters equals the number of points requires specific study. In addition, data sets presented so far only contain hyper-spheroidal clusters. Additional tests with arbitrarily shaped clusters have been carried out. These issues are treated in the next subsections.

3.4.1 Score Function Evolution

In Section 3.2, the score function (SF) is bounded. Therefore, the SF has a lower bound of zero (no cluster structure) and an upper bound of one (perfect clusters). The purpose of the study in this subsection is to investigate the behavior of the SF for a large number of clusters. More specifically, the limits of the SF when the number of clusters k tends to the number of points n is studied. When k tends to n, the wcd tends to zero (see Equation 3.11). This is the case when each point represents a single cluster. The evolution of bcd is described by Equation 3.16:

$$\lim_{k \to n} bcd = \frac{1}{n^2} \sum_{i=1}^{n} d(x, z_{tot})^2 \tag{3.16}$$

Equation 3.16 can be rewritten as a function of the standard deviation, σ:

$$\lim_{k \to n} bcd = \frac{\frac{1}{n} \sum_{i=1}^{n} d(x, z_{tot})^2}{n} = \frac{\sigma^2}{n} \tag{3.17}$$

Consequently, the limit for SF when the $k \to n$ can be written as:

$$\lim_{k \to n} SF = 1 - \frac{1}{e^{e^{\sigma^2/n}}} \tag{3.18}$$

Two situations occur depending on the order of magnitude of σ^2 and n. They are presented in Equation 3.19:

$$\lim_{k \to n} SF = \begin{cases} 1 & for\ \sigma^2 \gg n \\ \sim 0.63 & for\ \sigma^2 \ll n \end{cases} \tag{3.19}$$

The second case is the most likely to happen when data are standardized (see Section 2.2.2). The evolution of the SF with both the *bcd* and the *wcd* is plotted with respect to the number of clusters. This number varies from $k_{min} = 1$ to $k_{max} = 30$. Results for the data set *Overlapped* are shown in Figure 3.6. Starting from zero (single cluster), the *bcd* has its maximum at $k = 2$ and decreases monotonically. The *wcd* starts with a high value and decreases monotonically as well. Concerning the SF, a maximum is observed at the correct number of clusters $k = 3$. The SF tends to 0.63 which is the limit found by Equation 3.19.

Figure 3.7 shows the results for the *Noisy* data set. After reaching a maximum for $k = 7$, the value of the SF stabilizes as predicted by Equation 3.19. The *wcd* decreases monotonically with a *knee* at $k = 7$. It is observed that the *bcd* closely follows the *wcd* starting at $k = 7$.

Finally, the case of a single cluster - *SingleN* - is studied (Figure 3.8). The *bcd* has a typical increase and then stabilizes. Instead of decreasing, the *wcd* grows from 1 to 3 clusters. This shows that k should not be increased. Thus, the SF has a minimum at $k = 3$ clusters and then grows slowly. This shows that in addition to validating Equation 3.15, the SF evolution indicates a single cluster presence in the data set.

Empirical tests have also been carried out. For a precise comparison of indices, the starting centroids are chosen to be the same in five runs. For each index, the best result over these five runs is taken as the correct number of clusters. Seven data sets that contain 16, 25, 36, 49, 64, 81 and 100 clusters are used. Limits on k, k_{min} and k_{max}, are chosen to be, respectively, 2 and 110. Results are given in Table 3.10.

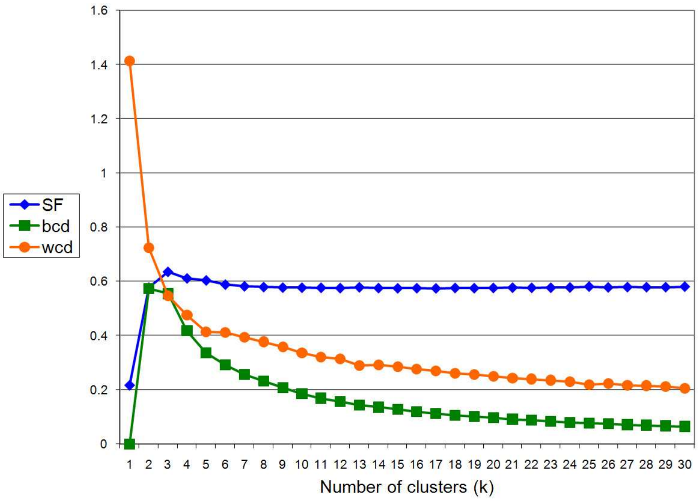


Figure 3.6: Evolutions of the SF and components *bcd* and *wcd* for the data set *Overlapped* from $k_{min} = 1$ to $k_{max} = 30$. For each value of k, the best over 20 runs is taken (see Section 3.3.1 for details).

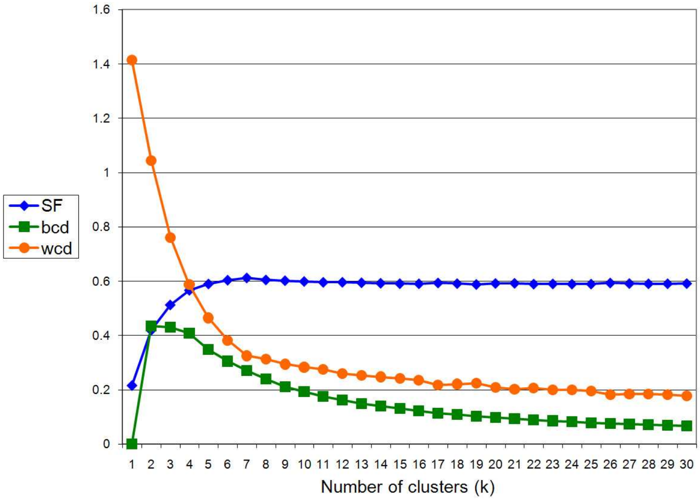


Figure 3.7: Evolutions of the SF and its main components *bcd* and *wcd* for the data set *Noisy* from $k_{min} = 1$ to $k_{max} = 30$. For each value of k, the best over 20 runs is taken.

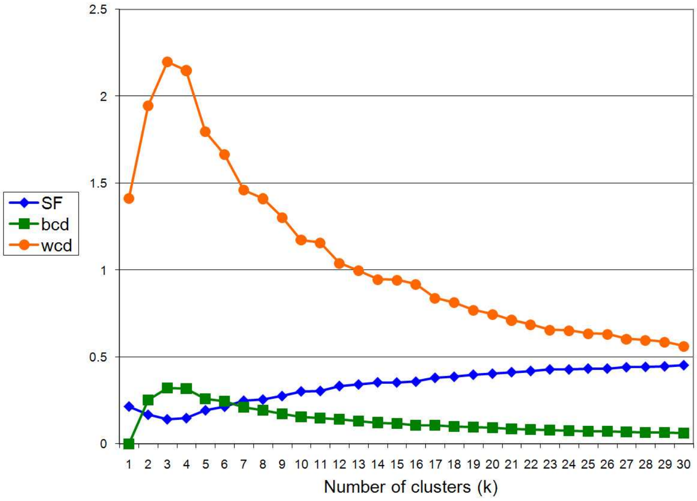


Figure 3.8: Evolutions of the SF and its main components *bcd* and *wcd* for the data set *SingleN* from $k_{min} = 1$ to $k_{max} = 30$. For each value of k, the best over 20 runs is taken.

Indices	**16**	**25**	**36**	**49**	**64**	**81**	**100**
DU	11	18	28	NA	NA	NA	NA
CH	20	34	84	76	68	73	84
DB	15	35	36	38	59	63	84
SI	15	28	52	50	83	98	108
MB	110	110	110	70	83	98	103
GE	NA	NA	NA	NA	NA	NA	NA
SF	20	34	58	76	68	74	84

Table 3.10: Estimated number of clusters for seven data sets containing respectively 16, 25, 36, 49, 64, 81 and 100 clusters. For simplifying this specific experiment, the best value over 5 runs with fixed K-means starting centroid locations are given. NA stands for not available (for example due to infinite or divide by zero issues).

It is observed that all indices have difficulty finding the correct number of clusters. This is likely due to the effect of the starting centroid locations. The probability of obtaining good centroid locations at the beginning - and therefore the correct number of clusters at the end - becomes smaller as the number of clusters increases (Tan et al., 2006). This issue can be resolved for many situations using methodologies to find better starting centroid locations (Pena et al., 1999).

However, the higher the number of clusters, the less effective these methodologies become. To illustrate the dependency of K-means results to initial centroid locations, an additional test has been carried out. The data set containing 49 clusters (see Table 3.10) is used again. However, in this case, initial centroid locations are chosen so that each starting position is in a distinct cluster. Aside from DU and GE, all indices find the correct number of clusters. This thus indicates that for high number of clusters, good results can be achieved when starting centroids are correctly placed.

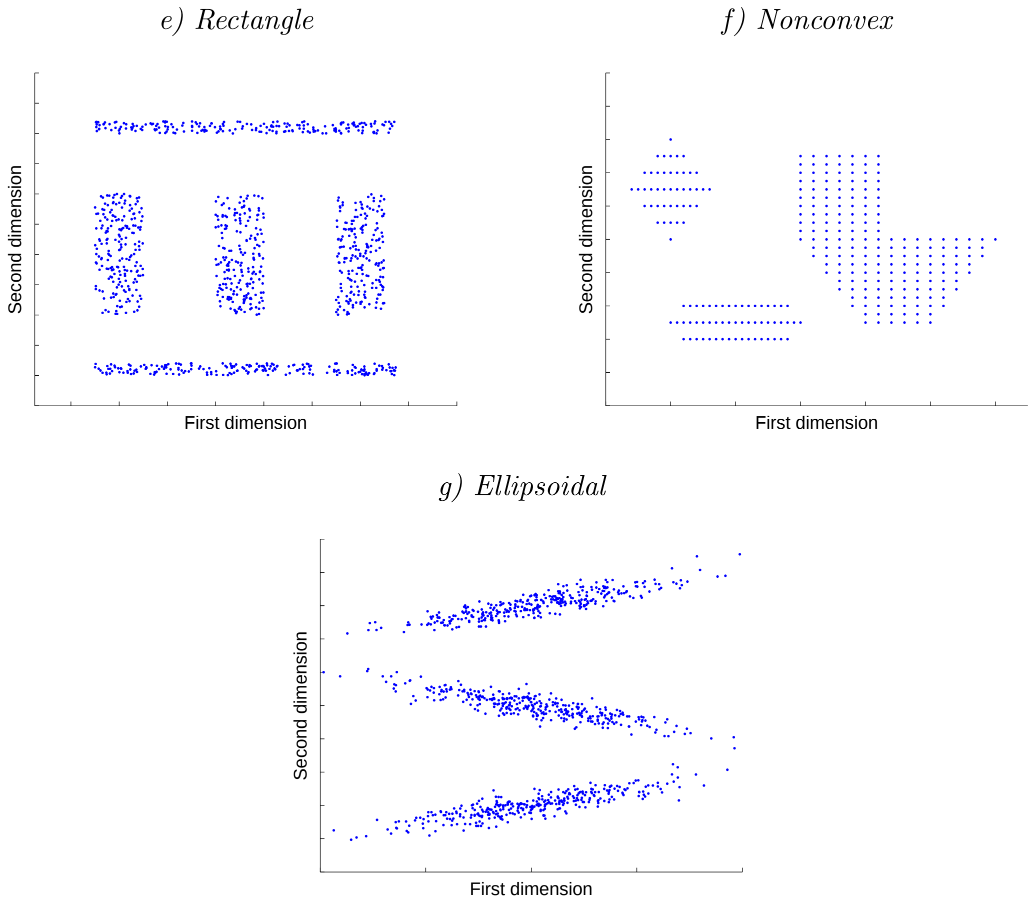
3.4.2 Arbitrarily Shaped Clusters

Up to this point, data sets used to test the different indices contained hyper-spheroidal clusters. The purpose of this subsection is to study arbitrarily-shaped clusters. Three new data sets are introduced. *Rectangle* contains 1000 points in 2D representing five rectangular clusters. The data set *Nonconvex* is made of 284 regularly-spaced points in 2D. It contains three clusters, one of them is not convex. Finally, *Ellipsoidal* is a data set made of 3 ellipsoidal clusters (1000 points in 2D). These data sets are shown in Figure 3.9.

Regarding the *Rectangle* data set, all indices overestimate the correct number of clusters (5). Results for different indices are: DU (9), CH (10), DB (10), SI (7), MB (10), GE (10) and SF (10). While it is clear that the SF is not able to find the real number of clusters, other indices have the same difficulty. This is mainly due to non-spheroidal shape of clusters. As stated in Tan et al. (2006), the clustering algorithm K-means is not reliable for non-spheroidal clusters.

Concerning the next data set, *Nonconvex*, the difficulty lies in the fact that one of the clusters is non-convex. In this case, the maximum value of SF is at $k = 4$ and although close, it overestimates the correct number of clusters (3). The following indices are also close to the real number of clusters: DU (2), DB (4) and MB (4). This is not the case for CH (6), SI (6) and GE (10). In the case of non-convex clusters, another clustering algorithm than K-means is advised.

In the last data set, *Ellipsoidal*, the clusters are far from spherical in shape. All indices fail when estimating the number of clusters (3). All indices overestimate the real number of clusters: DU (9), CH (10), DB (10), SI (10), MB (10), GE (10) and SF (10). Since all indices involves the calculation of some diameter or variance of clusters, the process fail when applied to strongly ellipsoidal shaped clusters. Therefore, a limitation of the score function, as well as other tested



e) Rectangle

f) Nonconvex

g) Ellipsoidal

Figure 3.9: Three new artificial data sets. *Rectangle* and *Ellipsoidal* contain 1000 points in 2D while *Nonconvex* is made of 284 points in 2D.

indices using K-means, is their restriction to data sets containing hyper-spheroidal clusters.

3.5 Conclusions

A new cluster validity index called score function has been proposed in this chapter. This leads to a new method to reliably estimate the number of clusters in a data set. The following conclusions come out of this chapter:

- Evaluation of results obtained through clustering is not straightforward. In addition, one issue of K-means concerns the user-defined number of clusters. The score function that has been developed in this work addresses both of these issues.
- The score function is found to be a correct estimator of the number of clusters in a data set. It is comprehensive since it can handle various situations such as perfect, single and sub-clusters. Its linear computational complexity make it easily applicable to a wide range of data set sizes.
- When used with K-means clustering, the SF (along with other validity indices) are limited to hyper-spheroidal clusters.

In this chapter, results of the score function are evaluated using benchmark data sets only. Additional results from the system identification methodology are given in Chapter 5. Clustering is also integrated as a tool for decision support for sensor placement (see Chapter 6).

CHAPTER

4

A New Algorithm for Feature Selection

“*The capacity of digital data storage worldwide has doubled every nine months for at least a decade, at twice the rate predicted by Moore’s Law for the growth of computing power during the same period.*” Fayyad and Uthurusamy (2002)

Overview

This chapter presents a new wrapper feature selection algorithm combining global search (PGSL) and support vector machines (SVM). After a short introduction to PGSL, a summary of SVM is given. The proposed approach is presented in detail and results on benchmark data sets are given.

4.1 Probabilistic Global Search Lausanne (PGSL)

The aim of feature selection is to find a subset of m features from a total of d that best satisfies a given criterion. For a given subset, a feature is either present or not. When finding all possible subsets m among d, Equation 4.1 gives the number of possibilities:

$$\sum_{m=0}^{d} C_d^m = \sum_{m=0}^{d} \frac{d!}{(d-m)!m!} = 2^d \tag{4.1}$$

Therefore, according to Equation 4.1, the number of possible feature combinations is combinatorial. A methodology for treating combinatorial problems involves the use of stochastic search.

PGSL (Raphael and Smith, 2003a) is a direct search algorithm that employs stochastic sampling to find the global minimum of a user defined objective function. Gradient calculations are not needed and no special characteristics of the objective functions (such as convexity) are required. PGSL performs global search through sampling the solution space using a probability density function (PDF). PGSL has three tuning parameters (for more details, see Raphael and Smith (2003a)):

- NS: number of samples (sampling cycle)
- NFC: number of loops in the focusing cycle
- NSDC: number of loops in the sub-domain cycle

At the beginning of search, a uniform PDF is assumed for the entire search space so that solutions are generated randomly. When good solutions are found, probabilities in those regions are increased so that more intense sampling is carried out in regions containing good solutions. The key assumption is that better sets of solutions are found in the neighborhood of good sets of solutions. The search space is gradually reduced so that convergence is achieved. The total number of PGSL iterations is the product of these three tuning parameters ($NS \cdot NFC \cdot NSDC$).

4.2 Support Vector Machines (SVM)

This subsection provides a summary of support vector machines (SVM). SVM are based on two concepts: the kernel trick and a separating hyperplane. The kernel trick transforms non-linear relationships from the initial space into linear relationships in order to discover relationships more easily in the feature space. A kernel is a function that evaluates the inner product between data points in some space:

$$K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y}) \tag{4.2}$$

SVM is a margin classifier that can benefit from the kernel trick. A test instance $\mathbf{z}$ is classified using the decision function (separating hyperplane) of the non-linear SVM given below:

$$y = sign\Big(\sum_{i=1}^{n} y_i \lambda_i K(\mathbf{x_i}, \mathbf{z}) + b\Big) \tag{4.3}$$

where n is the number of training samples, $y_i \in \{-1, 1\}$ is the class label of the training example $\mathbf{x_i}$, $\lambda \in \lambda_1, ..., \lambda_n$ are the Lagrange multipliers, $K(\mathbf{x_i}, \mathbf{z})$ is the chosen kernel function and b is a parameter related to the decision boundary. Training a SVM is done by minimizing the following objective function:

$$L(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \tag{4.4}$$

subject to the following constraints:

$$\sum_{i=1}^{n} \lambda_i y_i = 0 \tag{4.5}$$

$$0 \leq \lambda_i \leq C, \forall i \tag{4.6}$$

where C is a SVM tuning parameter representing the penalty for misclassifying training examples. The SVM formulation described here is for binary classification problems. Methods are available for multi-class SVM, for example in Weston and Watkins (1998). The choice of the kernel $K(\mathbf{x_i}, \mathbf{x_j})$ is important and generally depends on the application domain. The most commonly used kernel function is the Gaussian:

$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} \tag{4.7}$$

The main reason for using a Gaussian kernel is that it has only one parameter (standard deviation, σ) to tune (see Section 4.4). Furthermore, it has provided good results in several applications. Other types of kernel have been examined in the literature and new kernels can be created.

4.3 PGSL and SVM for feature selection

As mentioned in Section 2.4.1, a wrapper approach is characterized by four aspects. In the proposed algorithm, they are as follows:

- A state space of size: 2^d, where d is the total number of features
- An initial state: the initial seed in PGSL
- A termination condition: PGSL maximum number of iterations
- A search algorithm: PGSL

The PGSL-SVM methodology combines global search (PGSL) with support vector machine (SVM). The strategy is founded on the proposition that feature selection and classification stages should be optimized together and not separately. Figure 4.1 shows the flowchart for the overall wrapper feature selection procedure.

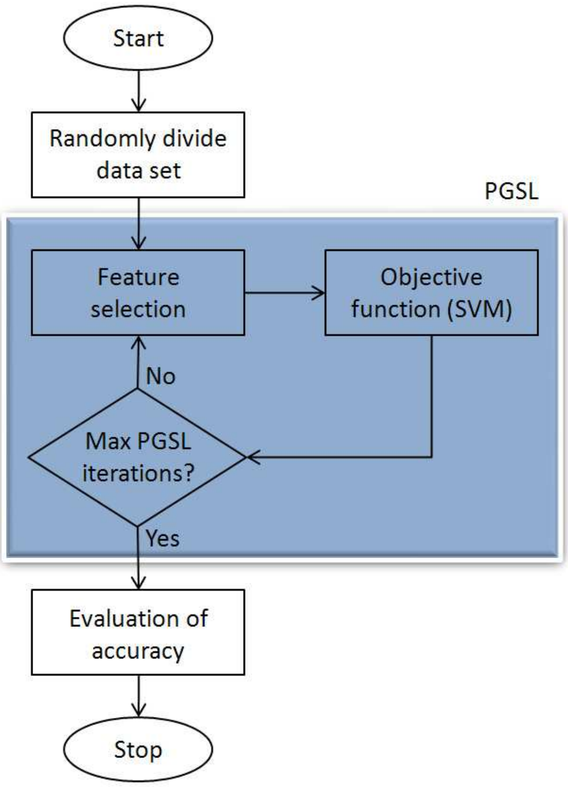


Figure 4.1: Flowchart of the feature selection process. The data set is first divided in training and test set. PGSL uses SVM with 10-fold cross-validation as the objective function. The mean error rate of the 10-fold cross-validation is given back to PGSL for the next iteration. Once the total number of PGSL iterations is achieved, the test set is used to evaluate the classification accuracy.

First, one third of the data set is randomly taken to be the testing set (*Randomly divide data set* step). To avoid any over-fitting bias within results (Reunanen, 2003), this test set is only used once, at the end of the process (*Evaluation of accuracy* step). PGSL is started with a random initial vector of dimension $d+2$ (*Feature selection* step). The first d values are rounded to either 1 or 0 (since PGSL uses continuous variables), respectively representing selected and non-selected features. The last two values are tuning parameters C and σ for Gaussian kernels in SVM.

The objective function that is minimized by PGSL is the classification error rate of the SVM. In the *Objective function (SVM)* step, a SVM with 10-fold cross-validation is run (see Section 2.4). The mean of the 10 obtained error rates is given back to PGSL as the value of the objective function to minimize. If the total number of PGSL iterations[1] is not achieved, the loop continues. Otherwise, the feature subset corresponding to the minimum error is returned by PGSL. These features are selected from both the training and test sets to respectively train and evaluate the final accuracy of the SVM (*Evaluation of accuracy* step). The result of this overall procedure (Figure 4.1) is averaged over five separate runs.

The generalization accuracy is not the only relevant criterion for evaluating a feature selection strategy. Other factors are also of importance. The number of calls to the objective function is crucial for comparison, since estimating the generalization error using 10-fold cross-validation is expensive in terms of computational time (for each PGSL iteration, 10 SVM are trained). This is a good estimator of the computational complexity of the feature selection process. Since it is not related to a specific computer, the values for different wrapper approaches are easily comparable. Therefore, while the accuracy and the number of selected features are observed (see the next section), the number of calls to SVM is fixed to be nearly the same (see Table 4.3). The number of features selected is of importance, as well. The fewer the number of features, the smaller is the amount of memory/time needed for the classification algorithm. In addition, as stated in Section 1, a small number of features helps in understanding the data.

It is noted that, due to the size of the solution space (2^d), several feature subsets may give good results for SVM. It is thus impossible to guarantee that the subset found is the best (the one that give the best SVM results) since the solution space is enormous. The aim is to find one solution among the set of good feature subsets. Thus, no study is made on the stability of the feature selection process. A feature selection algorithm that is stable is simply more *deterministic* than another one, however it does not mean that it performs better or it has found the best solution. The proposed approach is compared with random and GA-based feature selection to show its efficiency.

[1] The total number of PGSL iterations is the product of its three tuning parameters: NS, NFC, NSDC (see Section 4.1)

4.4 Results

The PGSL-SVM approach is compared with results from the literature on several UCI data sets (Merz and Murphy, 1996). Data sets have been chosen on the basis of their low number of missing values and their numeric features. Entries containing missing values have been discarded in the data preparation step to avoid issues related to missing data. Data sets are standardized with a zero mean and unit standard deviation. First, a comparative study is made. Detailed results are given afterward.

4.4.1 Comparative Results

As noted in Yang and Honavar (1998), it is usually not feasible to do a completely fair comparison of different techniques for feature selection. While certain papers use a separate test set, others use a 10-fold or 5-fold cross-validation to estimate the generalization error. In addition, values obtained are not always averaged over the same number of runs. For this reason, this study is aimed only at estimating the competitiveness of the PGSL-SVM approach.

In this experiment, PGSL tuning parameters are set following indications in the original paper by Raphael and Smith (2003a) and after some experimental testing. Values are fixed as follows: $NS = 2$, $NFC = 2 \cdot d$ and $NSDC = 2$, where d is the total number of features. SVM tuning parameters are fixed using PGSL ($C \in [0, 100]$, $\sigma \in [0, 10]$). A 10-fold cross-validation procedure is used to estimate the generalization ability. Table 4.1 shows the mean values of the cross-validation accuracy on 5 different runs.

Table 4.1 shows that the PGSL-SVM results are similar to other results in the literature. In comparison with other techniques, PGSL-SVM is better for several data sets while it is within the range of other results for a few data sets. It is thus competitive with existing feature selection techniques. An example of the difficulty of fair comparison in the literature is shown through the Ionosphere data set. Bradley and Fayyad (1998) used a separate test set, which is not the case of GA-ANN, for example, and PGSL-SVM at this stage (i.e. the *Randomly divide data set* step of Figure 4.1 is not yet done). For more details about training and test sets, see Section 2.1.

4.4.2 Detailed Results

Studies have shown that cross-validation strategies are subject to over-fitting (Kohavi and Sommerfield, 1995; Reunanen, 2003). Due to the high number of possible feature subsets, a feature subset may be found that is better than others on only these particular cross-validation folds. Therefore, an additional subset that has never been used for the feature selection process, is used as the test set (*Evaluation of accuracy* step).

Data sets	**GA-ANN (a)**	**FSV (b)**	**SVM (b)**	**HGA-1 (c)**	**SA-ES (d)**	**GA-ES (d)**	**FS-SFS (e)**	**RFE-SVM (f)**	**HGA-SVM (f)**	**PGSL-SVM**
Cleveland (13)	-	80.9	84.6	-	-	-	-	-	-	84.4
Cancer wisconsin (9)	98.0	-	-	-	-	-	-	-	-	96.8
Ionosphere (34)	94.5	84.1	86.1	95.4	92.0	89.4	92.0	91.7	92.8	96.1
Hepatitis (19)	85.2	-	-	-	-	-	-	-	-	81.0
Glass (9)	69.3	-	-	-	76.6	76.7	-	62.6	65.5	67.1

Table 4.1: Comparative study of 9 feature selection methods with PGSL-SVM on 5 data sets from the UCI Repository (Merz and Murphy, 1996). Numbers are percentage of correct classification. Compared results are taken from (a) Yang and Honavar (1998), (b) Bradley and Fayyad (1998), (c) Oh et al. (2004), (d) Loughrey and Cunningham (2005), (e) Liu and Zheng (2006) and (f) Huang et al. (2007).

Often, tuning parameters of SVM are manually set to a particular value as in Mao (2004). In this study, tuning parameters are set using a strategy depending on the method used. Four different methods are compared:

- **SVM**: Support vector machine without feature selection. SVM tuning parameters are chosen through a grid search ($C \in \{1, 10, 100\}$, $\sigma \in \{0.1, 1, 10\}$).
- **RAND-SVM**: Random selection of parameters. The solution space is sampled randomly to find subset of parameters. SVM tuning parameters are fixed to be the same as above.
- **GA-SVM**: GA feature selection combined with SVM. SVM tuning parameters are fixed according to Fröhlich et al. (2003). GA tuning parameters are based on the work by Yang and Honavar (1998). Probabilities of crossover and mutation are 0.6 and 0.001 respectively. Population size and number of generations are fixed, for each data set, so that the total number of GA iterations is the closest to PGSL (see Table 4.3).
- **PGSL-SVM**: PGSL feature selection combined with SVM. SVM tuning parameters are fixed by adding them to the PGSL search space ($C \in [0, 100]$, $\sigma \in [0, 10]$). The total number of iterations is dependent on the total number of features in the data set and is given in Table 4.3.

In this subsection, the methodology described in subsection 4.3 is applied. For large data sets (more than 200 samples), one third of the data is used as the test set (*Evaluation of accuracy* step). For data sets with less than 200 samples, no separate test set is used. In these cases, the *Evaluation of accuracy* step is done with a 10-fold cross-validation. Results of the four methods are given in Table 4.2.

First, it is visible that both GA-SVM and PGSL-SVM give better results than RAND-SVM. This shows that both strategies are effective. Regarding GA-SVM and PGSL-SVM, improvement with feature selection over standard SVM is visible for several data sets. *WDBC*, *Cancer wisconsin* and *Hungarian* show no improvement using feature selection. This is due to their small initial number of features and their importance for classification. *Cleveland* is the only data set to clearly perform worse with feature selection. This may be due to the fact that every feature is important in explaining the different classes. Therefore, removing even one of them significantly reduces classification accuracy.

Valuable improvements in classification accuracy are observed on several data sets. On the *Ionosphere* data set, GA-SVM and PGSL-SVM are better than SVM by 10.2% and 8.2% respectively. On *Zoo*, improvement are of 15.9% and 19.9%. Finally, the best improvement are shown for the *Hepatitis* data set, with an accuracy increase of 36.8% and 38.9%.

Regarding GA-SVM and PGSL-SVM, it is noted that their classification accuracy is nearly the same on average. PGSL-SVM performs marginally better on 6 data sets out of 11. This indicates that both strategies are equivalent in their generalization ability. A more interesting result is the mean number of features selected. For 8 data sets out of 11, PGSL-SVM finds sets with less number of features than GA-SVM, for the same order of accuracy. This is due to the fact that SVM tuning parameters are better fixed through PGSL-SVM. On *WDBC*, GA-SVM and PGSL-SVM find respectively 16.2 and 13.0 features for a difference of 1% in classification accuracy. PGSL-SVM has thus an improvement of 19.8% in the number of features. On *Lung cancer* and *Sonar*, the improvement is of 10.9% and 19.3% respectively.

The PGSL-SVM feature selection has two advantages over GA-SVM. First, with GA, the SVM tuning parameters have to be coded to match the usual binary format. This is not needed in the case of PGSL which uses continuous values. Second, PGSL has less tuning parameters to fix than GA. While GA has at least four tuning parameters, PGSL has a simple guideline concerning three variables. The main limitation of the proposed methodology, as with every wrapper-based approach, is the time consuming process of the classification algorithm evaluation. This time is further increased with standard cross-validation strategies.

The speed of convergence of GA and PGSL is dependent on the way their tuning parameters are fixed. Convergence studies on PGSL have been carried out in Raphael and Smith (2003b). For a fair comparison between GA and PGSL search strategies, it is ensured that the number

	SVM		RAND-SVM		GA-SVM				PGSL-SVM			
	Accuracy		Accuracy		Accuracy		# features		Accuracy		# features	
Data sets	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
WDBC (30)	97.9	1.3	84.8	6.0	97.5	1.2	16.2	1.3	96.5	1.1	13.0	1.2
Cleveland (13)	86.5	2.5	57.6	2.7	84.1	2.5	8.4	0.9	80.8	2.6	7.0	2.1
Cancer wisconsin (9)	96.5	1.7	96.4	1.2	96.2	0.8	5.6	1.1	95.6	0.9	4.8	1.1
Ionosphere (34)	84.0	1.0	66.6	17.0	92.6	1.0	17.6	2.4	90.9	1.3	16.4	3.6
Wine (13)	93.5	0.5	92.3	3.3	98.2	0.5	8.2	1.3	98.7	0.4	7.8	1.5
Hepatitis (19)	56.5	0.6	63.5	3.1	77.3	3.2	8.6	1.1	78.5	2.7	6.4	1.7
Glass (9)	59.7	5.1	31.1	4.3	62.3	5.2	4.8	1.1	61.4	4.5	5.4	1.1
Hungarian (13)	81.2	1.7	70.3	7.6	78.1	3.1	5.8	0.8	79.8	4.9	6.4	0.6
Sonar (60)	77.1	7.2	43.5	22.6	81.5	4.3	30.0	2.5	83.2	7.9	24.2	2.5
Zoo (16)	81.4	1.5	88.2	6.3	94.3	2.0	7.8	1.5	97.6	0.6	8.8	0.8
Lung cancer (57)	73.2	5.9	77.3	5.5	88.3	3.4	25.6	3.1	89.2	5.2	22.8	4.8

Table 4.2: Comparison of accuracy for SVM, random feature selection (RAND-SVM), GA-based feature selection (GA-SVM) and PGSL-based feature selection (PGSL-SVM). Numbers in brackets are dimensionality of data sets. For each strategy, 5 independent runs are made. *Mean* is the average and *Std* the standard deviation over the 5 runs. For RAND-SVM, the number of selected features is not given since it is random, and thus gives no useful information.

Data sets	**Size**	**GA-SVM**	**PGSL-SVM**
WDBC	569	240	240
Cleveland	297	110	104
Cancer wisconsin	683	72	72
Ionosphere	351	272	272
Wine	178	110	104
Hepatitis	80	156	152
Glass	214	72	72
Hungarian	294	110	104
Sonar	208	506	480
Zoo	101	132	128
Lung cancer	27	462	448

Table 4.3: Number of GA and PGSL iterations during the search for the best feature subset. Values are the number of calls to the 10-fold cross validation procedure. The best and the mean values of Table 4.2 corresponds to 5 different runs. The second column is the number of samples for each data set.

of calls to SVM is almost the same. Detail of the number of calls to the 10-fold cross-validation procedure using SVM is given in Table 4.3.

4.5 Conclusions

A new feature selection algorithm based on the wrapper concept is proposed in this chapter. This leads to an efficient feature selection algorithm. The following conclusions come out of this chapter:

- PGSL-SVM is an efficient feature selection strategy. It performs better than GA-SVM for feature selection on various data sets.
- The PGSL-SVM strategy finds subsets with a smaller number features than GA-SVM for the same order of accuracy and in the same amount of time.
- The strategy involving PGSL is easier to use since it has less tuning parameters than GA-based strategy. This number is of importance since bad tuning can lead to poor results.
- The fact that PGSL use continuous values helps fix the tuning parameters of SVM during the feature selection process.

In this chapter, the feature selection algorithm is tested on benchmark data sets. More real-case tests are performed in Chapter 5.

Chapter

5

Decision Support for System Identification

"*The bridge seemed to be among the things that last forever; it was unthinkable that it should break.*" (Wilder, 2004)

Overview

This chapter addresses the research question "*To what extent can data mining techniques support engineers during system identification tasks?*". A system identification methodology that utilizes multiple models is presented. The application of data mining techniques such as correlation, principal component analysis, clustering and feature selection to the task of extracting knowledge from multiple models are evaluated using a laboratory structure and a full scale bridge.

5.1 The Need for Multiple Models

Traditionally, system identification is treated as an optimization problem where the difference between model predictions and measurements is minimized. Values of model parameters for which model predictions best match measured data are determined by this approach. However, this approach is not reliable due to the abductive nature of the task and several types of errors (see Section 2.5). These two factors lead to the strategy of filtering multiple models. To further explain this issue, a practical example is presented. The multiple model strategy is demonstrated with a simple truss example. The structure is made of ten bars each with a cross-sectional area of $16cm^2$. Figure 5.1 shows the truss. Only the displacement at location A is measured using a sensor. The structure has a vertical displacement of 10.5 mm at position A when subject to a vertical load F of 40 kN at the same position. The objective is to detect damage in the truss. Three distinct candidate models are given in Table 5.1.

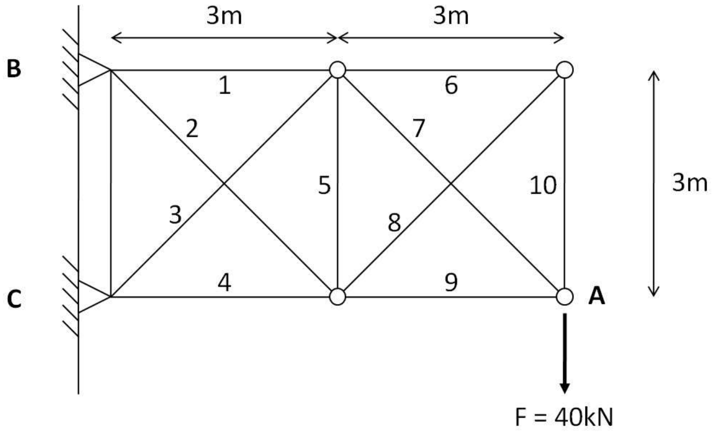


Figure 5.1: Schema of the truss structure used to justify the strategy of a multiple model approach for system identification.

Case	Damage scenario	Description	Displacement
Model 1	Element 2 damaged	87% area reduction	10.3 mm
Model 2	Element 6 and 7 damaged	69% area reduction	10.1 mm
Model 3	Support B damaged	displacement	11.0 mm

Table 5.1: Details of three models that can explain the behavior of the truss structure shown in Figure 5.1. For each model, the damaged element(s) and the modified area(s) are given. All other elements have an area of $16cm^2$.

All of them have predictions at A that lie within 5% of measurement (at point A) and will be part of a candidate model set for this identification problem. The uncertainty in identifying the model that correctly represents the structure is due to modeling and measurement errors and lack of sufficient measurements. Adding more sensors such as strain gauges on certain members can filter out some models from the candidate model set. However, minimizing the difference between errors and measurements in order to select a single model can lead to the wrong model.

Consequently, multiple models are needed for reliable system identification. When many models are presented, engineers might find it difficult to interpret the results (Section 5.3.2 and 5.3.8). Data mining is used to address this difficulty (see Section 5.3). The concept of multiple models also affects measurement system design since sensor placement is undertaken accounting for several models instead of one. This issue is developed in Chapter 6.

5.2 System Identification Methodology

5.2.1 Overall Methodology

The detailed methodology for the system identification process is given in Figure 5.2. This chapter focuses on the *knowledge extraction* part. Modules for *initial or single measurement cycle* and *subsequent measurement cycles* are described in Chapter 6. The *model creation* and *identification of candidate models* parts have been developed by Robert-Nicoud et al. (2005c). Techniques used in the *knowledge extraction* module have been introduced in Chapter 3 and 4. In this module, data mining methods are applied to the data set containing model parameters in order to obtain useful knowledge for engineers performing system identification.

The methodology is structured as follows:

Structural assumptions: Modeling assumptions define the parameters for the identification problem. The set of model parameters may consist of quantities such as elastic modulus, connection stiffness and moment of inertia. Each set of values for the model parameters corresponds to a model of the structure.

Model creation (compositional modeling): Compositional modeling is a framework for constructing adequate device models by composing model fragments selected from a model fragment library (see Section 2.5.2). Model fragments partially describe components and physical phenomena. A complete model is created by combining a set of fragments that are compatible. For modeling the behavior of structures, fragments represent support conditions, material properties, geometric properties, nodes, elements and loading. Assumptions are explicitly represented in model fragments so that the model composition module generates only valid models that are compatible with the assumptions chosen by users.

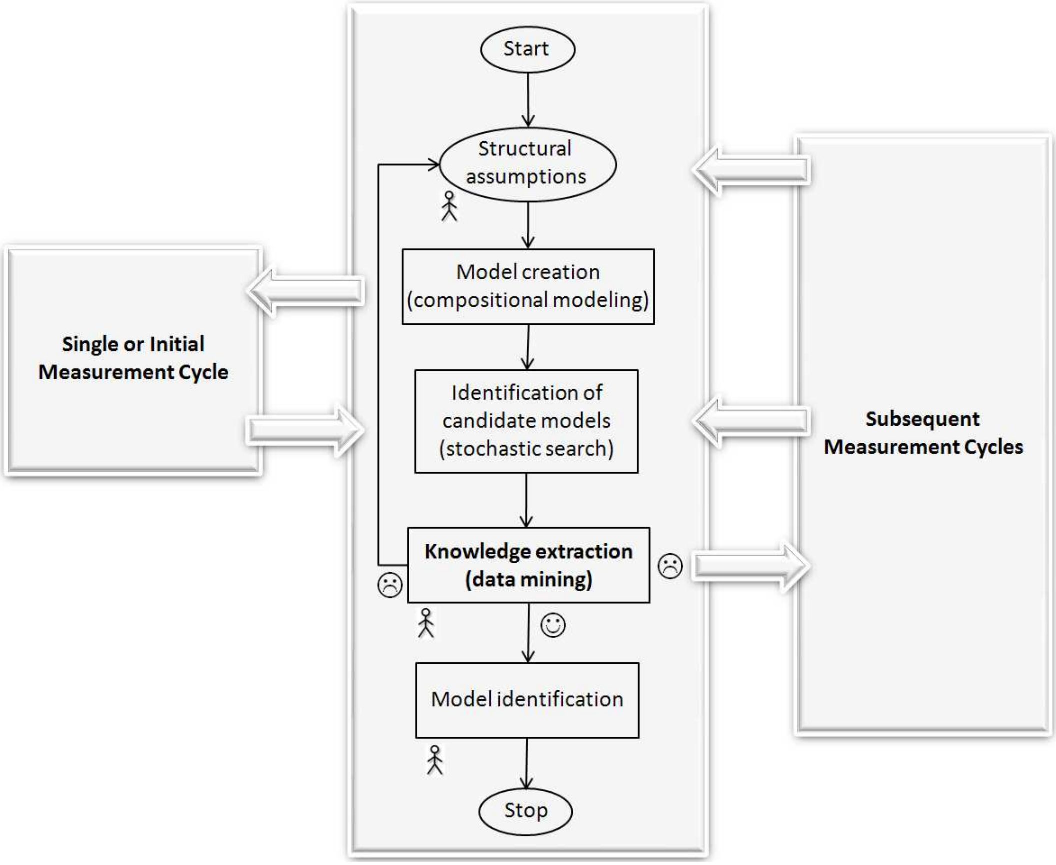

Start
Structural assumptions
Model creation (compositional modeling)
Identification of candidate models (stochastic search)
Knowledge extraction (data mining)
Model identification
Stop
Single or Initial Measurement Cycle
Subsequent Measurement Cycles
☹
☹
☺

Figure 5.2: Detailed methodology for the system identification process. This thesis focus on three modules: *knowledge extraction*, *single or initial measurement cycle* (Section 6.1) and *subsequent measurement cycles* (Section 6.2). The stick person indicates where human-computer interaction is needed, and the faces indicate whether or not the user finds the results to be satisfactory.

Identification of candidate models (stochastic search): The next step identifies - using stochastic search - a set of candidate models that may represent the real state of the structure. Measurements, a set of model parameters and an objective function (Equation 2.6) that evaluates models are needed to generate the set of candidate models. The search in the space of possible models is done through the PGSL algorithm (see Section 4.1). PGSL is used to minimize the cost function that evaluates the difference between measurements and model predictions.

Knowledge extraction: Data mining techniques such as Pearson's correlation and principal component analysis (PCA) are applied to the models for knowledge extraction. The aim is to help engineers understand the model parameters, their relationships and more generally the model space. In addition to these two techniques, clustering is used to group models into clusters. Models are grouped into clusters to i) facilitate visualization of the model space and ii) reduce the number of models given to engineers (the centroid of the cluster is a possible way of defining the cluster). The number of clusters is estimated through the score function. Visualization of clusters is improved through the use of principal components. It is thus supporting decisions of engineers regarding system identification. A feature selection algorithm is used to select relevant parameters that explain candidate models. This information is used by engineers for subsequent decisions in the system identification process. Details about these techniques are given in Chapter 3 and 4.

At this point, three situations may happen. If engineers have obtained enough knowledge from the different data mining techniques, the next step is *model identification*. The second and third situations occur when engineers are not satisfied with the available information. In the second situation, they modify assumptions (see the *structural assumptions* step). The third situation involves adding more sensors to obtain additional information. In this case, the next step is *subsequent measurement cycles*.

Model identification: In this final step, engineers - with the knowledge obtained and after possible iterations - identify the state of the system.

In this chapter, four data mining methods are used to extract information from models. First, the presence of relationships between parameters are examined using the correlation measurement (Section 5.3.4). Second, principal components analysis (PCA) is used to check whether there are parameters that are independent from others (Section 5.3.5). Clustering is used to group models into classes. The clustering process is outlined in Section 5.2.2 while results are given in Section 5.3.6. Finally, a feature selection algorithm is used to reveal important parameters that explain candidate models (Section 5.3.9). These methods are applied to models with two illustrative case studies (Section 5.3.1 and 5.3.7). Sections 5.3.2 and 5.3.8 empirically show the need of data mining techniques to interpret multiple models. In the case of correlation, PCA and clustering, only candidate models are used. Since the feature selection algorithm belongs to the supervised learning techniques (see Section 2.1), it needs both candidate and non-candidate models.

5.2.2 Clustering Algorithm

The methodology for grouping models into clusters combines PCA and K-means in order to improve visualization of results. After standardization, the PCA procedure (see Section 2.2.2) is applied to candidate models. Using all the principal components, the complete set of models is transformed into the feature space. The number of clusters is estimated using the score function (see Section 3.2). The K-means algorithm (see Section 2.3.1) is applied to the data in the feature space. Table 5.2 presents the pseudo-code of the methodology used.

Procedure for grouping models into clusters

```
Standardize the data (Section 2.2.2)
Transform the data using PCA (Section 2.2.2)
Loop $i$ from $k_{min}$ to $k_{max}$
    Run K-means $T$ times with $i$ clusters
    Evaluate K-means results using the score function (Section 3.3.1)
End
Select results with maximum value for the score function
```

Table 5.2: Pseudo-code of the clustering procedure combining PCA and K-means to separate models into clusters. k_{min} and k_{max} are respectively the lower and upper bound for the number of clusters and T is the number of times K-means is run.

The procedure to determine the best number of clusters is to evaluate the score function value for different number of clusters from k_{min} to k_{max}. The randomness of K-means, through its starting centroids, has to be taken into consideration. For this, the algorithm is run T times and the maximum value for the score function is chosen. More details can be found in Chapter 3.

5.3 Results

5.3.1 First Illustrative Case Study

A timber beam supported by springs is used to illustrate three data mining techniques from the *knowledge extraction* module. It is emphasized that even though this study focuses on a beam structure, it can be applied to other structures in other domains. After candidate models are selected using the *identification of candidate models* module, a set of parameters is chosen. In the present study, every model has the same set of parameters (see Section 7.4.5 for further details). The beam structure and the parameters are shown in Figure 5.3. Table 5.3 shows the names, descriptions and units of parameters that are contained in candidate models of the structure.

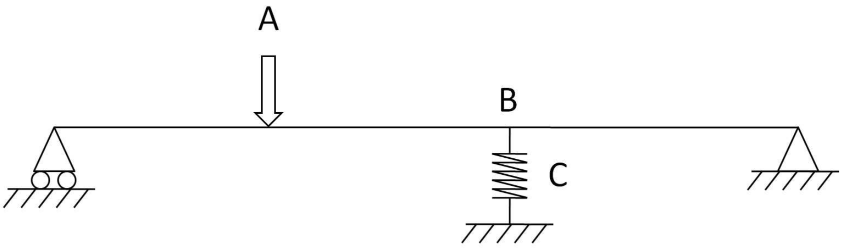


Figure 5.3: Schema of the timber beam structure. The load A is represented by its position from the left part of the beam and its intensity. B and C are two elements with particular properties (see Table 5.3).

Parameters	**Descriptions**	**Units**
p_1	Load position (A)	Node number
p_2	Load magnitude (A)	kN
p_3	Area of element C	m^2
p_4	Moment of inertia of element C	mm^4
p_5	E of element C	N/mm^2
p_6	Moment of inertia of element B	mm^4
p_7	E of element B	N/mm^2

Table 5.3: Names, descriptions and units of parameters (see Figure 5.3 for more details).

Only candidate models ($\varepsilon < \tau$)[1] are taken into account. This results in a $m \times p$ matrix M, where m is the number of models and p is the number of input parameters. For the next two data mining methods, a data set containing 3200 models is used (*identification of candidate models* step). Among these models, 300 are fixed to be candidate models. In this case, the threshold for candidate models (see Section 2.5.2) is taken to be $3.04 \cdot 10^{-4}$ (τ in Equation 2.6). As explained in Section 2.5.1 errors are as follows:

[1] For details, see Section 2.5.2.

- Measurement error (e_{meas}): difference between real and measured quantities in a single measurement.

- Modeling error (e_{mod}): difference between the prediction of a given model and that of the model which accurately represents the real behavior. This error is divided in:

  - e_1: error due to the discrepancy between the behavior of the mathematical model and that of the real structure. This is usually fixed to zero, since its variation is impossible to define for a range of practical situations.
  - e_2: introduced during the numerical computation of the solution of the partial differential equations representing the mathematical model. Fixed by engineers.
  - e_3: error due to the assumptions that are made during the simulation of the numerical model. Due to the sampling of multiple models according to parameters that cause this error, this value is fixed to zero.

5.3.2 Interpreting Multiple Models

In traditional system identification, engineers work with a single model and then declare that certain parameters are variables that require calibration with measurements. Through model updating techniques, they aim to assign model parameters in order for model predictions to match measurements. As explained in Section 2.5.2, this way of proceeding does not take into account both the abductive nature of the system identification task and compensating errors in the modeling and measurement processes. Using a multiple model strategy modifies the way engineers perform system identification. Indeed, instead of working with a single model, engineers are confronted with finding the best model among hundreds or thousands of candidate models.

The first question is: what is the best way to display a set of models? When working with a single model, engineers can simply work with an array containing two columns: the name of each parameter and its value. There is no need to visually represent the model in a solution space. When multiple models are possible, the solution space can be represented by a matrix where each row is a model and each column a parameter. It is obvious that engineers cannot manually go through hundreds of lines to study each model itself. Plotting these models is not straightforward since they are in a multidimensional space. To illustrate this issue, a set of models is plotted according to two parameters (Figure 5.4).

In Figure 5.4, each point is a model that is only represented by two parameters. In this example, these models have seven dimensions. Even visually, manual inspection is not feasible. Both the single model tradition and multidimensionality are two issues that make the task

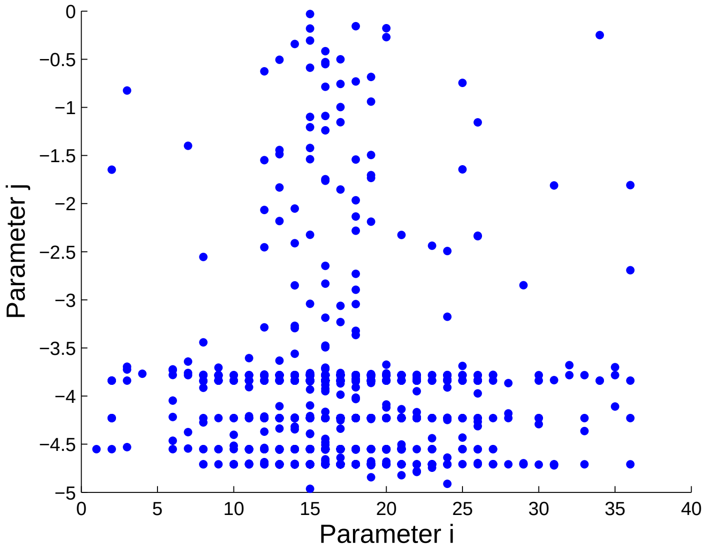


Figure 5.4: Plot illustrating the difficulty of interpreting multiple models. Models are plotted according to two out of seven possible parameters. Since each point is a different model, it is infeasible for engineers to interpret all of them manually.

of multiple model system identification very challenging for engineers. Providing 1000 model possibilities to engineers is useless if they have no way to interpret these models. Without data interpretation techniques, generating multiple models for the system identification task, although theoretically correct, is of no use. Therefore, data interpretation techniques, such as data mining, are essential for interpreting multiple model system identification results.

5.3.3 Type of Knowledge Extracted

Section 5.3.2 has shown the need for data interpretation in multiple model system identification. Data interpretation can be done with the use of data mining techniques (see Chapter 3 and 4). Above all, one has to consider the kinds of information that engineers may find useful. This is dependent on the task they perform. System identification is an example of such a task. In this work, several pieces of information can be useful to engineers, for example:

- How are model parameters related?
- Are some model parameters independent of others?
- Are there groups (clusters) of models?
- If clusters of models are present, how many are they?
- How can these clusters be visualized by engineers?
- What makes a model candidate?
- Are there parameters that have values which can help discriminate between candidate and non-candidate models?

These questions are expressed in different types of knowledge. Table 5.4 summarizes the type of knowledge that can be extracted, the type of models concerned and the technique used.

The third column corresponds to the data mining technique that has been chosen to extract the particular type of knowledge. Although other techniques can be used, data mining techniques used have been chosen on the basis of their efficiency in extracting different types of knowledge. Explanations related to the choice of the data mining techniques are given below.

When measuring independence between variables and linear relationships, several methodologies are possible, for example: covariance, correlation, mutual information and PCA. The Pearson's correlation measure is an extension of the simple covariance formula. The usual unit of measurement of the mutual information is the bit, which makes it not easily readable by engineers. PCA is not limited to comparison between two variables, as is correlation. Other

Type of knowledge extracted	**Type of models**	**Technique used**
Linearly independent parameters	Candidate	Pearson's correlation PCA
Linear relationships between parameters	Candidate	Pearson's correlation PCA
Number of clusters of models	Candidate	K-means clustering Score function
Visual representation of clusters	Candidate	K-means clustering PCA
Features explaining candidate models	Candidate and non-candidate	PGSL SVM

Table 5.4: Type of knowledge extracted, type of models mined and technique used.

techniques exist such as linear discriminant analysis (LDA). However, the latter deals with discrimination between possible classes and it is not the aim at this step (since only candidate models are present). PCA is preferred since it deals with the data in its entirety without focusing on the possible underlying class structure (Martinez and Kak, 2001).

For clustering a data set, many types of algorithms are possible. They are mentioned in Section 2.3. K-means is chosen due to its efficiency, easy implementation and ease of use. K-means has been identified as one of the 10 most important algorithms in data mining (Wu et al., 2008). Fuzzy clustering is not used since it is difficult to understand results in the context of multiple model system identification. Regarding the validity index, several have been tested. Results of these tests as well as the single cluster case, have motivated development of a new score function in the scope of this thesis. Finally, PCA has been chosen to display the results. It is chosen since it is easy to map the data back in the initial space. Indeed, models in the PCA space are not directly usable by engineers. The fact that PCA allows to transform the data back in the initial space is a plus.

Regarding features explaining candidate models, simple ranking techniques cannot detect the effect of feature combinations. This is why a more complex feature selection technique is used. Among existing techniques, wrapper approaches are the most efficient. Moreover, stochastic search algorithms do not require the monotonic assumption that greedy strategies impose. It is for these reasons that PGSL is used. It is combined with support vector machine (SVM) that has been recognized to belong to the 10 most important algorithms in data mining (Wu et al., 2008).

5.3.4 Correlation

With correlation measurements, the aim is to examine how two candidate model parameters p_i and p_j are related. For achieving this, a correlation matrix, in which each element is computed as explained in Section 2.2.1, is constructed. Rows and columns of the matrix correspond to one of the seven parameters of Table 5.3. For example, element $(2, 3)$ represents the correlation between p_2 and p_3. The correlation matrix is symmetric about its diagonal since correlation is a commutative operation. Correlations between certain parameters are observed, as shown in Table 5.5.

Correlation matrix of model parameters

	p_1	p_2	p_3	p_4	p_5	p_6	p_7
p_1	1.00	0.00	0.00	-0.00	0.00	0.00	0.00
p_2	0.00	1.00	0.20	0.53	-0.13	0.83	-0.04
p_3	0.00	0.20	1.00	0.36	-0.24	0.35	-0.01
p_4	0.00	0.53	0.36	1.00	-0.05	0.47	-0.05
p_5	0.00	-0.13	-0.24	-0.05	1.00	-0.04	-0.15
p_6	0.00	0.83	0.35	0.47	-0.04	1.00	-0.12
p_7	0.00	-0.04	-0.01	-0.05	-0.15	-0.12	1.00

Table 5.5: Correlation matrix for the seven parameters of Table 5.3.

If two parameters have a high degree of correlation, for example, greater (in absolute value) than 0.5, it is assumed that there is a relationship between them. Two notable results are present in Table 5.5. Firstly, p_1 (i.e. the load position) is not correlated with any other parameters since the first column of the correlation matrix is zero (except for the first element). This means that the load position is an independent parameter for system identification. This is due to the fact that for all candidate models, the position of the load is always the same. In other words, a good match between predictions and measurements is not obtained if the load is not in the correct position. Therefore, it is concluded that the load position can be estimated reliably using the system identification methodology.

The second notable result concerns p_2, the load magnitude. There are significant correlations in the second column of the correlation matrix. This implies that the load magnitude has strong correlation with other parameters. In other words, the load magnitude cannot be estimated independently of other parameters. Different combinations of the load magnitude and other parameters could result in the same degree of match with measurements.

5.3.5 Principal component analysis (PCA)

In this subsection, PCA is used as a "weighting method" that gives us an idea of the relationships among candidate model parameters. By examining the principal components, linear combinations of parameters may be visible. Similar to the correlation measurement, the PCA is an unsupervised data mining method. After applying PCA on the data, the first three principal components are examined. Remaining components are not needed since the first three components explain about 90% of the variations in the data as shown in Figure 5.5.

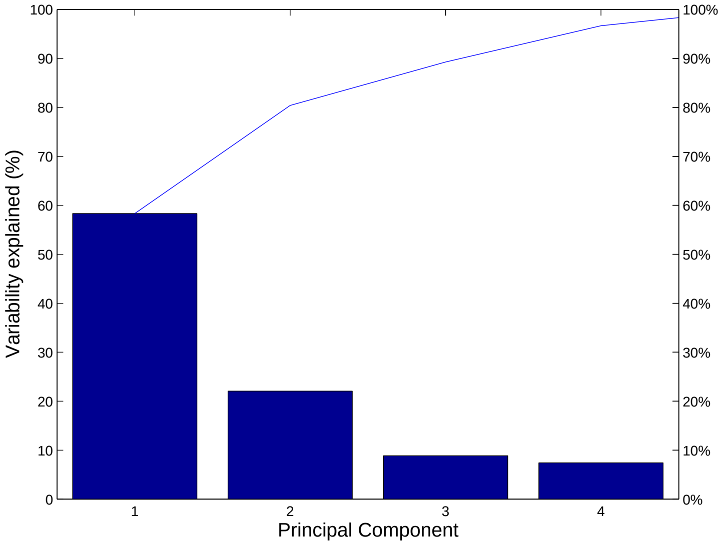


Figure 5.5: Percent variability of the data explained with the principal components. The horizontal line represents the sum of the variability explained with the principal components.

Figure 5.5 shows that about 60% of the candidate models differ in the value of the first principal component and about 20% models differ in the value of the second principal component. There is no significant variation in the values of remaining components. Instead of using the initial model parameters (see Table 5.3), new variables c_1, c_2, etc. are introduced. Using these new variables, a common characteristic of all candidate models is that the values of variables c_3, c_4, .. c_7 are nearly constant. Models differ mostly in the values of c_1 and c_2.

In this example, a few components explain most of the variations in data. For example, a two-dimensional plot (using only two principal components instead of seven parameters), shows about 80% of the total variability in the data. Showing data in two or three dimensions is straightforward, while displaying seven dimensions is an issue. When c_1 and c_2 are plotted in a two dimensional graph, a number of clusters are observed (see Figure 5.8). The clustering technique presented in Chapter 3 is applied to system identification in Section 5.3.6. The first three principal components (PC) are presented in Table 5.6.

First three PC			
	PC_1	PC_2	PC_3
w_1	0.00	0.00	0.00
w_2	-0.01	-0.01	0.00
w_3	-0.42	-0.14	-0.82
w_4	-0.55	-0.62	0.49
w_5	0.67	-0.71	-0.17
w_6	-0.27	-0.28	-0.17
w_7	-0.03	0.13	0.17

Table 5.6: The first three principal components (PC) ordered according to their ability in explaining the variability of the data. The w_i are the weighting coefficients for each initial parameter.

Each column of Table 5.6 contains the weighting coefficients (w_i) for each initial parameter within the linear equation that is used to compute a point in the new dimension space. Therefore, each coefficient w_i is a weight factor that represents the importance of the initial parameter in the new dimensional space. The first coefficient is zero for the first three principal components[2]. It means that p_1, the load position, has no influence on the variability of the data. As reported above, the load position is always the same for every candidate model. This is not the case with other parameters; they vary among candidate models.

The load position can be estimated reliably by plotting the MSE between model predictions and measurements against the load position for all the models (Figure 5.6).

In Figure 5.6, each point in the plot corresponds to a model. In this study, all generated models have a low mean square error (MSE). This is due to the nature of the study which is done on a simple laboratory structure. In this case, 300 models have been chosen as candidate models. In reality, the number of candidate models is fixed according to the threshold which is itself defined according to the structure. For more details related to the different errors, see Section 5.3.1. Details of error values are given for another case study in Section 5.3.9. All models that have a low MSE have the load on node 10. However, all models that have a load on node 10 need not have a low MSE because the values of other parameters might be wrong. A necessary although not sufficient condition for a model to be candidate is that the load is on

[2] This is also the case for other principal components.

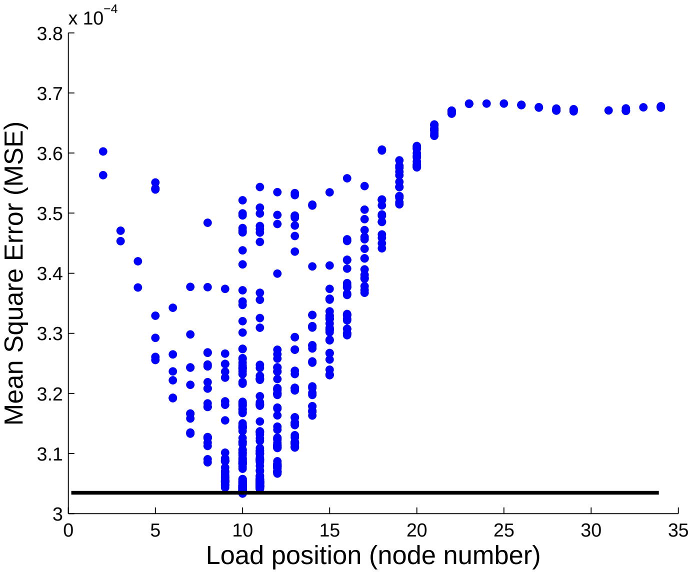


Figure 5.6: Mean square error (MSE) versus p_1, i.e. the load position. Each point on the plot is a model. The thick horizontal line shows the threshold for candidate models.

node 10. Therefore, this plot shows that identification is good with respect to the load position. This is not the case with other parameters. This is clear from the plot of the MSE versus p_3 shown in Figure 5.7.

For parameter p_3, candidate models are widely spread out. This parameter *alone* is not sufficient to estimate the reliability of system identification in this example. Variations in values of parameters of candidate models are represented by a few principal components. Since, these components are combinations of initial model parameters, it is concluded that there are relationships between model parameters which make system identification unreliable.

It is to be noted that one can visually see that the solution space has patterns. Consider the same case study with another set of parameters containing two loads instead of one. As shown in Figure 5.8, candidate models form a set of groups within the solution space. This leads to further investigation of the candidate model space using clustering techniques.

5.3.6 Clustering

The number of clusters of candidate models is useful information for engineers performing system identification. When the *identification of candidate models* step defined in Section 5.2.1 produces m candidate models, it does not mean that there are m completely different models

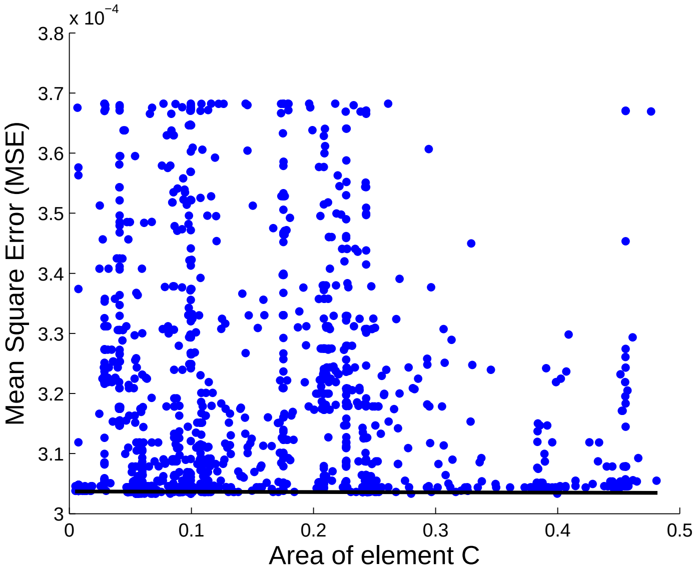


Figure 5.7: Mean square error (MSE) versus p_3, i.e. area of element C. Each point on the plot is a model. The thick horizontal line shows the threshold for candidate models.

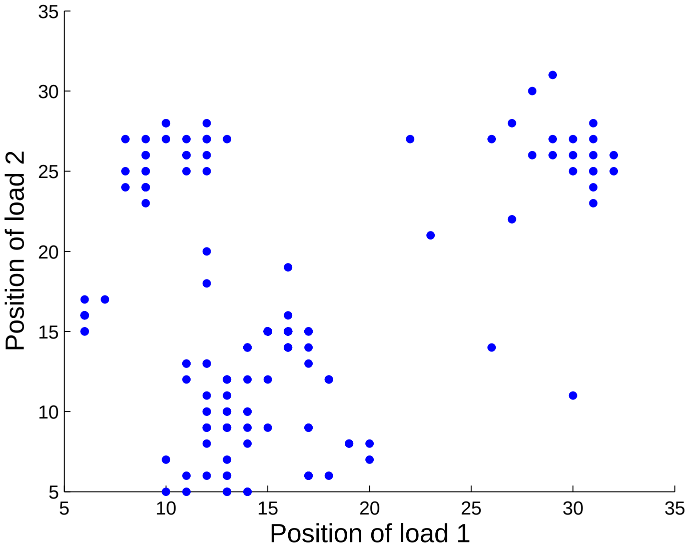


Figure 5.8: View of the model space using two parameters (position of load 1 and 2). Clusters of candidate models are visible.

of the structure. These m models might only differ slightly in a few values of parameters while representing the same state. In other situations, models might have important differences representing distinct classes which are referred to as clusters. In this problem, the number of clusters (i.e. the number of classes of models) for a data set is unknown.

Two important notions have been defined in Section 3.2: the within class distance (*wcd*) and the between class distance (*bcd*). Engineering meanings in terms of system identification need to be given to these two distances. They are both directly related to the space of models for the task of system identification using multiple models. The *wcd* represents the spread of models within one cluster. Since it gives information on the size of the cluster, a high *wcd* means that models inside the class are widely spread and that the cluster may not reflect physical similarity. The *bcd* is an estimate of the mean distance between the centers of all clusters and therefore, it provides information related to the spread of clusters. For example, a high *bcd* value means that classes are far from each other and that the system identification is not currently reliable. More details about the score function are given in Section 3.2. From a system identification point of view, *bcd* values indicate how different the k situations are. Values of *wcd* give overviews of sizes of groups of models.

The case study used below is inspired by the description given in Section 5.3.1. In this particular example, 300 models composed of seven parameters are identified. The six first parameters are position and magnitude of three loads while the last one is the stiffness of the spring (see Figure 5.3). After running the procedure described in Table 5.2, the number of clusters is chosen to be five. The results are shown in Table 5.7.

Number of clusters	**bcd**	**wcd**	**SF**
2	1.13	1.85	0.39
3	1.17	1.57	0.49
4	1.20	1.41	0.55
5	1.07	1.28	**0.56**
6	0.92	1.18	0.54
7	0.81	1.12	0.52
8	0.71	1.01	0.52

Table 5.7: Comparison of values for between class distance (*bcd*), within class distance (*wcd*) and score function (SF) for various numbers of clusters.

It can be seen that the maximum value for the score function is reached with five clusters. Values of the SF for 4 and 6 clusters are close to the maximum achieved with 5 clusters. This means that the number of clusters is not straightforward. Results can be double-checked visually as shown in Figure 5.9 bottom. This shows that the methodology needs user interaction. Engineers can interact in different ways. They can either fix the number of clusters according to the plot and the SF value, for example. They can also manually input the desired number of clusters. In this case, the number of clusters is considered to be part of the domain knowledge.

The procedure outlined in Table 5.2 is followed. To judge the improvement of the methodology with respect to the standard K-means algorithm, the two techniques are compared. Figure 5.9 shows the improvement in a visualization point of view. The top part of Figure 5.9 corresponds to standard K-means whereas the bottom part is the result of the methodology described in Section 5.2.2. It can be seen that the proposed methodology is better able to present results visually.

To conclude, the score function defined in Section 3.2 serves two purposes. First, it gives an idea of the performance of the clustering procedure. Second, it allows choice of a realistic value for the number of clusters. This number must be verified by the expert. Reducing the random effect of the procedure is achieved through several runs of the algorithm to compute the score function value.

5.3.7 Second Illustrative Case Study

To illustrate the feature selection algorithm (see Chapter 4), the Schwandbach bridge (designed by Maillart in 1933) is taken as a case study (Figure 5.10). This structure is inspected periodically and has been the subject of many verifications as codes have improved, for example Salvo (2006). The Schwandbach bridge is now a pedestrian bridge, although it could be reopened for traffic. Deflection measurements have not been carried out since the 1930s and while the bridge shows no visible evidence of deterioration, the question of taking measurements arises periodically. In Switzerland, bridges are traditionally measured for changes in deflection at mid-span during load tests. A single model (usually the design model) is used with the deflection measurement and the loading to determine values for parameters that have some uncertainty, such as the elastic modulus multiplied by the moment of inertia, $E \cdot I$. However, this bridge is too complex for such rudimentary model-calibration strategies.

While many assumptions are acceptable at the design stage for achieving safety and serviceability, they are not appropriate for interpreting measurements. For example, there is no physical hinge at the extremities of the vertical spandrel elements. These connections cannot be assumed to be fixed either since even small amounts of cracking reduce connection stiffness. Furthermore, not all connections are expected to have the same stiffness due to factors such as relative slenderness and varying locations on the structure. The Schwandbach bridge has 20 such connections. They are shown in Figure 5.11 using open circles. In this chapter, the

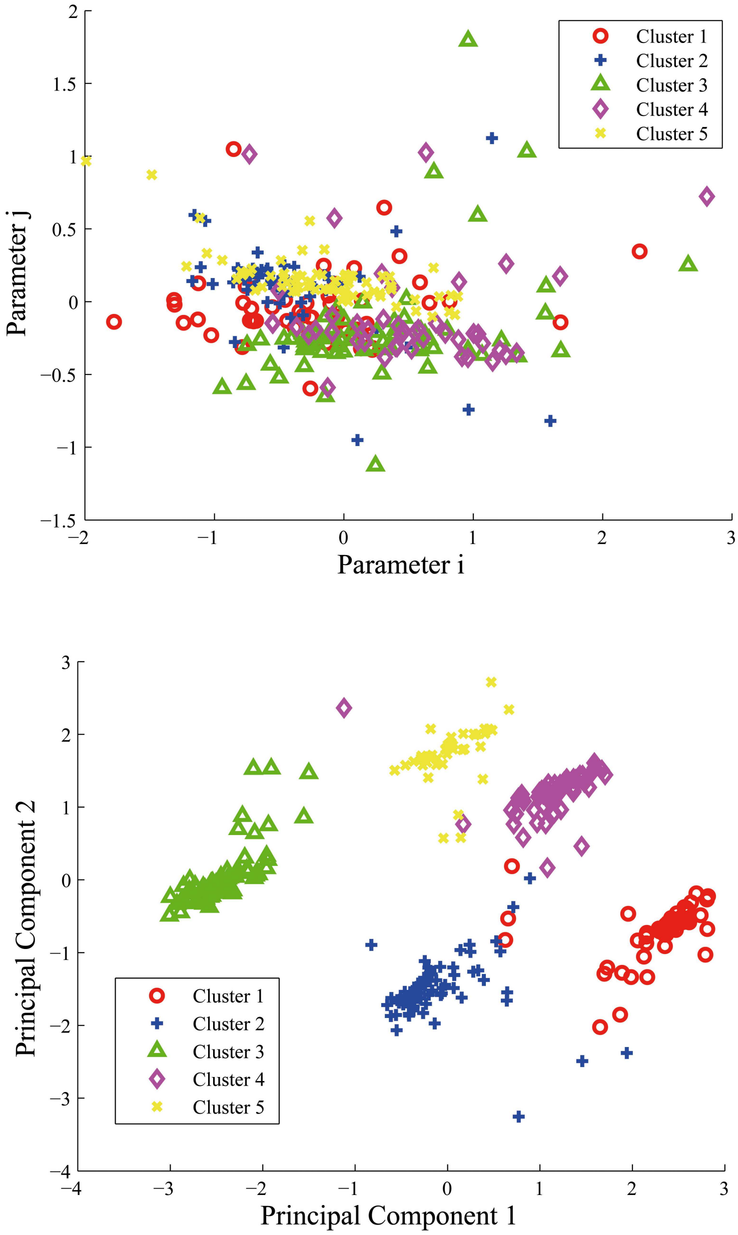


Figure 5.9: Visual comparison of standard K-means (top) with respect to the proposed methodology (bottom). Every point represents a model and belongs to one of the five possible clusters.

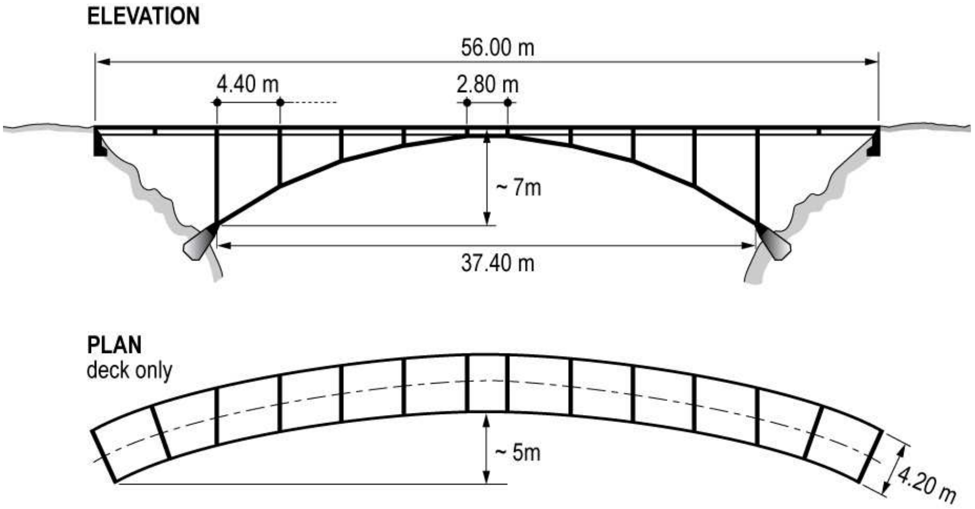


Figure 5.10: Schema of the Schwandbach bridge used to illustrate the feature selection algorithm.

knowledge extraction module (see Section 5.2) is used to select relevant model parameters (i.e. connections) that can explain why models become candidates through Equation 2.6.

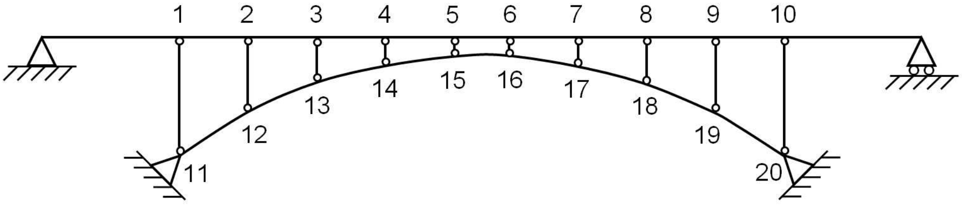


Figure 5.11: Schematic view of the bridge showing the 20 connections.

In the case of the Schwandbach bridge, the number of permutations and combinations of modeling assumptions - connection stiffnesses - results in several tens of thousands of possible models. Although this case has important technical and historical attributes, these conclusions are equally valid for most ordinary structures of moderate complexity.

Bridges are often tested periodically using static loads to check for strength degradation. The response of the bridge for trucks positioned on the bridge is measured using sensors. Engineers estimate the stiffness of the bridge from measured responses and compare those with results from previous tests. Such a scenario is simulated for the Schwandbach bridge. It is schematically represented in Figure 5.12.

For simulation, a three dimensional finite element model of the complete bridge is created.

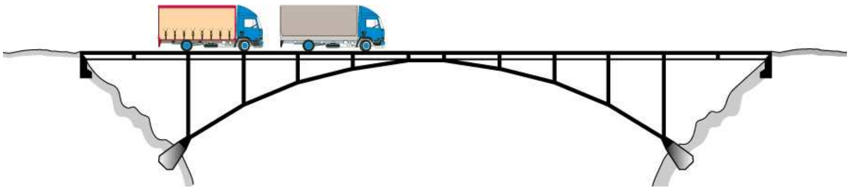


Figure 5.12: Example of the load case for the Schwandbach bridge given the scenario that the bridge is reopened for traffic.

The vertical slab-girder connections and the vertical slab-arch connections are modeled using rotational springs. A load test is simulated that involves two trucks. The details of the load test are given in Table 5.8.

Information	**Value**
Position of rear axle of left truck from left abutment	15 [m]
Distance between trucks	3.7 [m]
Distance front-rear axle	2.6 [m]
Front axle load	17 [kN]
Rear axle load	44 [kN]
Spacing between front wheels	1.8 [m]

Table 5.8: Details of the two trucks and their positions.

5.3.8 Interpreting Multiple Models (cont'd)

The case study presented in Section 5.3.7 illustrates the need for data interpretation as well. This case is much more complex than the previous one. Here, a real structure is modeled instead of a lab structure. In this particular example, 20 parameters are identified by engineers instead of 7 in the first case study. Therefore, the model space is enormous and thus very difficult to understand. It is for example difficult to understand why some models are candidates and others are not. Techniques such as PCA are only able to reduce the dimensionality by performing feature extraction[3]. Figure 5.13 shows the result of PCA on the Schwandbach bridge example.

In this example, it is observed that even 10 principal components only explain around 85% of the variability of the data (see Figure 5.5 for comparison). PCA is also limited to linearly transform data in the feature space. Thus, non-linear relationships are not taken into account.

[3] Building new features (i.e. principal components) from the initial ones.

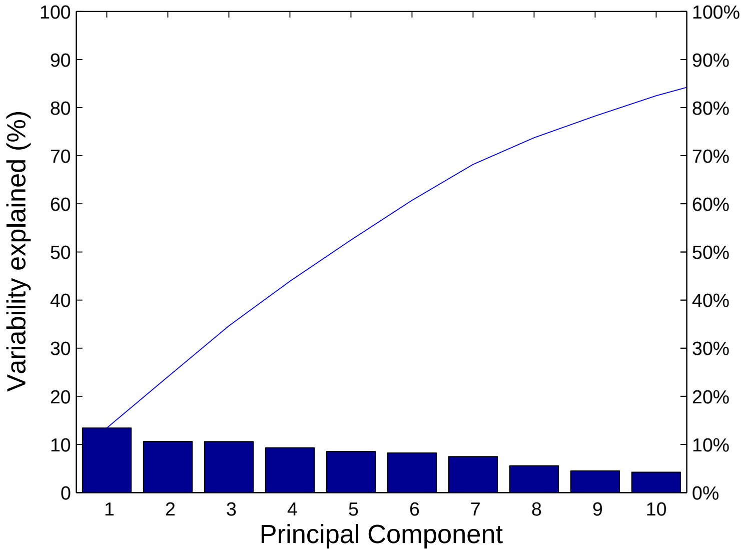


Figure 5.13: PCA results on the Schwandbach bridge example. The x-axis are the principal components while the y-axis the variability explained.

Another important point with PCA is that only candidate models are used. PCA explains the variability and linear relationships among candidate models, but gives no indication such as the importance of sets of parameters in explaining the candidate models.

It is interesting for engineers to know parameters that explain or differentiate candidate from non-candidate models. This task cannot be done manually due to the high number of models. Only data analysis techniques, such as data mining, provide support. This type of knowledge can be extracted through feature selection techniques. The aim is to select a subset of the features that best explain candidate models. This is represented by a classification task where the inputs are the parameters and the output is the class. Two values for the class are possible: candidate or non-candidate model. The aim of feature selection is to *select* the feature subset that best classify models into one of these two possible classes. The selected features give thus an idea to engineers of parameters that are important in explaining candidate models.

5.3.9 Feature Selection

When engineers are given model parameters that best separate candidate from non-candidate models, they can better understand why some models become candidates. The advantage of such knowledge is that it is easily readable by engineers. Therefore, it gives engineers a better understanding of the candidate model space. The case study introduced in Section 5.3.7 is used for illustrating feature selection.

First a set of 1000 models are generated using the methodology described in Section 5.2. There are five sensors on the structure at positions 1, 6, 10, 13 and 18 (see Figure 5.11). The threshold is defined by e_2 (see Section 5.3.1) which is $8\mu rad$. Among these models, 500 are candidate models. The starting point is thus a matrix of 1000 rows and 21 columns. The first 20 columns contain, for each model, the value of each connection stiffness. The last column corresponds to the class label. A candidate model is labeled with 1 and a non-candidate model with 2. At this point, the methodology described in Section 4.3 is run 5 times. The size of the test set is fixed as one third of the data set (33%). Results obtained are given in Table 5.9.

Feature selection results
Mean test accuracy: 97.2%
Standard deviation of test accuracy: 0.6
Mean number of features: 11.4
Standard deviation of number of features: 2.1

Table 5.9: Results obtained for feature selection on 1000 models over 5 independent runs.

First, it is observed that the standard deviation of the test accuracy is low. This means that results of the 5 different runs are close. Regarding the number of features, it is observed that around 11 connection stiffnesses are selected, in mean, out of 20. Thus, about half of the connection stiffnesses are useless in separating candidate from non-candidate models.

For this experiment, the number of PGSL iteration is set to 160 and 5 independent runs are averaged. The best test accuracy (97.9%) corresponds to the selection of the following parameters: p_2, p_4, p_7, p_9, p_{11}, p_{12}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19} and p_{20}. Therefore, with these 13 connection stiffnesses, one can argue that a model is candidate with 97.9% accuracy. These 13 parameters are shown with black dots in Figure 5.14.

This set of 13 features is of importance for engineers. It can be used to support further decisions. For example, the other 7 connection stiffnesses are not important in identifying candidate models. Variations at these positions do not help engineers for his system identification task. Since these 13 features have been selected, it means that the other 7 either contain similar information (they are redundant) or no information at all. The 13 connection stiffnesses are independent from each other since they contain no or few redundant information. This may change assumptions of engineers about the structure. Therefore, feature selection can give useful information to engineers who must decide on subsequent sensor placement and evaluate the validity of modeling assumptions. This conclusion is however related to the initial sensor

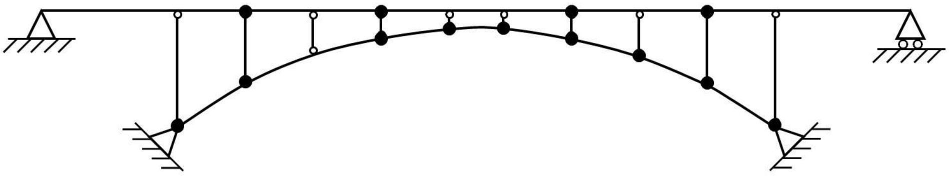


Figure 5.14: Representation of the 13 selected parameters (black dots) on the Schwandbach bridge. Sensors are at positions 1, 6, 10, 13 and 18 (see Figure 5.11).

positions. If another set of sensors is chosen, another set of features could be selected.

This issue is illustrated through the same case using different sensor locations. While in the previous example, sensors where at positions 1, 6, 10, 13 and 18, in this situation, they are at position 2, 5, 9, 11 and 20 (see Figure 5.11). Since the sensors are placed differently, information obtained is not the same. Thus, candidate models generated are different, as well (their predictions match measurements at the new sensor locations). Therefore, in the feature selection process, other parameters explaining candidate models are selected. They are shown in Figure 5.15.

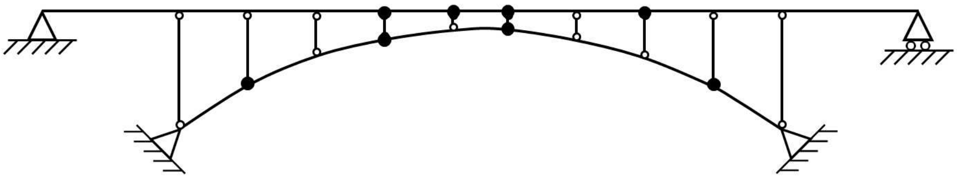


Figure 5.15: Representation of the 8 selected parameters (black dots) on the Schwandbach bridge. Sensors are at positions 2, 5, 9, 11 and 20 (see Figure 5.11).

As explained above, it is observed that the selected parameters depends on the sensor placement. In this example, less parameters are selected (8 instead of 13). The classification accuracy with these 8 parameters is 98.2%. It is noted that certain parameters are selected in both situations. It is the case of parameters p_4, p_{12}, p_{14}, p_{16} and p_{19}. Although no general conclusion can be drawn from these two situations, it is remarked that out of p_4, the recurrent parameters are all on the arch of the bridge. It is thus assumed that these parameters are the most important in explaining candidate models.

5.3.10 From Feature Selection to Feature Weighting

In Section 5.3.9, the feature selection algorithm was used to select parameters that explain candidate models. However, the only information given to engineers is whether a parameter is important or not in explaining candidate models. A step further is to show the relative importance of each parameter. For this, feature selection is generalized to feature weighting. An example in the case of genetic algorithms (GA) is given in Komosinski and Krawiec (2000).

The same procedure as described in Section 4.3 is used. However, instead of rounding the solution proposed by PGSL to a binary vector, it is rounded to two decimal numbers. For example, instead of obtaining a binary vector such as *[1 0 1 0 0 1]* which only gives important features in the case of six possible parameters, the detailed vector *[0.85 0.23 0.65 0.18 0.03 0.55]* is given. Therefore, instead of informing engineers that parameters p_1, p_3 and p_6 are important, one can rank the importance of each parameter.

The feature weighting procedure is applied to the second case study of Section 5.3.9. Table 5.10 shows the results of feature weighting on the case were initial sensors are at position 2, 5, 9, 11 and 20.

Relative importance of each parameter										
Parameters	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
Weights	0.14	0.67	0.34	0.37	0.86	0.02	1.00	0.83	0.36	0.42
Parameters	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}	p_{19}	p_{20}
Weights	0.91	0.60	0.62	0.59	0.66	0.04	0.00	0.00	0.60	0.18

Table 5.10: Parameters and their importance (weight) in explaining candidate models for the second case study of Section 5.3.9. These weights are represented graphically in Figure 5.16.

Relative importance of these parameters are given with a classification accuracy of 98.8%. First, it is noted that, for the same case study, if values are rounded, results are different from the ones observed in Figure 5.15. This is due to the fact that, during the feature subset search process, SVM are trained using weighted parameters instead of selected parameters. Thus, the SVM classification accuracy is not the same. Therefore, it guides PGSL differently (i.e. toward different solutions). It is thus important to understand that, in the case of global search in the parameter space, feature weighting is significantly different from feature selection.

Second, the results allow more specific comments regarding the importance of parameters. For example, parameters p_6, p_{16}, p_{17} and p_{18} are meaningless for explaining candidate models since their value is close or equal to zero. On the contrary, p_5, p_7, p_8 and p_{11} are the most meaningful parameters. Finally, the relative importance of parameters is noted. It is observed that parameters p_2, p_{12}, p_{13}, p_{14}, p_{15} and p_{19} are nearly the same (around 0.6). Particularly, parameters p_{12}, p_{13}, p_{14} and p_{15} show that useful information is present all along the left part of the arch.

One way to represent the results is to show the importance of each parameter on the bridge by an intensity factor. For example, colors can be used. In Figure 5.16, each parameter is devised by a color from white (weight of zero) to black (weight of one). Therefore, the darker the parameters, the better they explain candidate models.

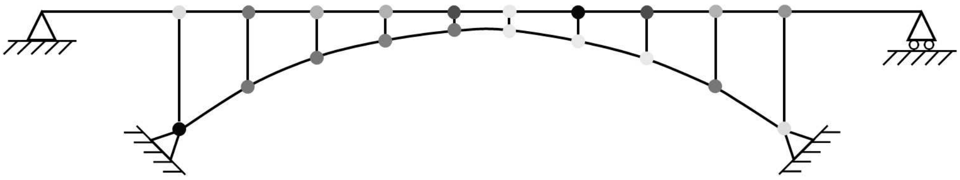


Figure 5.16: Graphical representation of the importance of each parameters in explaining candidate models. Each parameter is devised by a color from white (weight of zero) to black (weight of one). Specific values are given in Table 5.10.

It is also concluded that PGSL has an advantage over GA for feature weighting since the former uses continuous variables. PGSL is thus easily applicable to the general case of feature weighting.

5.4 Conclusions

A methodology that integrates data mining in the system identification process is evaluated. Techniques such as correlation, PCA, clustering and feature selection are used to extract knowledge that is useful for engineers. The following conclusions are of importance:

- A multiple model strategy is needed in system identification. This is demonstrated through a simple truss example.
- Data mining techniques are helpful for interpreting results leading to system identification.
- Data mining techniques such as correlation and PCA have the potential for bringing out common characteristics of the set of candidate models.
- Applied to system identification, correlation and PCA are able to i) bring out parameters of candidate models that are linearly independent and ii) improve visualization of models that have a high number of parameters.

- The use of K-means (for grouping models) and application of PCA (for displaying them) help in visualizing the solution space. This support is needed since the methodology involves the use of several models for system identification.

- Feature selection is an effective data mining process for supporting system identification since it informs engineers about parameters that are relevant in explaining candidate models.

This chapter thus successfully demonstrated how data mining can be integrated in an overall methodology for multiple model system identification.

Chapter

6 Decision Support for Sensor Placement

"*I have no problem paying an engineer for one day to place sensors and make measurements on a railway bridge. However, I have a problem paying the engineer for six months to make sense of the data.*" (A Swiss railways engineer)

Overview

Knowledge extracted through data mining is of better quality when relevant information is measured. In this chapter, two strategies for initial sensor placement are compared: greedy and global search. It is shown that the best strategy is mainly determined by the context of the measurement system. A methodology for iterative sensor placement, that integrates clustering, is proposed. Through the use of clustering and entropy, the next best sensor location is iteratively proposed to engineer. Several case studies are presented to validate the methodology.

6.1 Initial Sensor Placement

6.1.1 Sensor Placement using Entropy

In the field of model-based system identification, configuring a measurement system can be defined as finding optimal positions for sensors in order to best separate model predictions[1]. Following Robert-Nicoud et al. (2005b), the notion of entropy is used to measure the separation between predictions. The expression used to calculate entropy is the Shannon's entropy function (Shannon and Weaver, 1949) which comes from the field of information theory. Shannon's

[1] The term *predictions* will be used in place of *model predictions* for readability.

entropy function represents the disorder within a set. In the present work, a set is an ensemble of predictions for a particular system identification task. The entropy or disorder is maximum when predictions show wide dispersion.

Since the goal is to make maximum separation between models, positions with maximum prediction disorder are the most interesting. In other words, the best measurement location is the one with maximum entropy (model predictions have maximum variations). For a random variable X, the entropy $H(X)$ is given by Equation 6.1:

$$H(X) = -\sum_{i=1}^{|X|} P_i \cdot log(P_i) \tag{6.1}$$

where P_i are the probabilities of the $|X|$ different possible values of X. For practical purposes, $0 \cdot log(0)$ is taken to be zero. When a variable takes $|X|$ discrete values, the entropy is maximum when all values have the same probability $1/|X|$. Thus entropy is a measure of homogeneity in a distribution. A completely homogeneous distribution has maximum entropy.

In the present study, the entropy for a given sensor location is calculated from the histogram of predictions (see Figure 6.1). Given a set of candidate models (Raphael and Smith, 2003a; Robert-Nicoud et al., 2005b), the finite element method is used to compute predictions at all possible sensor locations. These predictions can be seen as a matrix in which each row corresponds to predictions for a model and each column is a specific sensor location. At each possible sensor location, a histogram containing predictions is built. Each bar in the histogram represents those models whose predictions lie within that interval. Note that intervals are defined by the accuracy of the measurement devices. At each iteration, the sensor location corresponding to maximum entropy of predictions is chosen. Sensors are therefore sorted in ascending order according to their efficiency in separating model predictions. For calculating the entropy, the probability P_i of an interval in the histogram has to be computed. It is given by Equation 6.2:

$$P_i = \frac{r_i}{r_{tot}} \tag{6.2}$$

where r_i is the number of predictions in the interval and r_{tot} the total number of predictions (see Figure 6.1). Therefore, for S possible sensor locations, S histograms are evaluated according to the entropy measure. The methodology for initial sensor placement is given in Figure 6.2 and described below. Details about other parts of the methodology are given in Section 5.2 and 6.2.

Model generation (stochastic sampling): In this step, a stochastic search algorithm is used to generate a set of models through sampling the model space. An initial model definition (from the *Model creation* step) is used as a starting point for sampling the model space.

Initial measurement system design: Starting with a set of models (*model generation* step), either a greedy or a global search algorithm is used to place sensors according to generated

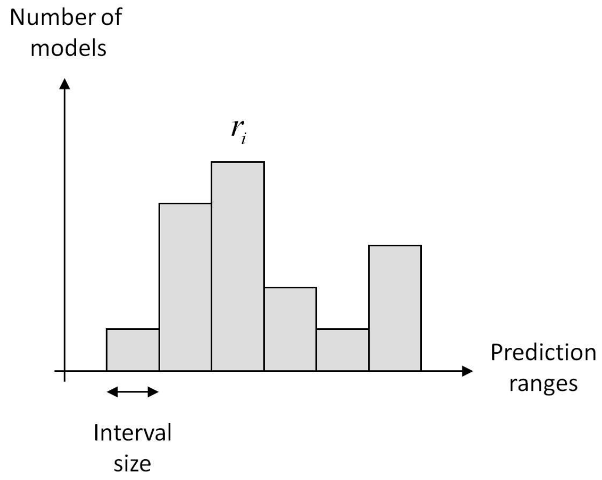


Figure 6.1: Histogram for a specific sensor position. The x-axis is the sensor prediction range. The y-axis is the number of models. The vertical size of each bar corresponds to the number of predictions lying in the interval. The probability P_i is the ratio of the number r_i of predictions in an interval by the total number of predictions r_{tot}.

models and potential sensor locations. Within each iteration, the algorithm finds the optimal sensor configuration using the entropy. Both greedy algorithm (Section 6.1.2) and global search (Section 6.1.3) can be used. Once the sensor placements are chosen, a set of measurements is obtained by the engineer. Note that in this work, simulated measurements are used[2]. Details about the initial measurement system design are given below.

6.1.2 Greedy Search

The greedy algorithm iteratively places each sensor at the best position and does not allow for subsequent relocation when more sensors are added. At each iteration, the sensor location corresponding to maximum entropy is chosen. On the histogram, sets of models lying within an interval size that is bigger than the limit, are considered to be non-identifiable[3]. The process is repeated with all subsets of non-identifiable models to place the next sensor. When all sensors are positioned, an array of dimension two is obtained. It contains, for each iteration, the location chosen for the sensor and the number of non-identifiable models. Sensors are therefore sorted in ascending order according to their efficiency in separating models. Using this methodology, it is observed that minimizing the biggest subset of non-identifiable models at each iteration

[2] Measurements are simulated using ANSYS, a finite element analysis software.

[3] The limit is defined by engineers.

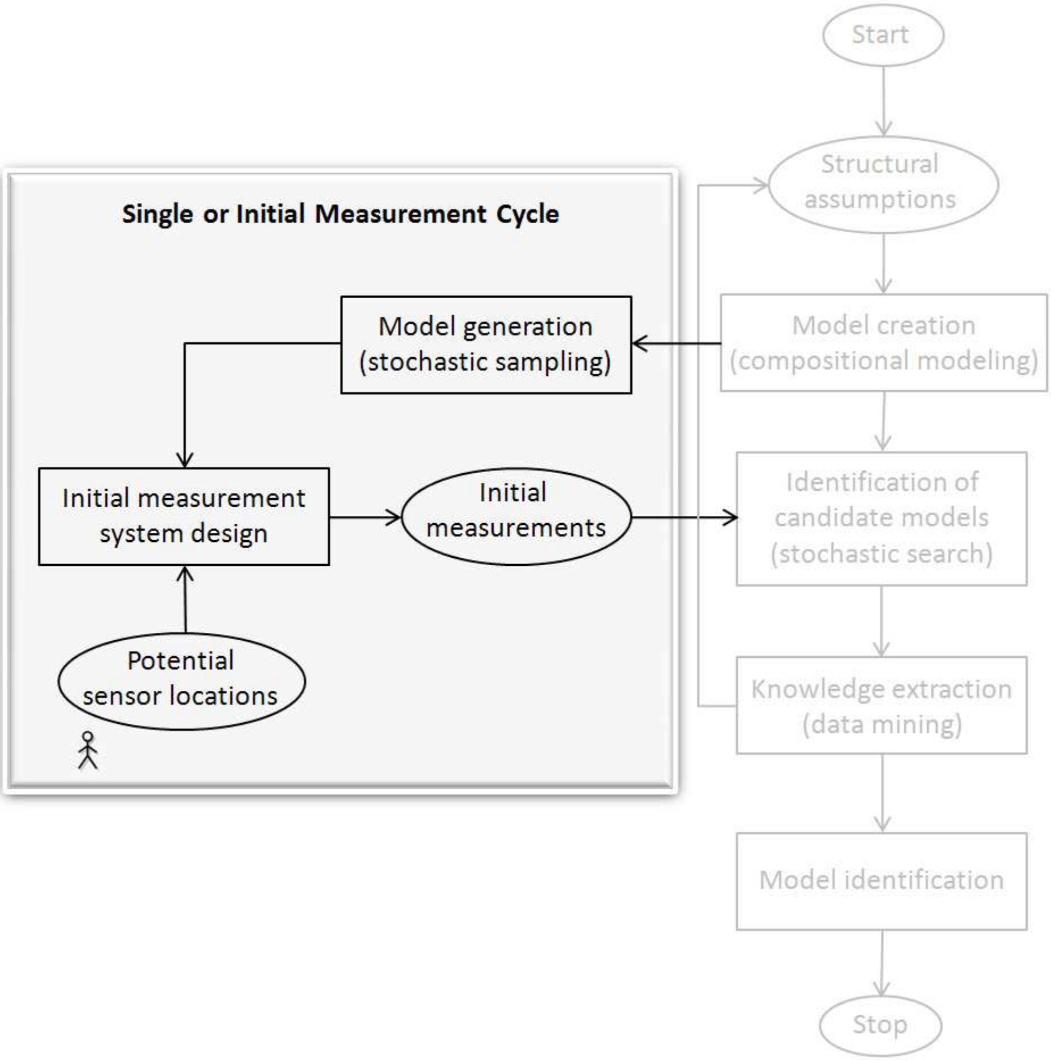

Start
Structural assumptions
Single or Initial Measurement Cycle
Model generation (stochastic sampling)
Model creation (compositional modeling)
Initial measurement system design
Initial measurements
Identification of candidate models (stochastic search)
Potential sensor locations
Knowledge extraction (data mining)
Model identification
Stop

Figure 6.2: Single or initial measurement cycle, part of the overall methodology for system identification. The stick person indicates where human-computer interaction is needed. Details about other parts of the methodology are given in Section 5.2 and 6.2.

corresponds to maximum entropy at each step.

6.1.3 Global Search

Analysis of Complexity

According to the greedy algorithm, previously selected set of sensor locations is unchanged when the next best sensor is determined. This is the principal drawback of the greedy solution. Strategies that accept a less attractive intermediate solution for a better overall solution are not allowed. A more rigorous approach is to test all possible configurations for sensor placement at each iteration of an additional sensor location. At each iteration, all combinations of i sensor(s) among n possible locations are tested. This has the following computational complexity:

$$\sum_{i=1}^{n} C_n^i = 2^n - 1 \tag{6.3}$$

The task of placing i sensors among n possible locations is combinatorial and the total complexity is exponential as shown by Equation 6.3. It is obvious that trying all possible solutions for the measurement system gives the best configuration. However, as explained in Culler and Hong (2004), the number of possible sensor locations can be extremely high. This makes the calculation of all possible configurations infeasible. Therefore, a global search algorithm is used. PGSL (Raphael and Smith, 2003a) is chosen due to its efficiency and ease of use. More details about PGSL can be found in Section 4.1.

Global Sensor Placement (GSP)

This section describes the global sensor placement (GSP) strategy. Unlike the greedy algorithm, the global strategy searches for the best possible solution within the whole solution space. As explained in Section 6.1.3, the solution space is exponential. Therefore, computing all possible solutions is infeasible. Using PGSL, only a subset of the solution space is tested. Better coverage of the solution space is possible by increasing the number of function evaluations and changing PGSL parameter values. The GSP algorithm is described next.

The following parameters are used by the algorithm: n is the total number of possible sensor locations, L is the limit for a group of models to be considered as identified and I is the number of intervals in the histogram of sensor values. The last two are considered to be search parameters of the GSP algorithm. The PGSL function is called i times, where i varies from 1 to n. Within each iteration, the globally optimal configuration of i sensors among n is obtained. There are two stopping criteria for the GSP main loop. Either the total number of possible locations, n, is reached, or there are no further models left to be identified. Both conditions are triggered by values of search parameters. For details about parameters for PGSL the reader is referred to Section 6.1.4. The GSP methodology is presented in Figure 6.3.

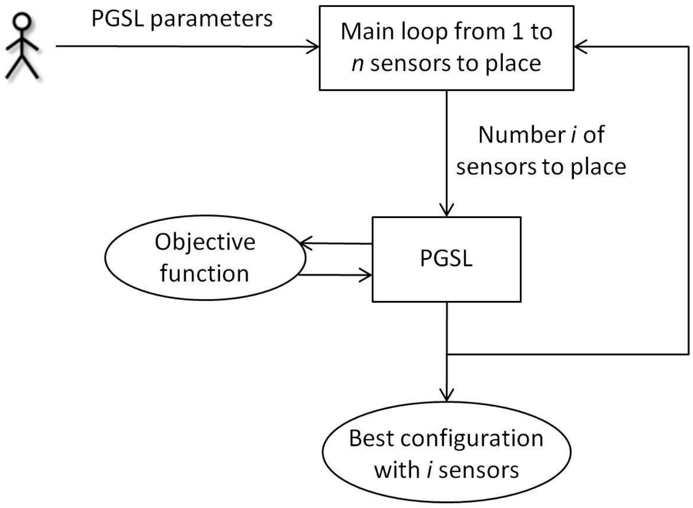

PGSL parameters
Main loop from 1 to *n* sensors to place
Number *i* of sensors to place
Objective function
PGSL
Best configuration with *i* sensors

Figure 6.3: Schema of the global sensor placement (GSP) methodology.

Within each iteration, PGSL finds the optimal solution by minimizing an objective function. The input of the objective function is a sensor configuration that is chosen by PGSL for evaluation. It consists of an array of dimension N containing zeros and ones. A value of 1 means that the sensor is present at this location and 0 means that the sensor is absent. For example, the vector *[0 1 0 0 1 0 0 0]* means that sensors 2 and 5 are present. The output of the objective function varies depending on what is minimized. The number of calls to the objective function is determined by the PGSL parameters (see Section 4.1).

Usually the objective function of an optimization algorithm (such as PGSL) is of the form $y = f(x)$ where x is the input and y the output. In this work, the objective function is a relatively complex procedure; it is not expressible in the form of an explicit mathematical expression. Note that instead of maximizing the entropy, the number of non-identifiable models is minimized. It is due to the fact that there is no way to calculate the entropy for two sensors that are chosen at the same time. Table 6.1 contains the pseudo-code of the objective function.

The main consequence of the GSP strategy is that sensor configuration at iteration i is not dependent on the one at $i-1$ as it is for the greedy algorithm. Figure 6.4 illustrates this concept.

The left part of the picture represents three iterations of the greedy algorithm and the right part shows the GSP for the same iterations. Black dots represent sensors that are selected in the previous iteration and are not changed in subsequent iterations in the greedy algorithm. In

Pseudo-code of the objective function

```
sel_sens contains the sensor configuration proposed by PGSL for evaluation
sub_cell contains all the models
For $i$ from 1 to $size(sel\_sens)$ do
  If $sel\_sens(i) = 1$ then
    While sub_cell is not empty do
      Current_models = first element of sub_cell
      Delete first element of sub_cell
      Calculate distributions (histograms)
      Update non_id with non-identifiable models corresponding to the chosen sensor
      new_cell = new sets of non-identifiable models
    End while
    Update sub_cell with new_cell
  End if
End for
Save current configuration as an already tested one
Return non_id
```

Table 6.1: Details of the objective function called by PGSL during the GSP procedure.

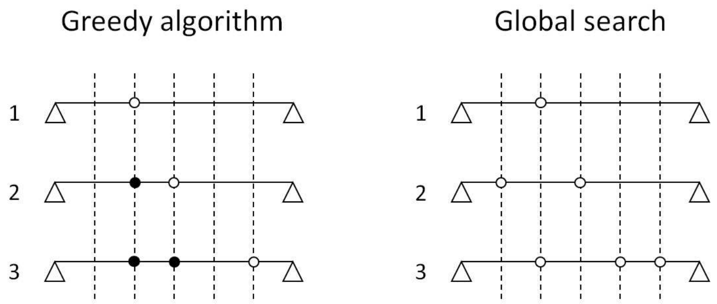


Figure 6.4: Schematic comparison between greedy algorithm (left) and global search (right) strategies. In global search, the configuration at iteration i is not related to the one at iteration $i-1$.

contrast, location results for each iteration of the GSP algorithm are independent of those from the previous iteration.

It is noted that the PGSL variables are continuous. However, the sensor placement task uses discrete variables. This issue could be resolved by rounding the PGSL values that vary from 0 to 1. However, a penalty would then be necessary for PGSL to propose the correct number of sensors at each iteration since there is no constraint on PGSL regarding the number of sensors to place. To simplify the problem, when placing i sensors, the i highest values of the PGSL vector are set to 1, while others are set to 0. The ones correspond to chosen sensor locations. For example, when PGSL proposes the vector [0.1 0.8 0.4 0.3 0.7 0.2 0.1 0.3] and in the case of placing 3 sensors, the transformed vector is: [0 1 1 0 1 0 0 0].

6.1.4 Results

In this section, a case study is used to evaluate the methodology. This involves a laboratory beam structure which is two meters long. A set containing 1000 models is created in order to represent the space of possible models. Models are randomly generated such that each model parameter has values within bounds specified by engineers. The limit chosen for non-identifiable set of models, L, must be an integer greater than 1. The interval number, I, is an integer greater than or equal to 2. In this case, the number of possible sensor locations is 8. Even though the size of the solution space, 2^8, is small, the example illustrates key aspects of the methodology. Parameters taken for the PGSL algorithm are the following: NS = 512, NFC = 1 and NSDC = 1. Consequently, the total number of evaluations of the objective function is 512 ($512 \cdot 1 \cdot 1$). Since the objective of the present study is to find out whether global search can identify better configurations of sensors, a large number of iterations is used in order to ensure the identification of the globally optimal solution. Since PGSL is more suited for larger problems involving continuous variables, a large number of iterations compared to the size of the solution space is needed here. Table 6.2 shows the results obtained when the biggest subset of non-identifiable models is used as the objective function.

Number of sensors	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Greedy search	205	87	38	31	26	25	25	25
Global search	205	82	35	29	26	25	25	25

Table 6.2: Example for greedy and global search algorithms when minimizing the biggest subset of non-identifiable models. Numbers in the table correspond to the size of this subset at iteration i.

First, it can be seen that results of the global search are close to that of greedy algorithm. When placing 2, 3 and 4 sensors, the global search proposes a slightly better solution. However, it is not certain that global search is performing better in general. Second, after iteration 5, there is no clear improvement on the number of non-identifiable models. This means that placing more than 5 sensors in this case will not improve the system identification process. Moreover, the number of sensors from which there is no more improvement is the same for greedy and global search. In other words, in this case, the greedy strategy is sufficient to obtain this information.

When the total number of non-identifiable models is used as the objective function (instead of the biggest subset), results are different. Figure 6.5 shows the difference between selecting the biggest subset of non-identifiable models (left) and the total number of non-identifiable models (right).

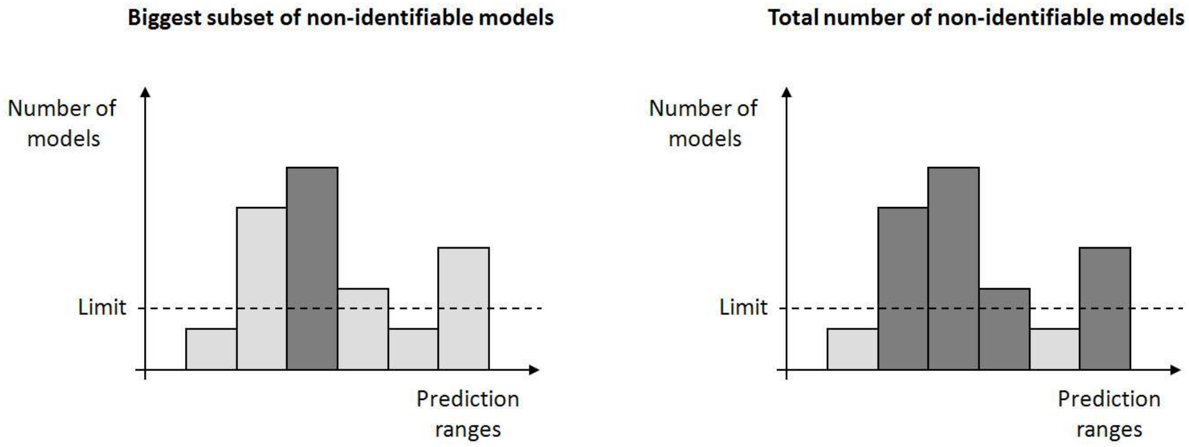


Figure 6.5: Difference between selecting the biggest subset of non-identifiable models (left) and the total number of non-identifiable models (right). Details about the limit are given in Section 6.1.2.

Assume that two sensor locations, A and B identify most number of models using the greedy algorithm. A different set of sensor locations C and D might identify more models due to the fact that locations A and B cross-identify some models. Models are said to be cross-identified when more than one sensor location identify them. When minimizing the total number of non-identifiable models, Table 6.3 is obtained.

In this case, the global search strategy has found better solutions then greedy search for iterations 2 to 7. These optima have not been found by the greedy algorithm. Table 6.4 contains the sensor configurations for the 5 first iterations.

Table 6.4 shows that from iteration 2, sensor configurations are not the same for greedy

Number of sensors	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Greedy search	995	958	889	754	546	481	402	322
Global search	995	946	789	537	443	383	349	322

Table 6.3: Example for greedy and global search algorithms when minimizing the total number of non-identifiable models. Numbers correspond to the total set of non-identifiable models at iteration i.

Number of sensors	**Greedy Search**	**Global Search**
1	[0 0 0 0 0 0 1 0]	[0 0 0 0 0 0 1 0]
2	[0 1 0 0 0 0 1 0]	[0 0 0 0 1 1 0 0]
3	[1 1 0 0 0 0 1 0]	[1 0 0 0 1 1 0 0]
4	[1 1 0 0 1 0 1 0]	[1 0 0 0 1 1 0 1]
5	[1 1 0 0 1 1 1 0]	[1 0 0 1 1 1 0 1]

Table 6.4: Example of sensor configurations chosen for greedy and global search algorithms. In this case, the objective function is the total number of non-identifiable models. Configurations correspond to the five first iterations of the example given in Table 6.3.

as they are for global search. Both strategies are the same for placing the first sensor. The sensor configuration at iteration 2 in global search is independent from the one at iteration 1, which would be impossible with the greedy algorithm. Consequently, depending on the objective function used, global search can be equal or better than the greedy algorithm. However, global search is useless for iterative sensor placement (see Section 6.2). It is thus concluded that the choice of the strategy depends above all on the context and the future use of the measurement system.

6.2 Iterative Sensor Placement

6.2.1 Sensor Placement using Clustering

The process of system identification goes from measurements (consequences) to possible models (causes). This is an abductive task. The unreliability of abductive tasks, and the presence of compensating errors, are the motivations for multiple-model system identification. The correct model for the structure should be contained in the model sets given after model generation. Clustering techniques (see Section 5.2.2) aid in eliminating incorrect models from these model sets and thus rapidly converge to the correct model. Visualizing distributions of models in multi-dimensional parameter spaces is difficult for engineers without suitable computing tools. The use of a data mining method such as clustering can also give engineers a better idea of the topology of the candidate model space. The clustering process is explained in detail in Section 5.2.2.

6.2.2 Detailed Methodology

The overall objective of this part of the thesis is to propose a methodology for improving an existing measurement system - by correctly adding new sensors - in order to support system identification. To achieve this goal, the following methodology is proposed. A schema of this part of the methodology is given in Figure 6.6 and details about it are given below. Details about other parts of the methodology are given in Section 5.2 and 6.1.

Representative model selection: In this step, a few models representing each cluster are selected. Only models which are close to the center of the cluster are selected. In this study, 5% of the total number of models in each cluster are taken to be representative models. This number has been chosen after experimental testing. Then, Shannon entropy is used as a measure of prediction separability to identify the next measurement location (see Equation 6.1). If model sets have high values of entropy, more candidate models can be filtered.

The first stopping criterion is the entropy of remaining sensors. If the entropy of predictions

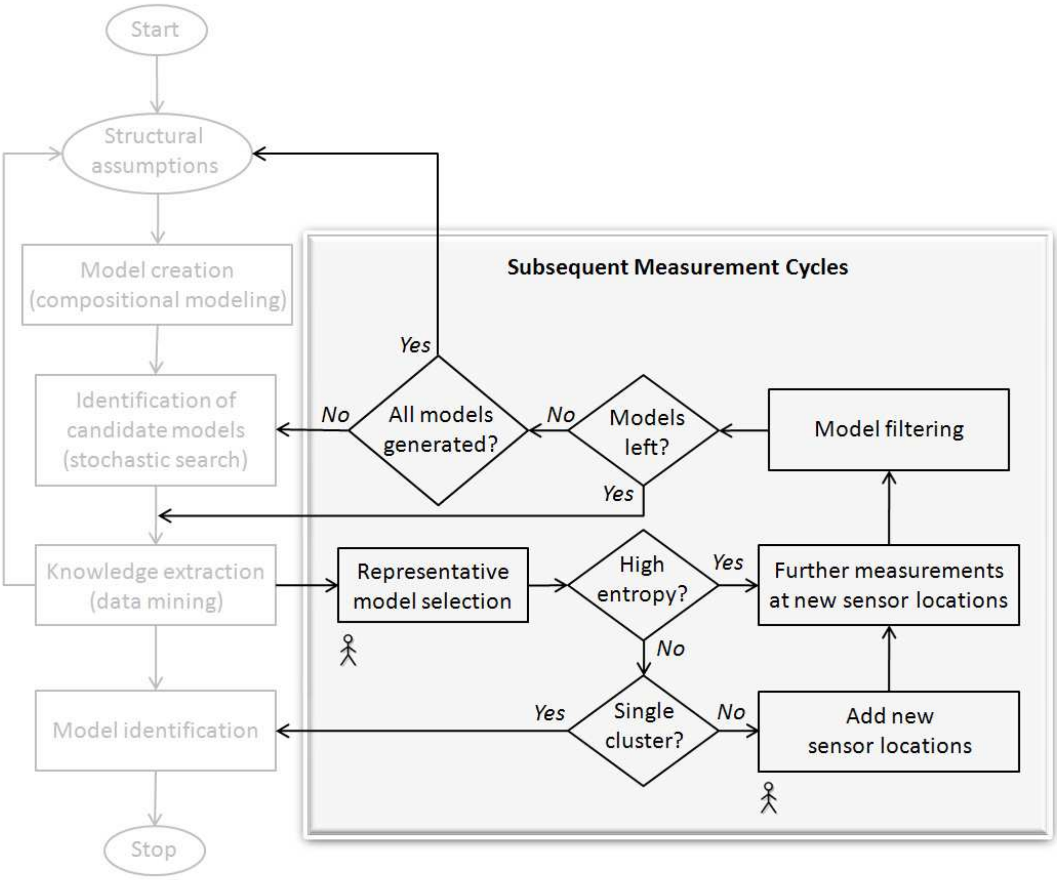

Start
Structural assumptions
Model creation (compositional modeling)
Identification of candidate models (stochastic search)
Knowledge extraction (data mining)
Model identification
Stop
Subsequent Measurement Cycles
Yes
No
All models generated?
No
Models left?
Yes
Model filtering
Representative model selection
High entropy?
Yes
Further measurements at new sensor locations
No
Yes
Single cluster?
No
Add new sensor locations

Figure 6.6: Schema showing the iterative sensor placement part of the methodology for system identification. The stick person indicates where human-computer interaction is needed. Details about other parts of the methodology are given in Section 5.2 and 6.1.

is not significant (below 1) at every sensor location, then the entropy is considered as *low*. If this is not the case, the next step is *sensor addition and further measurements*. If this is the case, it is then checked if there is a single cluster. For that, the score function defined in Section 3.2 is used. Equation 3.15 is thus used as a stopping criterion for further sensor addition. Such a condition may mean that the current set of measurement locations is incapable of further filtering models. The next step is thus *add new sensor placement locations*. If there is only one cluster and the entropy is low, the center of all remaining models is given to engineers as the correct model for the structure (*model identification* step).

Add new sensor placement locations: engineers have to provide other measurement location(s) to the algorithm in order to find the correct model. With the obtained knowledge, engineers propose additional possible sensor location(s) to the system.

Sensor addition and further measurements: During this step, entropies of selected representative models are used to find the position of the next sensor. The location with the highest entropy is chosen as the best position for the next measurement. Then, the measurement is taken on the structure.

Model filtering: In this step, sensor measurements at the new location are compared for every candidate models. Candidate models that do not predict the measurement within the threshold are eliminated from the current set of models. If there are models left, then the next step is *knowledge extraction* through clustering. However, if no model is left, then it is likely that all models were not generated by the *identification of candidate models* step. While it may be possible to generate all models for a lab structure, it is practically impossible to generate all possible candidate models in a real complex structure. In that case, the *identification of candidate models* phase is revisited. On the other hand, if all candidate models have been generated, then some assumptions related to modeling the structure are incorrect. Therefore, structure assumptions have to be checked and modified by engineers (*structural assumptions* step). Generating all possible models for very complex structures may not always be feasible.

6.2.3 Case Study: Schwandbach Bridge

To demonstrate the methodology for sensor addition, the Schwandbach bridge (see Section 5.3.7) is taken as a case study. The Schwandbach bridge has 20 connections. They are shown in Figure 6.7 (numbers from 1 to 20). Possible sensor locations are shown as well (numbers from 1 to 27). The system identification methodology (see Section 6.2) is used to determine the behavior of the structure.

Details of the load case are given in Section 5.3.7. Measurements at different sensor locations (see each example of Section 6.2.4) are given as input to the model generation module. The parameters of the models generated, however, are the logarithms of the stiffness. In this paper,

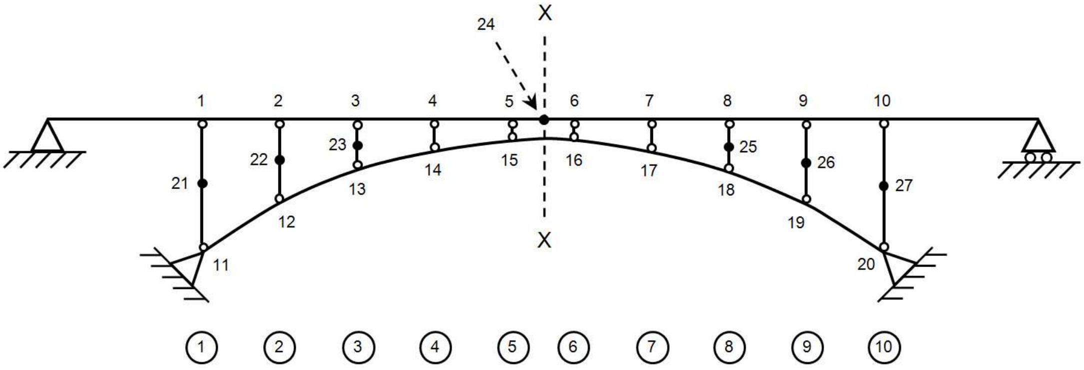


Figure 6.7: Schematic view of the bridge showing the 20 connections (1-20), the 17 possible sensor locations (1-10, 21-27) and the 10 vertical walls (1-10 circles).

only inclinometers are used. Sensor precision are $9.5\mu rad$ (micro radian), τ (see Section 2.6) is taken to be the sum of τ_{meas} ($3\mu rad$) and τ_{pred} ($8\mu rad$).

6.2.4 Results

Example 1

This example illustrates the ability of the proposed methodology to iteratively add sensors in a systematic way in order to identify uniquely the system. The bridge has 10 vertical walls and therefore 10 wall-girder connections and 10 wall-arch connections. For this example, it is assumed that the stiffnesses of the connections in walls 1, 2, 9 and 10 are the same. Other assumptions are (a) symmetry about axis X-X, (b) the stiffness values of the top and bottom connections are equal for each wall and (c) the stiffness values of these connections lie between 10^6 and 10^{12} Nm/rad . Thus, there are three parameters in this example. p_1 represents the stiffness of the connections of walls 3 and 8, p_2 for walls 4 and 7 and p_3 for walls 5 and 6. p_1, p_2 and p_3 are permitted to vary between 6 and 12 (in log space).

For simulation, a model representing the real structure is required. The correct model for this example is given in Table 6.6. Assuming no measurement error, the predictions given by this model are taken to be the measurements. The starting measurement system is assumed to consist of inclinometers measuring the rotation at the following locations: 1, 10 and 24 (Figure 6.7). Since there are only three parameters, models can be directly visualized in three-dimension plots. A total of 1000 candidate models are generated for this example. At the first iteration, only sensor locations on the deck are chosen. This decision is taken because it is easier to place sensors on the deck of the bridge. When the entropy for sensors on the deck is low (below 1), then other sensor locations are also included. Table 6.5 shows the number of models remaining and the selected sensors.

Iterations	**0**	**1**	**2**	**3**	**4**
Number of models	1000	926	907	906	10
Selected sensor location	4	6	5	23	

Table 6.5: Evolution of the number of models at each iteration for example 1 with respect to selected sensor location.

The first observation concerns the sensors on the deck. They filter fewer candidate models compared to the sensor on the vertical wall. After four iterations, the entropy values at the remaining sensor locations are close to zero. Therefore, there is no need to add more than four sensors. Furthermore, at iteration 4, the value of Equation 3.15 is 0.46, which is close to the empirical 0.6 found in Section 3.3.3. This information, in addition to the visual display of the remaining models, can be interpreted as a single cluster by the engineer. Consequently, the mean of this cluster is calculated, and the model closest to this mean is given to the engineer. A plot of the models in the initial parameter space at iteration 0 and 4 are given in Figure 6.8. The model identified as well as the correct model (providing idealized measurements) are given in Table 6.6.

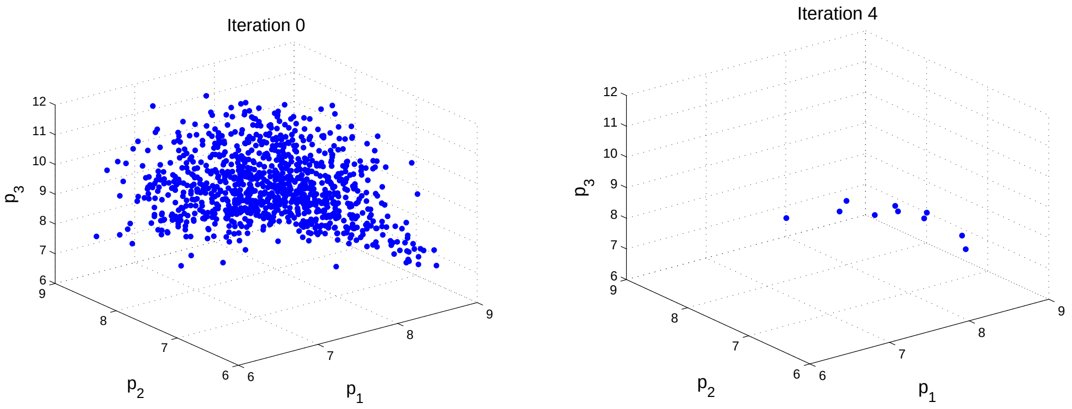


Figure 6.8: Models in the initial parameter space at iteration 0 (left) and 4 (right).

Figure 6.8 shows how the candidate model space decreases from iteration 0 to 4. From Table 6.6 it is noted that the model identified is very close to the correct model for this example. This is especially true for parameters p_1 and p_3. This illustrates the ability of the proposed

Parameters	p_1	p_2	p_3
Correct model	8.0	8.0	8.0
Model identified	8.2	7.4	8.1

Table 6.6: Values of parameters [Nm/rad] for the correct (providing idealized measurements) and identified models in the case of example 1 (in log scale).

methodology to uniquely identify structural systems. This example has only three parameters and a single cluster of models. A more complex example is shown below.

Example 2

In practical situations, the identification problem may involve dozens of parameters. In such cases, it is impossible to visualize the model space as was done for the previous example for reasons of high dimensionality. The identification methodology is illustrated for such an example. The Schwandbach bridge is again considered, however, with more elaborate modeling assumptions. Symmetry about X-X (see Figure 6.7) is assumed. This example models 10 parameters. Each parameter corresponds to two connections, one on either side of X-X. Here, the starting measurement system consists of inclinometers at the following locations: 1, 10, 23, 24 and 25 (Figure 6.7). The stiffness values of each connection vary between 10^2 and 10^{12} Nm/rad. 1719 candidate models are generated for this example. Input data for the PCA part of the methodology are the stiffness values of 10 sets of connections.

The number of clusters is estimated using the score function. The starting point for PCA is a matrix where each row is a different model and each column contains values of a parameter. Figure 6.9 shows the curve of the score function from $k_{min} = 1$ to $k_{max} = 8$ clusters at the very first iteration.

The first observation from Figure 6.9 is regarding the global maximum achieved for $k = 6$. This number has to be interpreted carefully since values for $k = 5$ and $k = 7$ are close to the global maximum. This result has to be combined with the PCA plot of the models (Figure 6.10). The role of engineers here is to carefully interpret these results. This is generally required of the user in any data mining task. According to the results of Figure 6.9, the number of clusters is chosen to be six for this case. The clustering result after applying Table 5.2 procedure is given in Figure 6.10.

In Figure 6.10, every point represents a model. Although all principal components are used in the K-means algorithm, only the first two components are used for visualization. The reader must be aware of the fact that other dimensions (i.e. other principal components) explain these

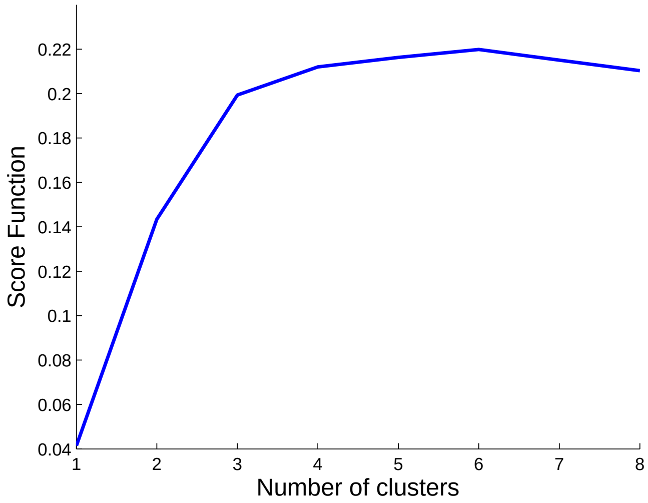


Figure 6.9: Curve of the score function from $k_{min} = 1$ to $k_{max} = 8$ clusters. The best value is taken over $t = 20$ runs.

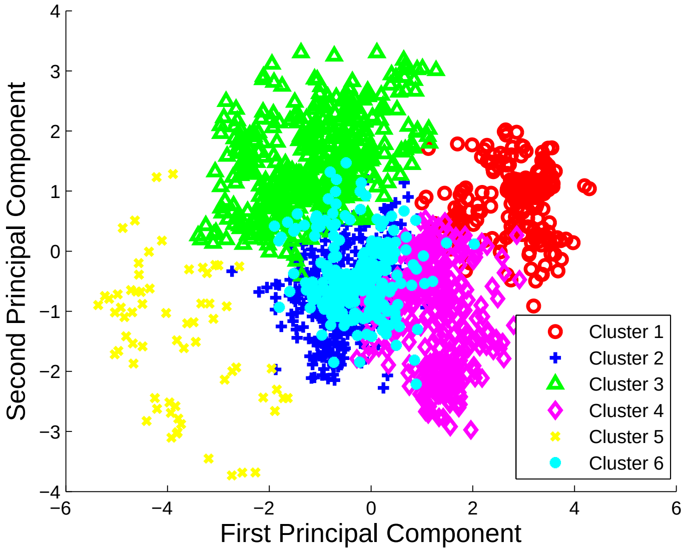


Figure 6.10: Clustering results at the very first iteration. Each point represents a model that is displayed using the first two principal components (out of 10).

data. Even if not well defined, clusters are already visible. In addition, clusters also contain outliers. This is not an issue since the score function is using the cluster size as a weight in Equation 3.10 and 3.11. Again, this plot taken alone is not enough to estimate the correct number of clusters. This is mainly due to the dimensionality of the data set and the overlapping between clusters. Combined with Figure 6.9, it can help engineers estimate the most reliable number of clusters. The centroid of each cluster defines a possible state of the structure. Instead of having to examine 1719 models, engineers can examine the six groups of models, each represented by its center. Indeed, the center of each cluster represents a bridge with a particular set of stiffness values for the connections.

The next step is to iteratively add sensors to reduce the total number of models. Representative models are selected in each cluster for evaluating entropy. Representative models are chosen around each cluster centroid. This way, only models that *represent* the cluster are taken into account. The selected set of representative models is 5% of the total number of remaining models. This set is proportionate to the cluster size (i.e. the number of models inside the cluster). Therefore, bigger clusters have more influence on the selection of the next sensor. Figure 6.11 shows the representative models selected at the first iteration.

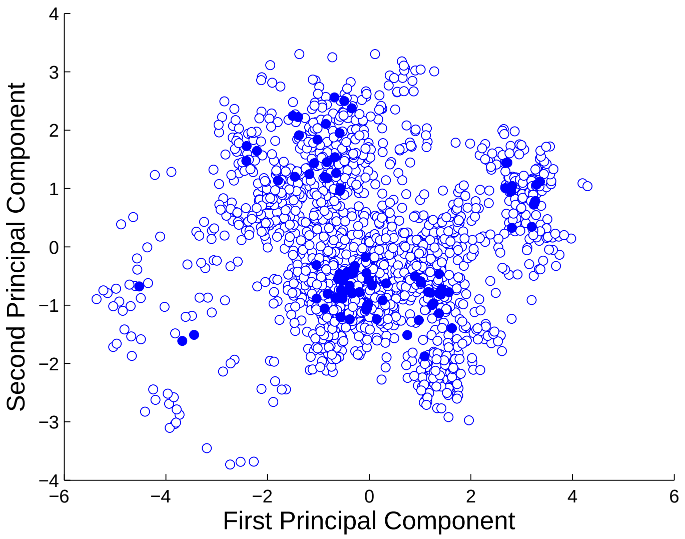


Figure 6.11: Plot of the representative models (solid circles) among other models (open circles) for the first iteration.

The plot of Figure 6.11 shows that representative models are a good representation of each cluster. Entropy is calculated at every remaining sensor location for the representative model predictions and a sensor is added at the location with highest entropy. The entropy value is found to be a valid stopping criteria for the methodology. Once the new sensor is known, a new measurement is taken. All models whose predictions do not match the new measurement are eliminated. Figure 6.12 shows a plot of the models and their error (Equation 2.6) after adding the new sensor.

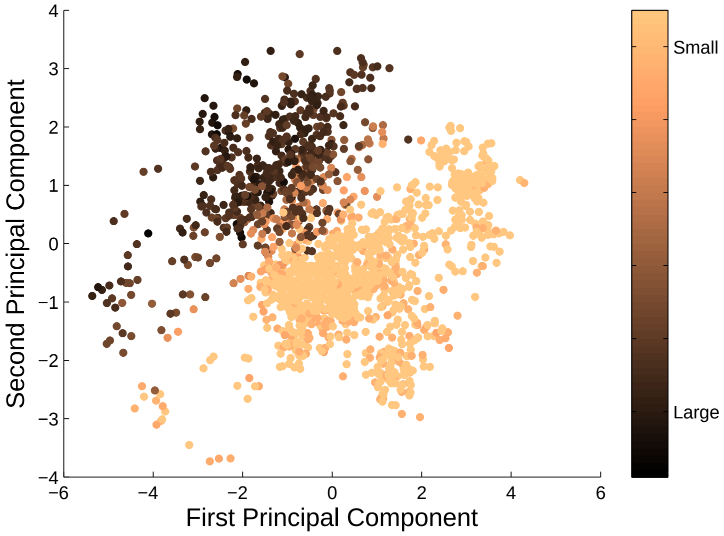


Figure 6.12: Plot of the error of each model after adding the first new sensor (sixth sensor). The darker the model, the higher the error between its predictions and the measurements.

Models with a high error (dark) are filtered for the next iteration. This is repeated until the entropy of model predictions is zero for every sensor location. At each iteration, the number of models is either reduced or the same.

In this case, the methodology is unable to converge to the unique model for the bridge. At iteration 3, multiple clusters are still present. Indeed, Equation 3.15 has a value of 0.13, which informs engineers that there are multiple clusters in the model space. This indicates that the remaining sensor locations are incapable of further reducing the number of candidate models. At this juncture, engineers can consider adding more load cases, including other sensor types and augmenting the set of sensor locations. Engineers could also opt to look at the cluster centroid from each cluster.

Table 6.7 shows that sensors on the deck are useful for reducing the number of candidate models in this example. This was not the case in the previous example. Therefore, it can be concluded that the choice of sensor locations is understandably dependent on the parameter set.

Iterations	**0**	**1**	**2**	**3**
Number of models	1719	923	243	71
Selected sensor location	8	21	26	

Table 6.7: Evolution of the number of models at each iteration for example 2 with respect to selected sensor location.

Table 6.8 shows the entropy of each sensor from iteration 0 to 2 (all entropy values are 0 at iteration 3). From Table 6.8, it is observed that locations on the vertical walls have a higher entropy and are thus better than locations on the deck to identify the system. At iterations 0 and 1, all locations on the deck have an entropy that is smaller than entropies for locations on the walls. The table also shows that the best location for a particular iteration is dependent on the locations chosen in the previous iteration. At iteration $i+1$ the entropy for a given sensor is not the same as at iteration i. After each iteration, models are filtered, and therefore the entropy of each remaining sensor may be different. In this example no unique model is identified, rather the model closest to the mean of every cluster is given to the engineer. Values of the identified models as well as the correct model are given in Table 6.9.

Iteration 0		**Iteration 1**		**Iteration 2**	
Sensor	Entropy	Sensor	Entropy	Sensor	Entropy
26	3.58	**21**	2.47	**26**	1.49
21	3.45	27	1.93	22	1.31
27	3.12	26	1.88	2	0.00
22	3.12	22	1.64	3	0.00
8	2.46	3	0.86	4	0.00
3	2.30	7	0.67	5	0.00
4	2.19	2	0.00	6	0.00
2	2.04	4	0.00	7	0.00
7	1.96	5	0.00	9	0.00
9	1.86	6	0.00	27	0.00
6	1.46	9	0.00		
5	0.90				

Table 6.8: Sensors and their corresponding entropy to every sensors. Values in bold represent the chosen sensors. After iteration 2, the entropy value is zero for every remaining sensor location.

Parameters	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
Correct model	3.0	3.0	7.0	7.0	10.0	10.0	7.0	7.0	3.0	3.0
Identified model 1	5.1	6.1	5.0	4.5	10.0	10.0	6.6	6.3	4.7	5.1
Identified model 2	7.2	6.6	7.1	7.4	9.9	10.0	6.8	6.4	7.7	6.9
Identified model 3	7.1	4.5	6.1	7.1	10.1	10.0	7.1	5.7	5.5	4.0
Identified model 4	3.2	3.3	5.2	5.6	10.0	10.1	5.4	6.6	3.6	6.2
Identified model 5	4.7	7.8	5.0	4.8	7.6	10.0	7.4	5.2	8.3	9.6
Identified model 6	5.0	6.2	6.8	6.5	10.1	10.1	6.9	6.4	5.7	5.8

Table 6.9: Correct model of the problem and models identified in the case of example 2 (in log scale).

From Table 6.9 it is noted that more than one model is proposed as a correct model. Among them, only solution 4 is closest to the correct model. The values for the different parameters show some common features among the solutions. Nearly all models have a value of 10 for both p_5 and p_6. Since the variation in these parameters is very small, they are likely to have a much larger influence on predictions than other parameters. The other parameters do not significantly affect the behavior of the bridge. In other words, the connections closer to the ends could be modeled as hinged or rigid and it would not generate changes in displacements that are detectable with the precision of inclinometers considered in this study. However, sensor technology is improving day-by-day and precision of sensors are gradually increasing. Further choices also include measurement of other phenomena using other sensors.

Finally, for a better visualization of the obtained solutions, three models (out of the six identified) are shown in Figure 6.13. Although these models are clearly different, they are all candidate models (their predictions are close to measurements).

6.2.5 Conclusions

Both initial and iterative sensor placement are important parts of the methodology. It is most likely to obtain useful knowledge if sensors are optimally placed. A global strategy is compared to a greedy algorithm for initial sensor placement. A clustering algorithm is integrated in a greedy strategy for iterative sensor placement. Conclusions are given below:

- For a simple case study, global search provides the same results as greedy search when the size of the biggest subset of non-identifiable models is chosen as the minimization function. When the total number of non-identifiable models is used as the minimization function, there are good solutions which are missed by the greedy algorithm.

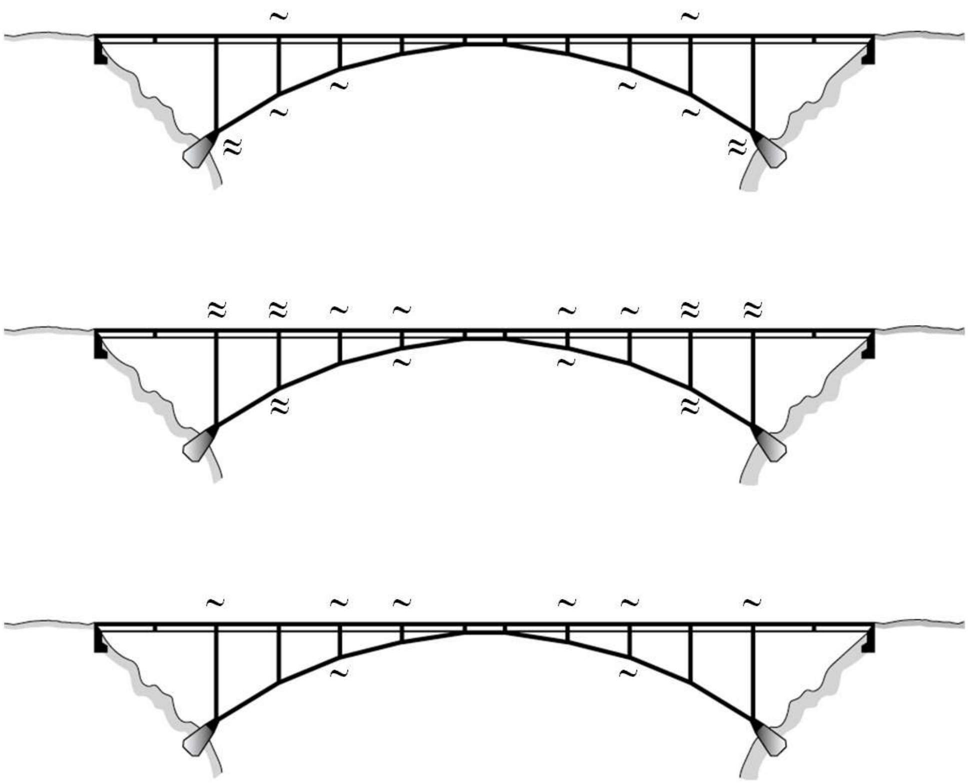


Figure 6.13: Three candidate models taken from Table 6.9 (models 3, 4 and 5). A double tilde means a big crack (stiffness below 4 in log scale) and a single tilde means a stiffness varying from 4 to 6 (in log scale). Finally, positions with no symbol are rigid connections (more than 6 in log scale). The difference between predictions and measurements is small in each case.

- The choice between greedy algorithm and global search should be done according to the context. When additional measurements are required from an existing measurement system, greedy algorithm is the best choice.

- When improving measurement systems, the choice of sensor locations depends on the parameter set.

- Low entropy values obtained at every sensor position are found to be a good stopping criteria for the iterative sensor placement methodology.

- Clusters may indicate different possible types of behavior of a structure, thus guiding subsequent decision making related to new measurements and further system identification studies.

Data mining is thus a useful decision support tool for engineers performing system identification in iterative measurement-evaluation methodologies.

Chapter 7

Conclusions

"*I am turned into a sort of machine for observing facts and grinding out conclusions.*" (Charles Darwin, 1809-1882)

Overview

This chapter starts with a discussion of the results of this thesis. A discussion of the research methodology and its validation follows. Conclusions are then drawn for each chapter. Finally, opportunities for future work are presented.

7.1 Discussion

In this work, the potential for data mining strategies in system identification is evaluated. For example, valuable knowledge is extracted through clustering (Chapter 3) and feature selection (Chapter 4). These techniques are integrated into an overall methodology for system identification (Chapter 5). An iterative methodology for sensor placement is also proposed and supported through data mining techniques such as clustering (Chapter 6).

A new index for hard clustering called the score function (SF), is presented and studied in depth in this work. More specifically, the SF is better or as good as six other validity indices (Dunn, Calinski-Harabasz, Davies-Bouldin, Silhouette, Maulik-Bandyopadhyay and Geometric) for the K-means algorithm on hyper-spheroidal clusters. In addition, the SF has been tested successfully on multidimensional real-life data sets. The SF can also accommodate perfect and single cluster cases. In order to identify the single cluster case, an empirical condition is formulated. It is important to note that calculating the SF is computationally efficient.

Concerning feature selection, the PGSL-SVM results show that this technique is competitive with others in the literature. In comparison to other techniques, PGSL-SVM is better for several data sets while it is within the range of other results for a few data sets. This new feature selection algorithm combines global search (PGSL) and support vector machines (SVM) in a wrapper approach. It is found that PGSL-SVM is an efficient feature selection strategy since it performs in general as well as GA-SVM (genetic algorithm SVM) for feature selection on various data sets. The PGSL-SVM strategy finds feature subsets that are smaller than GA-SVM. It is noted that the proposed strategy is easier to use since it requires less tuning of parameters than GA-based strategies.

This work is a new application for the data mining field. It is a successful example of application in the area of system identification in the domain of civil engineering. A system identification methodology that accounts for factors influencing the reliability of identification is proposed. The importance of a multi-model approach is demonstrated with a real case study. The need for data mining techniques to make sense of multiple models is shown using two different examples. Linearly independent variables are identified by the correlation measurement. The first principal components in PCA consist of independent variables whose values are identified. A few principal components are sufficient for explaining the variation in data, implying relationships among variables. Clustering is useful for grouping models into clusters thus providing information to engineers about possible model classes. Clusters may indicate different possible types of behavior of a structure. Feature selection highlights parameters that explain candidate models. This can thus guide subsequent decision making related to new measurements and further system identification studies.

This methodology raises an important issue regarding data interpretation. Firstly, results of data mining have to be interpreted carefully. The user thus has an important role in ensuring that the methodology is successful. Secondly, even if the methodology is well applied, results are not necessarily entirely useful. For example, data might be noisy (poor sensor precision), or may have missing values (low sensor quality) or may be missing useful information (bad sensor configuration) and this may preclude obtaining useful results.

An example of challenges associated with applying data mining to system identification is given below. Assume that, after applying data mining, three clusters of models are obtained. The methodology alone is not able to *interpret* these clusters. Suppose that two clusters group similar information. Although the clustering algorithm has generated three clusters, only the user is able to identify that there are only two clusters that have physical meaning. Therefore, data mining is only able to suggest possible additional knowledge. As written in Kuonen (2004), "*Even the most advanced algorithms cannot figure out what is most important*". The process of acquiring and confirming that knowledge is of practical use for decision is left for engineers.

Regarding the initial sensor placement, depending on the objective function used, global search can be equal or better than the greedy algorithm. However, global search is less useful for iterative sensor placement, especially when sensors cannot be moved (see Section 6.2). Therefore, the choice of the strategy depends above all on the context and the future use of the measurement system. The ability of the iterative sensor placement methodology to uniquely identify the system is illustrated in a case study. This example has only three parameters and a unique cluster of models. Using a more complex case study, it has been found that the choice of sensor locations is dependent upon the parameter set. Furthermore, the entropy value obtained at every sensor position is an iterative indication of the number of sensors needed on the structure.

7.2 Validation

The research methodology and plan followed in this research is summarized in Figure 1.2. Each activity shown in the figure has been evaluated using one or several validation data (benchmark data sets, laboratory structure and Schwandbach bridge). Parameters used for validation are also given in Figure 1.2. The research performed is thus empirically validated using these parameters. Below, a summary of the evaluation of each research activity in terms of these parameters is provided.

The potential for existing data mining techniques was evaluated by applying them to a laboratory structure and assessing the following three parameters:

1. Amount of additional knowledge generated
2. Increase in the level of support
3. Enhancement of visualization capabilities

Each of these three parameters are detailed below. The following are examples of important knowledge generated by data mining:

- Independent parameters
- Linear relationship between parameters
- Number of principal components

Regarding the increase in level of support, it is measured by the reliability of system identification. The knowledge obtained informs engineers about independent (and thus reliable) parameters. Through this knowledge, engineers are supported in the overall iterative methodology for system identification.

The visualization capabilities are measured by the improvement in presenting multiple models to engineers. To present thousands of multidimensional models to engineers, visualization capabilities are enhanced through:

- Dimensionality reduction (PCA)
- Visual representation of clusters

Thus, the three validation parameters indicate that existing data mining techniques have potential to provide reasonable level of support for system identification. However, current data mining techniques are limited in reliably and efficiently providing the following useful knowledge:

1. The number of clusters in data
2. Parameters explaining candidate models

The development of a new index for cluster validity helps answer the first point. Examples of capabilities of the score function are:

- Estimating the number of clusters in a given data set
- Linear computational complexity
- Ability to handle perfect, single and sub-cluster cases

Regarding the parameters explaining candidate models, the efficiency of the new wrapper feature selection algorithm (PGSL-SVM) is shown by the following aspects:

- Good classification accuracy
- Subsets with a smaller number features than genetic algorithms (GA)
- Less tuning parameters to fix than GA

The above techniques are integrated in the new methodology for decision support and the new measurement system design. The overall methodology is validated on the laboratory structure and the Schwandbach bridge. The parameters used to validate the methodology are:

1. The amount of remaining information to measure
2. The number of remaining models
3. The number of sensors placed

Regarding the first point, it is measured by entropy calculations. This measure gives an idea of the information remaining. When the entropy value at each sensor location is low (< 1), no more sensors are needed on the structure. The entropy value decreases iteratively until no additional sensors are needed. The number of remaining models measures the speed of convergence of the algorithm. Finally, the number of sensors placed measures the efficiency of the sensor placement algorithm. The fewer sensors are needed, the better it is.

From the parameters mentioned in this section, it is concluded that the developed algorithms, in combination with existing data mining techniques, have good potential for system identification tasks.

7.3 Conclusions

The following conclusions are related to the application of data mining techniques to system identification. Several data mining techniques have been integrated in the system identification methodology, thus avoiding narrow points of view that can happen when dealing with only one method. It is well known that "*when the only tool you have is a hammer, everything looks like a nail*" (Zadeh, 2001). This work brings together different ideas and concepts to avoid this problem. It is shown that, when integrated in an overall framework for decision support, data mining techniques constitute a valuable tool for engineers performing system identification. General conclusions of this work are separated in four groups and detailed in the next subsections.

7.3.1 Cluster Validity

A score function is proposed to support two important issues in clustering which are i) evaluating obtained results and ii) estimating the number of clusters. The following conclusions are drawn:

- The score function is a reliable index for estimating the number of clusters in a given data set.
- The score function can be used on a wide range of data set sizes, since its computational complexity is linear.
- The generalization abilities of the score function are high due to its particularities, such as its ability to handle perfect, single and sub-clusters cases

The score function is able to efficiently estimate the number of groups in a given data set, thus increasing clustering results understanding.

7.3.2 Feature Selection

A new feature selection algorithm using global search and support vector machine (SVM) in a wrapper approach has been proposed. Experiments on several data sets have lead to the following conclusions:

- PGSL-SVM is an efficient feature selection strategy since it performs in general as well as GA-SVM for feature selection on various data sets.
- The PGSL-SVM strategy finds subsets with a significantly smaller number features than GA-SVM for the same order of accuracy and time.
- The strategy involving PGSL is easier to use since it has less tuning parameters than GA-based strategies. This number is of importance since bad tuning can lead to poor results.

Global search algorithms (PGSL) and kernel methods (SVM) are examples of useful tools for searching the space of possible feature combinations which is a combinatorial problem.

7.3.3 System Identification Methodology

A methodology that integrates data mining into the system identification process is proposed. The process is iterative and requires engineer interaction. The overall methodology is tested on an existing structure to validate its possibilities in decision support. Although this approach has much potential to be generalized to many applications, the scope of the conclusions is limited to the applications that were studied in this thesis. The following conclusions are of interest:

- Data mining techniques are essential for interpreting the results of system identification
- Clusters can guide subsequent decision making related to further system identification studies since they can indicate different possible types of behavior of a structure.
- The combination of K-means and PCA improves understanding of the model space as it allows an efficient visualization for engineers, even if the data set is multi-dimensional.
- The application of data mining to complex tasks such as system identification requires an expert user since the methodology is iterative and needs input from engineers.
- The overall applicability of the methodology goes beyond toy examples and laboratory cases since it is applied to a real structure in Switzerland.

Data mining is a valuable tool which, when used by engineers, increase their understanding of the system for subsequent decision making.

7.3.4 Sensor Placement

As part of the overall methodology for system identification, clustering is integrated in a greedy strategy for sensor placement. Although this approach has much potential to be generalized to many applications, the scope of the conclusions is limited to the applications that were studied in this thesis. Conclusions are given below:

- When integrated in a methodology for iterative sensor placement, clustering is found to be a relevant tool for supporting engineers by improving subsequent sensor placement on existing structures.
- Greedy and global search strategies should be used according to the context. Experiments show that whereas global search is more efficient for initial sensor placement, a greedy strategy is more suitable for iterative sensor placement.
- When improving measurement systems, the choice of sensor locations depends on the parameter set chosen by engineers.
- The entropy value obtained at every sensor position is an iterative indication of the number of sensors needed on the structure. It is therefore used as a stopping criterion.

In addition to providing information about the candidate model space, data mining is found to be a valuable tool for supporting additional sensor placement.

7.4 Future Work

During the period that was available for this research, several ideas and concepts have emerged. However, due to time limitations only a few have been developed, implemented and tested. Nevertheless, generation of these experimental ideas are also important contributions of this research. Below is a list of possible subsequent research in the areas covered by this thesis. They serve as starting points for future research in this field.

7.4.1 Improvement of Clustering Procedure

Further work can be performed on clustering. For example, in Section 5.2.2, self-organizing map (SOM) can be used instead of PCA. In addition, other clustering algorithms are available. Stability of clusters can also be studied. For example, through consensus clustering (Monti et al., 2003), it is possible to represent consensus of results across multiple runs of a clustering algorithm. It thus helps to know how the clustering algorithm (e.g. K-means) is affecting results.

7.4.2 Supervised Learning on Clusters

Once clusters are found, they can be labeled and considered as classes of models. Thus, a supervised learning technique can be used to obtain additional knowledge from these classes. For example, a decision tree can be built on the data set where each cluster is considered as a different class. Figure 7.1 gives a possible methodology for supervised learning on clusters.

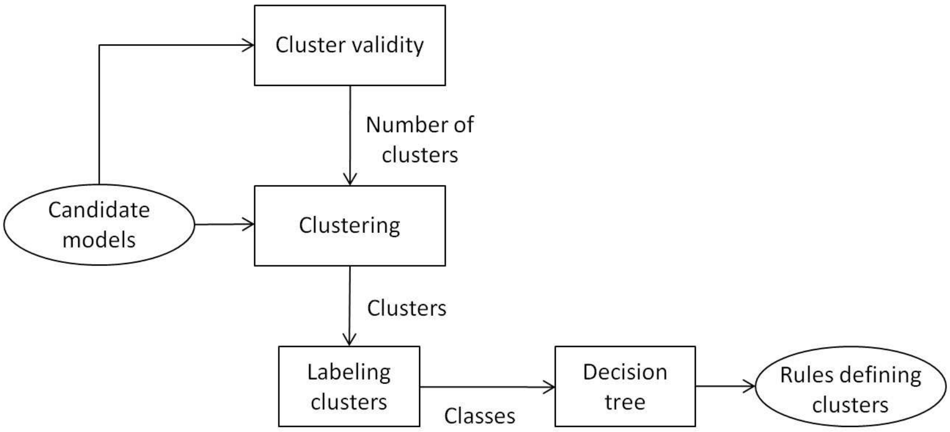


Figure 7.1: A possible strategy for extracting rules from clusters.

This decision tree thus leads to a set of rules. The idea is to give engineers simple and readable rules which characterize each class. In this way, engineers would have rules describing classes. Given these rules, engineers can have a better idea of the topology of each cluster (i.e. each cluster would be defined by a set of rules on the parameters).

7.4.3 Feature Selection for Dynamic Response

All the data in this thesis come from static measurements. For example, a truck is placed on a bridge and then system identification is performed to discover the state of the structure. The next step is dynamic measurements. Measurements are taken every t seconds. Dynamic feature selection may be necessary. A different subset of features is selected every t seconds. It is thus possible to see how important different parameters are, according to the position of the truck.

7.4.4 Association Rule Mining on Model Parameters

Association rules mining can be applied to both candidate and non-candidate models to obtain information for engineers. When used on model parameters, association rules of the form "*80% of candidate models have* $p_1 < a$" or "*If a model is candidate and has* $p_3 > b$ *then there is a*

95% chance that $p_5 < c$" may be useful. Association rules can thus provide useful knowledge to engineers related to the reliability of system identification.

7.4.5 Data Containing Varying Number of Parameters

A limitation of current work is that only models containing a fixed number of parameters can be processed. This is due to the fact that most current data mining techniques are limited to data sets containing a fixed number of columns (i.e. parameters). Advances in relational data mining may provide more flexibility. In relational data mining, several "tables" with different number of parameters can be processed together. Dimensionality reduction techniques may be useful. Examples are multidimensional scaling (MDS), locally linear embedding (LLE) and isomap (Hadid and Pietikainen, 2004).

7.4.6 Improvement of Sensor Placement Methodology

Regarding the iterative sensor placement strategy, several extensions are possible. In this work, the number of representative models is fixed by the user. It may be difficult to find a relevant number without a specific framework to help. Work toward devising a standard way of estimating the number of representative models required from each cluster to identify subsequent measurement locations would be of interest. Another issue is the number of candidate models required for correct system identification. In the present work, this number is fixed by the size of the model space (number of possible models) and the computational time required to sample the solution space. For example, this problem can be treated probabilistically.

Another approach to sensor placement is to account for measurement redundancy to perform system identification. This is already done in fault diagnosis for example (Isermann, 1993). Frisk and Krysander (2007) state that redundancy, which is needed, is provided by several sensors.

Chapter 8

Appendix

8.1 Acronyms

Below is a list of abbreviations used in this thesis:

ACM	Association for Computing Machinery
ANN	Artificial Neural Network
ARR	Analytical Redundancy Relations
BCD	Between Class Distance
CBR	Case-Based Reasoning
CH	Calinski-Harabasz index
CSP	Constraint Satisfaction Problem
DB	Davies-Bouldin index
DIKW	Data Information Knowledge Wisdom hierarchy
DNA	Deoxyribonucleic Acid
DU	Dunn index
GA	Genetic Algorithm
GE	Geometric index
GSP	Global Sensor Placement
KDD	Knowledge Discovery in Databases
LDA	Linear Discriminant Analysis
MB	Maulik-Bandyopadhyay index
NA	Not Available
NS	Number of Samples
NFC	Number of loop in the Focusing Cycle
NSDC	Number of loop in the Sub-Domain Cycle
PC	Principal Component

PCA	Principal Component Analysis
PDF	Probability Density Function
PGSL	Probabilistic Global Search Lausanne
RFE	Recursive Feature Elimination
SA	Simulated Annealing
SBFS	Sequential Backward Floating Search
SBS	Sequential Backward Selection
SF	Score Function
SFFS	Sequential Forward Floating Search
SFS	Sequential Forward Selection
SI	Silhouette index
SOM	Self-Organizing Map
STD	Standard Deviation
SVM	Support Vector Machine
WCD	Within Class Distance

8.2 Miscellaneous

Several user manuals have been used to benefit from the LaTeX formatting:

- "*Une courte introduction à LaTeX* 2ϵ", Oetiker, T., Partl, H., Hyna, I., Schlegl, E. and Herrb, M., 2001.
- "*Natural Science Citations and References - natbib*", Daly, P.W., 2006.
- "*Using LaTeX to Write a PhD Thesis*", Talbot, N., 2006.
- "*The geometry package*", Umeki, H., 2002.

List of Figures

List of Tables

References

Abad, P., Suarez, A., Gasca, R., and Ortega, J. (2002). Using supervised learning techniques for diagnosis of dynamic systems. In *International Workshop on Principles of Diagnosis (DX-02)*.

Abudayyeh, O., Dilbert-DeYoung, A., Rasdorf, W., and Melhem, H. (2006). Research publication trends and topics in computing in civil engineering. *Journal of Computing in Civil Engineering*, 20(1):2–12.

Ackoff, R. (1989). From data to wisdom. *Journal of Applied Systems Analysis*, 16:3–9.

Addanki, S., Cremonini, R., and Penberthy, J. (1991). Graphs of models. *Artificial Intelligence*, 51(1-3):145–177.

Aery, M. and Chakravarthy, S. (2005). emailsift: email classification based on structure and content. In *Fifth IEEE International Conference on Data Mining*, pages 18–25.

Ailamaki, A., Faloutos, C., Fischbeck, P., Small, M., and VanBriesen, J. (2003). An environmental sensor network to determine drinking water quality and security. *SIGMOD Rec.*, 32(4):47–52.

Alonso, C., Rodriguez, J., and Pulido, B. (2004). Enhancing consistency based diagnosis with machine learning techniques. *Lecture Notes in Computer Science*, 3040:312–321.

Ari, I. (2004). Using statistical correlation for dependency analysis of cache replacement policies. Technical report, University of California.

Bagajewicz, M. and Sanchez, M. (2000). Cost-optimal design of reliable sensor networks. *Computers and Chemical Engineering*, 23(11):1757–1762.

Bahlmann, C., Haasdonk, B., and Burkhardt, H. (2002). Online handwriting recognition with support vector machines - a kernel approach. In *Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 49–54.

Balakrishnan, K. and Honavar, V. (1998). Intelligent diagnosis systems. *Journal of Intelligent Systems*, 8(3/4):239–290.

Banan, M., Banan, M., and Hjelmstad, K. (1994). Parameter estimation of structures from static response: 1. computational aspects. *Journal of Structural Engineering*, 120(11):3243–3258.

Banta, J. and Abidi, M. (1996). Autonomous placement of a range sensor for acquisition of optimal 3-d models. In *Proceedings of the 1996 IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, volume 3, pages 1583–1588.

Barzilay, O. and Brailovsky, V. (1999). On domain knowledge and feature selection using a support vector machine. *Pattern Recognition Letters*, 20(10):475–484.

Bellinger, G., Castro, D., and Mills, A. (2005). Data, information, knowledge, and wisdom.

Ben-Hur, A., Elisseeff, A., and Guyon, I. (2002). A stability based method for discovering structure in clustered data. In *In Pacific Symposium on Biocomputing*, pages 6–17.

Berkhin, P. (2002). Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA.

Bezdek, J. and Pal, N. (1998). Some new indexes of cluster validity. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):301–315.

Bi, J., Bennett, K., Embrechts, M., and Breneman, C. (2003). Dimensionality reduction via sparse support vector machine. *Journal of Machine Learning Research*, 3:1229–1243.

Bins, J. and Draper, B. (2001). Feature selection from huge feature sets. In *International Conference on Computer Vision*, pages 159–165, Vancouver, Canada.

Blum, A. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271.

Bolshakova, N. and Azuaje, F. (2003). Cluster validation techniques for genome expression data. *Signal Processing*, 83(4):825–833.

Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In Haussler, D., editor, *5th Annual ACM Workshop on COLT*, pages 144–152. ACM Press.

Bouguessa, M., Wang, S., and Sun, H. (2006). An objective approach to cluster validation. *Pattern Recognition Letters*, 27(13):1419–1430.

Bradley, P. and Fayyad, U. (1998). Refining initial points for k-means clustering. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 91–99, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Brownjohn, J. (2007). Structural health monitoring of civil infrastructure. *Philosophical Transactions of the Royal Society A*, 365(1851):589–622.

Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., and Dougherty, E. (2007). Model-based evaluation of clustering validation measures. *Pattern Recognition*, 40(3):807–824.

Cabanes, G. and Bennani, Y. (2008). A simultaneous two-level clustering algorithm for automatic model selection. In *International Conference on Machine Learning Applications*.

Cakmakov, D. and Bennani, Y. (2002). *Feature Selection for Pattern Recognition*. Informa, Skopje.

Calinski, R. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3:1–27.

Candillier, L., Meyer, F., and Boullé, M. (2007). Comparing state-of-the-art collaborative filtering systems. In Perner, P., editor, *5th International Conference on Machine Learning and Data Mining in Pattern Recognition*, volume LNAI 4571 of *LNCS*, pages 548–562. Springer Verlag.

Carbone, P. (2000). What is the origin of data mining? *The Edge*, 4(2).

Caruana, R. and Freitag, D. (1994). Greedy attribute selection. In *International Conference on Machine Learning*, pages 28–36.

Catbas, F., Ciloglu, S., Hasancebi, O., Grimmelsman, K., and Aktan, A. (2007). Limitations in structural identification of large constructed structures. *Journal of Structural Engineering*, 133(8):1051–1066.

Cattan, J. and Mohammadi, J. (1997). Analysis of bridge condition rating data using neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 12(11):419–429.

Chan, Z., Collins, L., and Kasabov, N. (2006). An efficient greedy k-means algorithm for global gene trajectory clustering. *Expert Systems with Applications*, 30(1):137–141.

Chantler, M., Coghill, G., Shen, Q., and Leitch, R. (1998). Selecting tools and techniques for model-based diagnosis. *Artificial Intelligence in Engineering*, 12(18):81–98.

Chapelle, O., Vapnik, V., Bousquet, O., and Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159.

Chaudhary, M., Abe, M., Fujino, Y., and Yoshida, J. (2000). Performance evaluation of two base-isolated bridges using seismic data. *Journal of Structural Engineering*, 116(10):1187–1195.

Chen, P., Buchheit, R., Garrett, J., and McNeil, S. (2005a). Web-vacuum: Web-based environment for automated assessment of civil infrastructure data. *Journal of Computing in Civil Engineering*, 19(2):137–147.

Chen, S., Wang, X., and Harris, C. (2005b). A search algorithm for global optimisation. *LNCS 2611*, pages 1122–1130.

Chen, Y.-W. and Lin, C. (2006). *Feature Extraction, Foundations and Applications*, chapter Combining SVMs with various feature selection strategies, pages 315–324. Springer.

Cheng, H., Chen, H., Jiang, G., and Yoshihira, K. (2007). Nonlinear feature selection by relevance feature vector machine. In Perner, P., editor, *Machine Learning and Data Mining in Pattern Recognition*, LNAI 4571, pages 144–159. Springer Verlag.

Cheung, Y.-M. (2005). On rival penalization controlled competitive learning for clustering with automatic cluster number selection. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1583–1588.

Chou, C., Su, M., and Lai, E. (2004). A new cluster validity measure and its application to image compression. *Pattern Analysis Applications*, 7(2):205–220.

Cleveland, H. (1982). Information as resource. *The Futurist*, pages 34–39.

Commault, C., Dion, J., and Agha, S. (2006). Structural analysis for the sensor location problem in fault detection and isolation. In *In Proceedings of IFAC Safeprocess*, Beijing, China.

Cooley, M. (1987). *Architecture or Bee?* The Hogarth Press.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press.

Culler, D. and Hong, W. (2004). Wireless sensor networks. *Communications of the ACM*, 47(6):32–33.

Darwiche, A. (2000). Model-based diagnosis under real-world constraints. *The AI Magazine*, 21(2):57–73.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1(3):131–156.

Davies, A. and Fearn, T. (2005). Back to basics: the principles of principal component analysis. *Spectroscopy Europe*.

Davies, D. and Bouldin, W. (1979). A cluster separation measure. *IEEE PAMI*, 1:224–227.

de Kleer, J. (2006). Improving probability estimates to lower diagnostic costs. In *17th International Workshop on Principles of Diagnosis (DX 06)*.

de Kleer, J. and Williams, B. (1987). Diagnosing multiple faults. *Artificial intelligence*, 32:97–130.

Dietterich, T. (2003). Machine learning. *Nature Encyclopedia of Cognitive Science*.

Ding, C. and He, X. (2004). K-means clustering via principal component analysis. In *Proceedings of the* 21^{st} *International Conference on Machine Learning*, ACM International Conference Proceeding Series, page 29. ACM Press.

Ding, Y. and Harrison, R. (2007). Relational visual cluster validity (rvcv). *Pattern Recognition Letters*, 28(15):2071–2079.

Domer, B., Raphael, B., Shea, K., and Smith, I. F. C. (2003). A study of two stochastic search methods for structural control. *Journal of Computing in Civil Engineering*, 17(3):132–141.

Dunn, J. (1974). Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4:95–104.

Edwards, A. (1984). *An introduction to Linear Regression and Correlation.* W.H. Freeman and Company, second edition edition.

Eliot, T. (1934). *The Rock.* Faber & Faber.

Engelman, L. and Hartigan, J. (1969). Percentage points of a test for clusters. *Journal of the American Statistical Association*, 64:1647–1648.

Evgeniou, T., Pontil, M., Papageorgiou, C., and Poggio, T. (2003). Image representations and feature selection for multimedia database search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):911–920.

Falkenhainer, B. and Forbus, K. (1991). Compositional modeling: Finding the right model for the job. *Artificial Intelligence*, 51:95–143.

Faltings, B. (2006). *Handbook of Constraint Programming*, chapter Distributed Constraint Programming, pages 699–729. Elsevier.

Famili, A., Liu, G., and Liu, Z. (2004). Evaluation and optimization of clustering in gene expression data analysis. *Bioinformatics*, 20(11):1535–1545.

Faschingbauer, G. and Scherer, R. (2007). Model and sensor data management for geotechnical engineering application. In *24th W78 Conference: Bringing ITC knowledge to work*.

Fayyad, U. and Uthurusamy, R. (2002). Evolving data mining solutions for insights. *Communications of the ACM*, 45(8):28–31.

Fernandez, R. and Viennet, E. (1999). Face identification using support vector machines. In *Proceedings of the European Symposium on Artificial Neural Networks*, pages 195–200.

Flach, P. (2001). On the state of the art in machine learning: a personal review. *Artificial Intelligence*, 131:199–222.

Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555.

Flintsch, G. and Chen, C. (2004). Soft computing applications in infrastructure management. *Journal of Infrastructure Systems*, 10(4):157–166.

Fraley, C. and Raftery, A. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588.

François, D. (2007). *High-dimensional data analysis: optimal metrics and feature selection.* PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, Belgium.

Frawley, W., Piatetsky-Shapiro, G., and Matheus, C. (1992). Knowledge discovery in databases: An overview. *AI Magazine*, 13(3):57–70.

Freyermuth, B. (1991). An approach to model based fault diagnosis of industrial robots. In *IEEE International Conference on Robotics and Automation.*

Frisk, E. and Krysander, M. (2007). Sensor placement for maximum fault isolability. In *18th International Workshop on Principles of Diagnosis (DX-07)*, pages 106–113, Nashville, USA.

Friswell, M. and Mottershead, J. (1995). *Finite Element Model Updating in Structural Dynamics.* Kluwer Academic Publishers.

Fröhlich, H., Chapelle, O., and Schölkopf, B. (2003). Feature selection for support vector machines by means of genetic algorithms. In *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 142–148.

Garatti, S., Bittanti, S., Liberati, D., and Maffezzoli, A. (2007). An unsupervised clustering approach for leukaemia classification based on dna micro-arrays data. *Intelligent Data Analysis*, 11(2):175–188.

Gold, C. and Sollich, P. (2002). Model selection for support vector machine classification. *ArXiv Condensed Matter e-prints.*

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Professional.

Gordon, A. (1996). *Data science, classification and related methods (eds. Hayashi, C. and Yajima, K. and Bock H.H. and Ohsumi, N. and Tanaka, Y. and Baba, Y.)*, chapter Cluster Validation, pages 22–39. Springer.

Greene, D. and Cunningham, P. (2006). Efficient prediction-based validation for document clustering. In *ECML*, pages 663–670.

Grossman, R., Kamath, C., Kegelmeyer, P., Kumar, V., and Namburu, R., editors (2001). *Data Mining for Scientific and Engineering Applications.* Kluwer Academic Publishers.

Guérif, S. and Bennani, Y. (2006). Selection of clusters number and features subset during a two-levels clustering task. In *Artificial Intelligence and Soft Computing.*

Guérif, S. and Bennani, Y. (2007). Dimensionality reduction through unsupervised features selection. In *International Conference on Engineering Applications of Neural Networks*, pages 98–106.

Guratzsch, R. and Mahadevan, S. (2006). Sensor placement design for shm under uncertainty. In *Third European Workshop on Structural Health Monitoring*, Granada, Spain.

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

Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L., editors (2006). *Feature Extraction, Foundations and Applications.* Series Studies in Fuzziness and Soft Computing. Springer.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002a). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002b). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422.

Hadid, A. and Pietikainen, M. (2004). Selecting models from videos for appearance-based face recognition. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 304–308.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2002a). Cluster validity methods: part 1. *SIGMOD Rec.*, 31(2):40–45.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2002b). Cluster validity methods: part 2. *SIGMOD Rec.*, 31(3):19–27.

Halkidi, M., Vazirgiannis, M., and Batistakis, I. (2000). Quality scheme assessment in the clustering process. In *Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, volume 1910 of *LNCS*, pages 265–267. Springer-Verlag.

Han, J. and Kamber, M. (2001). *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers.

Hand, D., Mannila, H., and Smyth, P. (2001). *Principles of Data Mining.* MIT Press.

Harding, J., Shahbaz, M., Srinivas, S., and Kusiak, A. (2006). Data mining in manufacturing: A review. *Journal of Manufacturing Science and Engineering*, 128(4):969–976.

Harris, J. (1984). Coronary angiography and its complications. the search for risk factors. *Archives of Internal Medicine*, 144(2).

He, X., Cai, D., and Niyogi, P. (2005). Laplacian score for feature selection. In *Proceedings of Neural Information Processing Systems.*

Hearst, M. (2006). Clustering versus faceted categories for information exploration. *Communications ACM*, 49(4):59–61.

Hermes, L. and Buhmann, J. (2000). Feature selection for support vector machines. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 712–715.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press.

Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2003). A practical guide to support vector classification. Technical report, National Taiwan University.

Huang, J., Cai, Y., and Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recogn. Lett.*, 28(13):1825–1844.

Huang, P.-W. and Liu, C.-L. (2006). Using genetic algorithms for feature selection in predicting financial distresses with support vector machines. In *IEEE International Conference on Systems, Man, and Cybernetics.*

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304.

Isermann, R. (1993). Fault diagnosis of machines via parameter estimation and knowledge processing: tutorial paper. *Automatica*, 29(4):815–835.

Jackson, J. (1991). *A user's guide to principal components.* Wiley.

Jain, A. and Dubes, R. (1988). *Algorithms for Clustering Data.* Prentice Hall.

Jain, A., Duin, R., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37.

Jain, A., Topchy, A., Law, M., and Buhmann, J. (2004). Landscape of clustering algorithms. In *Proceedings of the 17th International Conference on Pattern Recognition*, pages 260–263.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323.

Janik, W. and Fuchs, A. (1991). Process- and signal-model based fault detection of the grinding process. In *IFAC Symposium SAFEPROCESS.*

Jolliffe, I. (2002). *Principal Component Analysis.* Springer.

Jonas, J. and Haper, J. (2006). Effective counterterrorism and the limited role of predictive data mining. Policy Analysis. 584.

Jorgenson, D., Hunter, J., and Nadiri, M. (1970). The predictive performance of econometric models of quarterly investment behavior. *Econometrica*, 38(2):213–224.

Kantardzic, M. and Zurada, J., editors (2005). *Next Generation of Data-Mining Applications*, chapter Trends in data-mining applications : from research labs to fortune 500 companies. Wiley-IEEE Press.

Kaufman, L. and Rousseeuw, P. (1990). *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons.

Kim, M. and Ramakrishna, R. (2005). New indices for cluster validity assessment. *Pattern Recognition Letters*, 26(15):2353–2363.

Kim, M., Yoo, H., and Ramakrishna, R. (2004). Cluster validation for high-dimensional datasets. In *LNAI 3192*, pages 178–187. Springer-Verlag Berlin Heidelberg.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimisation by simulated annealing. *Science*, 220(4598):671–680.

Kleinberg, J. (2002). An impossibility theorem for clustering. In *16th conference on Neural Information Processing Systems*.

Köb, D. and Wotawa, F. (2004). Introducing alias information into model-based debugging. In *15th International Workshop on Principles of Diagnosis (DX 04)*.

Kohavi, R. and John, G. (1998). *Feature Selection for Knowledge Discovery and Data Mining*, chapter The Wrapper Approach, pages 33–50. Kluwer Academic Publishers.

Kohavi, R. and Sommerfield, D. (1995). Feature subset selection using the wrapper model: Overfitting and dynamic search space topology. In *The First International Conference on Knowledge Discovery and Data Mining*, pages 192–197. AAAI Press, Menlo Park, California.

Komosinski, M. and Krawiec, K. (2000). Evolutionary weighting of image features for diagnosing of cns tumors. *Artificial Intelligence in Medicine*, 19(14):25–38.

Koonce, D., Fang, C.-H., and Tsai, S.-C. (1997). A data mining tool for learning from manufacturing systems. In *Proceedings of the 21st international conference on Computers and industrial engineering*, pages 27–30, Essex, UK. Elsevier Science Publishers Ltd.

Korkmaz, E., Du, J., Alhajj, R., and Barker, K. (2006). Combining advantages of new chromosome representation scheme and multi-objective genetic algorithms for better clustering. *Intelligent Data Analysis*, 10(2):163 – 182.

Kothari, R. and Pitts, D. (1999). On finding the number of clusters. *Pattern Recognition Letters*, 20(4):405–416.

Kriegel, H.-P., Borgwardt, K., Kröger, P., Pryakhin, A., Schubert, M., and Zimek, A. (2007). Future trends in data mining. *Data Mining and Knowledge Discovery*, 15(1):87–97.

Krysander, M. and Frisk, E. (2007). Some theoretical results on sensor placement for diagnosis based on fault isolability specifications. Technical Report LiTH-R-2770, Department of Electrical Engineering.

Krysander, M. and Nyberg, M. (2002). Fault diagnosis utilizing structural analysis. In *CCSSE*, Sweden.

Kudo, M. and Sklansky, J. (2000). Comparison of algorithms that select features for pattern classiers. *Pattern Recognition*, 33:25–41.

Kumar, A. and Zhang, D. (2005). Biometric recognition using feature selection and combination. *LNCS 3546*, pages 813–822.

Kumar, V. (2003). Sensor: the atomic computing particle. *SIGMOD Rec.*, 32(4):16–21.

Kuonen, D. (2004). Data mining and statistics: What is the connection? The Data Administration Newsletter. http://www.tdan.com/view-articles/5226.

Kusiak, A. (2002). Data mining in decision making. In *SPIE Conference on Data Mining and Knowledge Discovery: Theory, Tools and Technology IV*, pages 155–165, Orlando, FL.

Kyrgyzov, I., Kyrgyzov, O., Maitre, H., and Campede, M. (2007). Kernel mdl to determine the number of clusters. In Perner, P., editor, *MLDM 2007*, LNAI 4571, pages 203–217. Springer-Verlag Berlin Heidelberg.

Lam, B. and Yan, H. (2005). A new cluster validity index for data with merged clusters and different densities. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 798– 803.

Lange, T., Roth, V., Braun, M., and Buhmann, J. (2004). Stability-based validation of clustering solutions. *Neural Computation*, 16:1299–1323.

Langley, P. (1996). *Elements of machine learning.* Morgan Kaufmann Publishers.

Langley, P. and Simon, H. (1995). Applications of machine learning and rule induction. *Communications of the ACM*, 38(11):54–64.

Lavrac, N., Motoda, H., and Fawcett, T. (2004). Editorial: Data mining lessons learned. *Machine Learning*, 57(7):5–11.

Lee, J. and Park, S. (2003). Agent and data mining based decision support system and its adaptation to a new customer-centric electronic commerce. *Expert Systems with Applications*, 25(17):619–635.

Lendasse, A., Lee, J., De Bodt, E., Wertz, V., and Verleysen, M. (2001). Dimension reduction of technical indicators for the prediction of financial time series - application to the bel20 market index. *European Journal of Economic and Social Systems*, 15(2):31–48.

Levy, M. and Salvadori, M. (2002). *Why Buildings Fall Down: How Structures Fail*. W. W. Norton & Company.

Lexin, L. and Hongzhe, L. (2004). Dimension reduction methods for microarrays with application to censored survival data. Technical report, Center for Bioinformatics & Molecular Biostatistics. http://repositories.cdlib.org/cbmb/surv2.

Li, D., Li, H., and Fritzen, C. (2006). On the physical significance of the norm based sensor placement method. In *Proceedings of the Third European Workshop on Structural Health Monitoring*, pages 1135–1143. DEStech publications, Inc.

Li, T., Zhu, S., and Ogihara, M. (2003). Algorithms for clustering high dimensional and distributed data. *Intelligent Data Analysis*, 7(4):305–326.

Li, Y., Dong, M., and Hua, J. (2008). Localized feature selection for clustering. *Pattern Recognition Letters*, 29(1):10–18.

Lin, S.-W., Tseng, T.-Y., Chen, S.-C., and Huang, J.-F. (2006). A sa-based feature selection and parameter optimization approach for support vector machine. In *IEEE International Conference on Systems, Man, and Cybernetics*.

Ling, R. (1972). On the theory and construction of k-clusters. *Computer Journal*, 15:326–332.

Liu, H. and Motoda, H., editors (1998). *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publishers.

Liu, P. and Chian, C. (1997). Parametric identification of truss structures using static strains. *Journal of Structural Engineering*, 123(7):927–933.

Liu, Y., Liao, X., Li, X., and Wu, Z. (2004). A tabu clustering algorithm for intrusion detection. *Intelligent Data Analysis*, 8(4):325–344.

Liu, Y. and Zheng, Y. (2006). Fs_sfs: A novel feature selection method for support vector machines. *Pattern Recognition*, 39(7):1333–1345.

Ljung, L. (1999). *System Identification - Theory For the User*. Prentice Hall.

Loughrey, J. and Cunningham, P. (2005). Using early stopping to reduce overfitting in wrapper-based feature weighting. Technical Report TCD-CS-2005-41, University of Dublin, Department of Computer Science.

Lovell, M. (1983). Data mining. *The Review of Economics and Statistics*, 65(1):1–12.

Lubasch, P., Schnellenbach-Held, M., Freischlad, M., and Buschmeyer, W. (2006). Knowledge discovery in bridge monitoring data: A soft computing approach. In *EG-ICE*, pages 428–436.

Luxburg, U., Bousquet, O., and Schölkopf, B. (2004). A compression approach to support vector model selection. *Journal of Machine Learning Research*, 5:293–323.

Mani, M. (2003). Understanding the semantics of sensor data. *SIGMOD Rec.*, 32(4):28–34.

Mao, K. (2004). Feature subset selection for support vector machines through discriminative function pruning analysis. *IEEE Transactions on systems, man, and cybernetics*, 34(1):60–67.

Martinez, A. and Kak, A. (2001). Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233.

Maulik, U. and Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions Pattern Analysis Machine Intelligence*, 24(12):1650–1654.

McNamara, J., Lanza di Scalea, F., and Fateh, M. (2004). Automatic defect classification in long-range ultrasonic rail inspection using a support vector machine-based 'smart system'. *Insight*, 46(6).

McQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press.

Melhem, H. and Cheng, Y. (2003). Prediction of remaining service life of bridge decks using machine learning. *Journal of Computing in Civil Engineering*, 17(1):1–9.

Menzies, T. and Hu, Y. (2003). Data mining for very busy people. *Computer*, 36(11):22–29.

Meo, M. and Zumpano, G. (2005). On the optimal sensor placement techniques for a bridge structure. *Engineering Structures*, 27:1288–1497.

Merz, C. and Murphy, P. (1996). *UCI* machine learning repository. http://www.ics.uci.edu/∼mlearn/MLSummary.html.

Milligan, G. and Cooper, M. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179.

Mitchell, T. (1997). *Machine learning*. McGraw-Hill.

Mitchell, T. (2006). The discipline of machine learning. Technical Report CMU-ML-06-108, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

Molina, L., Belanche, L., and Nebot, A. (2002). Feature selection algorithms: A survey and experimental evaluation. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*. IEEE Computer Society.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1-2):91–118.

Mukherjee, R. and Memik, S. (2006). Systematic temperature sensor allocation and placement for microprocessors. In *43rd ACM/IEEE Design Automation Conference*, pages 542–547.

Mullen, T., V., A., and D.L., H. (2006). Customer-driven sensor management. *IEEE Intelligent Systems*, 21(2):41–49.

Narendra, P. and Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Trans. Computers*, 26(9):917–922.

Nath, S. (2006). Crime pattern detection using data mining. In *IEEE/WIC/ACM International Conference on Web Intelligence and International Agent Technology*, pages 41–44.

Nguyen, H. and Chan, C. (1999). Applications of data analysis techniques for oil production prediction. *Artificial Intelligence in Engineering*, 13:257–272.

Oh, I.-S., Lee, J.-S., and Moon, B.-R. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1424–1437.

Ordonez, C. (2006). Integrating k-means clustering with a relational dbms using sql. *IEEE Transactions on Knowledge and Data Engineering*, 18(2):188–201.

Packhama, I., Rafiqb, M., Borthwickb, M., and Denham, S. (2005). Interactive visualisation for decision support and evaluation of robustness in theory and in practice. *Advanced Engineering Informatics*, 19:263–280.

Pal, N. and Jain, L., editors (2005). *Advanced Techniques in Knowledge Discovery and Data Mining*, chapter Preface. Springer.

Pal, S. and Mitra, P. (2004). *Pattern Recognition Algorithms for Data Mining.* CRC Press.

Pan, X., Ye, X., and Zhang, S. (2005). A hybrid method for robust car plate character recognition. *Engineering Applications of Artificial Intelligence*, 18(8):963–972.

Papadimitriou, C. (2005). Pareto optimal sensor locations for structural identification. *Computer Methods in Applied Mechanics and Engineering*, 194(12-16):1655–1673.

Papadimitriou, C., Beck, J., and Au, S. (2000). Entropy-based optimal sensor location for structural model updating. *Journal of Vibration and Control*, 6.

Parker, D., Frazier, W., Rinehart, H., and Cuevas, P. (2006). Experimental validation of optimal sensor placement algorithms for structural health monitoring. In *Proceedings of the Third European Workshop on Structural Health Monitoring*, pages 1144–1150. DEStech publications, Inc.

Pelleg, D. and Moore, A. (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In *Proc. 17th International Conference on Machine Learning*, pages 727–734. Morgan Kaufmann, San Francisco, CA.

Pena, J., Lozano, J., and Larranaga, P. (1999). An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 20(10):1027–1040.

Perner, P. (2006). Recent advances in data mining. *Engineering Applications of Artificial Intelligence*, 19(4):361–362.

Piatetsky-Shapiro, G. (1991). Knowledge discovery in real databases: A report on the ijcai-89 workshop. *AI Magazine*, 11(5):68–70.

Piatetsky-Shapiro, G. (2007). Data mining and knowledge discovery 1996 to 2005: overcoming the hype and moving from "university" to "business" and "analytics". *Data Mining and Knowledge Discovery*, 15(1):99–105.

Portinale, L., Magro, D., and Torasso, P. (2004). Multi-modal diagnosis combining case-based and model-based reasoning: a formal and experimental analysis. *Artificial Intelligence*, 158(2):109–153.

Posenato, D., Lanata, F., Inaudi, D., and Smith, I. F. C. (2006). Model free interpretation of monitoring data. In *EG-ICE*, LNAI 4200, pages 529–533.

Pudil, P., Novovičová, J., and Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125.

Pulinets, S., Gaivoronska, T., Leyva Contreras, A., and Ciraolo, L. (2004). Correlation analysis technique revealing ionospheric precursors of earthquakes. *Natural Hazards and Earth System Sciences*, 4:697–702.

Raghuraj, R., Bhushan, M., and Rengaswamy, R. (1999). Locationg sensors in complex chemical plants based on fault diagnostic observability criteria. *AIChE*, 45(2):310–322.

Rakotomamonjy, A. (2003). Variable selection using svm-based criteria. *Journal of Machine Learning Research*, 3:1357–1370.

Raphael, B. and Smith, I. F. C. (2003a). A direct stochastic algorithm for global search. *Journal of Applied Mathematics and Computation*, 146(2-3):729–758.

Raphael, B. and Smith, I. F. C. (2003b). *Fundamentals of Computer-Aided Engineering*. Wiley.

Raphael, B. and Smith, I. F. C. (2005). Engineering applications of a direct search algorithm, pgsl. In *Proceedings of the 2005 ASCE Computing Conference*. American Society of Civil Engineers. CDROM.

Raymer, M., Punch, W., Goodman, E., Kuhn, L., and Jain, A. (2000). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(2):164–171.

Reich, G. W. and Park, K. C. (2001). A theory for strain-based structural system identification. *Journal of Applied Mechanics*, 68(4):521–527.

Reich, Y. and Barai, S. (1999). Evaluating machine learning models for engineering problems. *Art. Int. in Eng.*, 13:257–272.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382.

Reunanen, J. (2006). *Feature Extraction, Foundations and Applications*, chapter Search Strategies, pages 119–136. Springer.

Robert-Nicoud, Y. (2003). *Une méthodologie mesures-modèles pour l'identification de systèmes de génie civil*. PhD thesis, EPFL, Lausanne, Switzerland.

Robert-Nicoud, Y., Raphael, B., Burdet, O., and Smith, I. F. C. (2005a). Model identification of bridges using measurement data. *Computer-Aided Civil and Infrastructure Engineering*, 20(2):118–131.

Robert-Nicoud, Y., Raphael, B., and Smith, I. F. C. (2000). Decision support through multiple models and probabilistic search. In *Proceedings of Construction Information Technology*, pages 765–779.

Robert-Nicoud, Y., Raphael, B., and Smith, I. F. C. (2004). Improving the reliability of system identification. *Next Generation Intelligent Systems in Engineering*, 4(199):100–109.

Robert-Nicoud, Y., Raphael, B., and Smith, I. F. C. (2005b). Configuration of measurement systems using shannon's entropy function. *Computers and structures*, 83(8-9):599–612.

Robert-Nicoud, Y., Raphael, B., and Smith, I. F. C. (2005c). System identification through model composition and stochastic search. *Journal of Computing in Civil Engineering*, 19(3):239–247.

Sakai, T., Imiya, A., Komazaki, T., and Hama, S. (2007). Critical scale for unsupervised cluster discovery. In Perner, P., editor, *MLDM 2007*, LNAI 4571, pages 218–232. Springer-Verlag Berlin Heidelberg.

Salem, S. and Nandi, A. (2005). New assessment criteria for clustering algorithms. In *2005 IEEE Workshop on Machine Learning for Signal Processing*, pages 285–290.

Salvo, A. (2006). Ponts de robert maillart. Technical report, EPFL-MCS, Lausanne, Switzerland.

Sanayei, M., Imbaro, G., McClain, J., and Brown, L. (1997). Structural model updating using experimental static measurements. *Journal of Structural Engineering*, 123(6):792–798.

Sanchez-Marre, M., Gilbert, K., Sojda, R., Steyer, J., Struss, P., and Rodríguez-Roda, I. (2006). Uncertainty management, spatial and temporal reasoning and validation of intelligent environmental decision support systems. In *Proceedings of the iEMSs Third Biennial Meeting: "Summit on Environmental Modelling and Software"*, Burlington, Vermont, USA. International Environmental Modelling and Software Society.

SanJuan, E. and Ibekwe-SanJuan, F. (2006). Text mining without document context. *Inf. Process. Manage.*, 42(6):1532–1552.

Saunders, C., Gammerman, A., Brown, H., and Donald, G. (2000). Application of support vector machines to fault diagnosis and automated repair. In *In Proceedings of Eleventh International Workshop on Principles of Diagnosis (DX '00).*

Schnalzer, R., Reda Taha, M., McCuskey, M., Quintana, S., and Camp, J. (2006). Identifying bridge performance patterns in a bridge inventory database: An analytical investigation. In

Proceedings of Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, Canada. CD-ROM.

Schulte, R., Bohle, K., Fritzen, C., and Schuhmacher, G. (2006). Optimal sensor placement for damage identification - an efficient forward-backward selection algorithm. In *Proceedings of the Third European Workshop on Structural Health Monitoring*, pages 1151–1159. DEStech publications, Inc.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Sempere, J. and Lopez, D. (2003). Learning decision trees and tree automata for a syntactic pattern recognition task. *Pattern Recognition and Image Analysis*, pages 943–950.

Shannon, C. and Weaver, W. (1949). *The Mathematical Theory of Communication.* University of Illinois Press.

Sharma, N. (2005). The origin of the data information knowledge wisdom hierarchy. School Of Information, University of Michigan.

Shirazi Kia, S., Noroozi, S., Carse, B., and Vinney, J. (2005). Application of data mining techniques in predicting the behaviour of composite joints. In *The Eighth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering.* Civil-Comp Press. CDROM.

Simek, K., Fujarewicz, K., Swierniak, A., Kimmel, M., Jarzab, B., Wiench, M., and Rzeszowska, J. (2004). Using svd and svm methods for selection, classification, clustering and modelling of dna microarray data. *Engineering Applications of Artificial Intelligence*, 17:417–427.

Smith, L. (2002). A tutorial on principal components analysis.

Smyth, P. (2000). Data mining: Data analysis on a grand scale? Technical Report CA 92697-3425, Information and Computer Science, University of California.

Soares, C., Brazdil, P., and Kuba, P. (2004). A meta-learning method to select the kernel width in support vector regression. *Machine Learning Journal*, 54(3):195–209.

Soibelman, L. and Kim, H. (2002). Data preparation process for construction knowledge generation through knowledge discovery in databases. *Journal of Computing in Civil Engineering*, 16(1):39–48.

Spiegler, I. (2003). Technology and knowledge: bridging a "generating" gap. *Information and Management*, 40(7):533–539.

Stalker, R. and Smith, I. F. C. (2002). Structural monitoring using engineer-computer interaction. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 16:203–218.

Steinbauer, G. and Wotawa, F. (2005). Detecting and locating faults in the control software of autonomous mobile robots. In *16th International Workshop on Principles of Diagnosis (DX 05)*.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, 36(2):111–147.

Struss, P. (2006). *Fault Detection, Supervision and Safety of Technical Processes*, chapter A model-based methodology for the integration of diagnosis and fault analysis during the entire life cycle. Elsevier.

Struss, P. (2007). *to appear in the Handbook of Knowledge Representation*, chapter Model-based Problem Solving. Elsevier.

Struss, P. and Price, C. (2004). Model-based systems in the automotive industry. *AI Magazine*, 24(4):17–34.

Svanerudh, P., Raphael, B., and Smith, I. F. C. (2002). Lowering costs of timber shear-wall design using global search. *Engineering with Computers*, 18(2):93–108.

Tan, P.-N., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Addison Wesley.

Tenenhaus, A., Giron, A., Viennet, E., Béra, M., Saporta, G., and Fertil, B. (2007). Kernel logistic pls: A tool for supervised nonlinear dimensionality reduction and binary classification. *Computational Statistics and Data Analysis*, 51(9):4083–4100.

Theodoridis, S. and Koutroumbas, K. (1999). *Pattern Recognition*. Academic Press.

Tibshirani, R. and Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational & Graphical Statistics*, 14(18):511–528.

Tibshirani, R., Walther, G., and Hastie, T. (2000). Estimating the number of clusters in a dataset via the gap statistic. Technical Report 208, Dept. of Statistics, Stanford University.

Travé-Massuyès, L., Escobet, T., and Olive, X. (2006). Diagnosability analysis based on component-supported analytical redundancy relations. *IEEE Transaction on Systems, Man, and Cybernetics: Part A*, 36(6):1146–1160.

Travé-Massuyès, L., Ironi, L., and Dague, P. (2003). Mathematical foundations of qualitative reasoning. *AI Magazine*, 24(4):91–106.

Tuomi, I. (1999). Data is more than knowledge: implications of the reversed knowledge hierarchy for knowledge management and organizational memory. In *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences*.

Vafaie, H. and Imam, I. (1994). Feature selection methods: Genetic algorithms vs. greedy-like search. In *Proceedings of the International Conference on Fuzzy and Intelligent Control Systems*.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer.

Vesanto, J. and Alhoniemi, E. (2000). Clustering of the selforganizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600.

Webb, A. (2002). *Statistical Pattern Recognition*. Wiley.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2001). Feature selection for svms. In *Advances in Neural Information Processing Systems*.

Weston, J. and Watkins, C. (1998). Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London.

Wilder, T. (2004). *The Bridge of San Luis Rey: A Novel*. HarperCollins.

Wilking, D. and Roefer, T. (2004). Realtime object recognition using decision tree learning. In *RoboCup*, volume 3276 of *Lecture Notes in Artificial Intelligence*, pages 556–563. Springer.

Witten, I. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers.

Worden, K. and Burrows, A. (2001). Optimal sensor placement for fault detection. *Engineering Structures*, 23:885–901.

Wu, S. and Chow, T. (2004). Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density. *Pattern recognition*, 37(2):175–188.

Wu, X., Kumar, V., Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., Zhou, Z.-H., Steinbach, M., Hand, D., and Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14:1–37.

Wu, Z. and Li, C. (2006). *Feature Extraction, Foundations and Applications*, chapter Feature selection with transductive support vector machines, pages 325–341. Springer.

Xiong, N. and Svensson, P. (2002). Multi-sensor management for information fusion: issues and approaches. *Information Fusion*, 3(2):163–186.

Xu, L., Yan, Y., Cornwell, S., and Riley, G. (2005). Online fuel tracking by combining principal component analysis and neural network techniques. *IEEE Transactions on Instrumentation and Measurement*, 54(4):1640–1645.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678.

Yan, L., Fraser, M., Oliver, K., Elgamal, A., Conte, J., and Fountain, T. (2005). Traffic pattern recognition using an active learning neural network and principal components analysis. In *The Eighth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering.* Civil-Comp Press. CDROM.

Yang, J. and Honavar, V. (1998). *Feature Selection for Knowledge Discovery and Data Mining*, chapter Feature subset selection using a genetic algorithm, pages 117–136. Kluwer Academic Publishers.

Yang, X., Cao, A., and Song, Q. (2006). A new cluster validity for data clustering. *Neural Processing Letters*, 23(3):325–344.

Younis, M., Akkaya, K., and Kunjithapatham, A. (2003). Optimization of task allocation in a cluster-based sensor network. In *Proceedings of the Eighth IEEE International Symposium on Computers and Communication*, volume 1, pages 329– 334.

Yuan, X., Yuan, X., Yang, F., Peng, J., and Buckles, B. (2003). Gene expression classification: Decision trees vs. svms. In *FLAIRS Conference*, pages 92–97.

Yun, C.-B., Yi, J.-H., and Bahng, E. (2001). Joint damage assessment of framed structures using a neural networks technique. *Engineering Structures*, 23:425–435.

Zadeh, L. (2001). Applied soft computing - foreword. *Applied Soft Computing Journal*, pages 1–2.

Zeleny, M. (1987). Management support systems: Towards integrated knowledge management. *Human Systems Mangement*, 7(1):59–70.

Zhou, L., Shi, Y., Feng, J., and Sears, A. (2005). Data mining for detecting errors in dictation speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(5):681–688.

Zhuang, D., Zhang, B., Yang, Q., Yan, J., Chen, Z., and Chen, Y. (2005). Efficient text classification by weighted proximal svm. In *Fifth IEEE International Conference on Data Mining*, pages 538–545.

Zins, C. (2006). Redefining information science: from information science to knowledge science. *Journal of Documentation*, 62(15):447–461.

Zins, C. (2007). Conceptual approaches for defining data, information, and knowledge. *Journal of the American Society for Information Science and Technology*, 58(4):479–493.

Zongker, D. and Jain, A. (1996). Algorithms for feature selection: An evaluation. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 2, pages 18–22.

Index

Publications

2008

- Saitta S., Kripakaran P., Raphael B. and Smith I.F.C., “Improving System Identification Using Clustering”, accepted for publication in the Journal of Computing in Civil Engineering, 2008.

- Saitta S., Raphael B. and Smith I.F.C., “A Comprehensive Validity Index for Clustering”, accepted for publication in the Journal of Intelligent Data Analysis, 2008.

- Smith I.F.C. and Saitta S., “Improving Knowledge of Structural System Behavior through Multiple Models”, accepted for publication in the Journal of Structural Engineering, 2008.

2007

- Raphael B., Domer B., Saitta S. and Smith I.F.C., “Incremental development of CBR strategies for computing project cost probabilities”, Advanced Engineering Informatics, Vol 21, No 3, pp. 311-321, 2007.

- Kripakaran P., Saitta S., Ravindran S. and Smith I.F.C., “Measurement System Design Using Damage Scenarios”, Computing in Civil Engineering, Proceedings of the 2007 International Workshop on Computing in Civil Engineering (ASCE), L. Soibelman and B. Akinci (Ed.), pp. 615-623, 2007.

- Saitta S., Raphael B. and Smith I.F.C., “A Bounded Index for Cluster Validity”, In: P. Perner (Ed.), Machine Learning and Data Mining in Pattern Recognition, LNAI 4571, Springer Verlag, Heidelberg, pp. 174-187, 2007.

- Kripakaran P., Saitta S., Ravindran S. and Smith I.F.C., "System Identification: Data Mining to Explore Multiple Models", Proceedings of 3rd Conf. on Structural Health Monitoring and Intelligent Infrastructure, ISHMII, Winnipeg, Canada, CDROM, 2007.

- Kripakaran P., Saitta S., Ravindran S. and Smith I.F.C., "Optimal Sensor Placement for Damage Detection: Role of Global Search", 1st International Workshop on Decision Support in Structural Health Monitoring, Proceedings of the 18th International Conference on Database and Expert System Applications (DEXA), pp. 302-306, 2007.

2006

- Saitta S., Raphael B. and Smith I.F.C., "Combining Two Data Mining Methods for System Identification", EG-ICE, LNAI 4200, pp. 606-614, 2006.

- Domer B., Raphael B. and Saitta S., "KnowPrice2: Intelligent Cost Estimation for Construction Projects", EG-ICE, LNAI 4200, pp. 147-152, 2006.

- Saitta S., Raphael B. and Smith I.F.C., "Data Mining for Decision Support in Multiple-Model System Identification", 2nd WSEAS International Symposium on Data Mining, Portugal, CD-ROM, 2006.

- Smith I.F.C. and Saitta S., "Multiple-Model Updating to Improve Knowledge of Structural System Behavior", 17th Analysis and Computation Specialty Conference, American Society of Civil Engineers, Reston VA, USA, CDROM, 2006.

- Saitta S., Raphael B. and Smith I.F.C., "Rational design of measurement systems using information science", Responding to Tomorrow's Challenges in Structural Engineering, Proceedings of IABSE Conference in Budapest, IABSE Report 92, 2006.

- Smith I.F.C., Saitta S., Ravindran S. and Kripakaran P., "Challenges of data interpretation", Proceedings 18th SAMCO Workshop, Prague, pp. 37-57, 2006.

2005

- Saitta S., Raphael B. and Smith I.F.C., "Data mining techniques for improving the reliability of system identification", Advanced Engineering Informatics, Vol 19, No 4, pp. 289-298, 2005.

- Saitta S., Raphael B. and Smith I.F.C., "Supporting Engineers during System Identification", Computing in Civil Engineering, Proceedings of the ASCE Computing Conference, American Society of Civil Engineers, Reston VA, USA, CDROM, 2005.

- Saitta S., Raphael B. and Smith I.F.C., "Data Mining for System Identification Support", 12th EG-ICE Workshop, Poland, CD-ROM, 2005.

Sandro Saitta

Computer Science Engineer, PhD Student

Avenue Dapples 19
1006 Lausanne
Switzerland
26 years old, Swiss, Single

Mobile: ++41 (0)78 640 41 48
Email: sandro.saitta@gmail.com
Web: www.saittaweb.com
Blog: www.dataminingblog.com

DOMAINS OF INTEREST

Data mining, business intelligence, decision support, search engine optimization

EDUCATION

- **Ph.D. in Computer Science**, Ecole Polytechnique Fédérale de Lausanne (EPFL) (2004-present)
  Development of data mining methodologies for engineering applications.

- **M.Sc. in Computer Science**, Ecole Polytechnique Fédérale de Lausanne (EPFL) (1999-2004)
  Master thesis on prediction of airborne pollen in collaboration with MeteoSwiss.

- **High School Diploma**, Gymnase du Cessnov, Yverdon (2001-2004)
  Specialization in mathematics.

INDUSTRY EXPERIENCE AND COLLABORATIONS

- **Cost estimation project for Tekhne Management S.A.** (2004 – 2007)

  Case based reasoning techniques are used to predict project costs in the construction industry. Within the time and budget allocated, a reliable cost estimation software has been developed.

- **Prediction of airborne pollen for MeteoSwiss** (2003 – 2004)

  Several data mining techniques were experimented for predictive purposes regarding pollen concentration in Switzerland. Results obtained during the predefined time period showed an improvement of 10% over traditional methods employed by MeteoSwiss.

- **Internship on a data warehouse project at BKW** (2003)

  For two months, I was involved in a project related to reporting and data warehousing at BKW (Bernische KraftWerke) in Bern. During this internship, I was introduced to technologies such as OLAP and SAP.

ACADEMIC EXPERIENCE

- **Ph.D. in data mining for decision support in engineering** (2004 – present)

  Data mining techniques are applied to extract knowledge in engineering system identification. Results show a high potential for decision support using methods such as clustering and feature selection. Additional contributions in the domain of sensor placement are provided using probabilistic search techniques.

- **Website development for the University of Geneva** (2004)

  A website presenting a text processing software was created in collaboration with Geneva University. The aim was to introduce the software to users. Challenges included listening and understanding customer's needs while supporting them in a domain they didn't know.

- **Project on updating a Java program** (2003)

  After a successful semester project, I was invited to extend my work by updating an existing program that performed French language processing. Understanding the code and adding new functionalities while preserving software stability were the main challenges.

SKILLS

Programming: Java, C++, MATLAB, HTML, XML, LaTeX, VB, Delphi, Lisp, Prolog
Software: Office, JBuilder, NetBeans, Emacs
Operating systems: Windows/Linux
Languages: French (native), English (fluent), German (intermediate), Italian (intermediate)

ACADEMIC ACTIVITIES

Reviewer for various journals and conferences (2006 – present)
Principal teaching assistant in a computer science course (2004 – present)
Student supervisor on a computer science project (2006 – 2007)

EXTRA-CURRICULAR ACTIVITIES

Blogging
Blog about data mining that receives more than 1300 visits each month.

Personal interests
Search engine optimization (SEO) and website marketing.

Sports and fun
Tennis player in a team (interclub competition), ski with friends, reading/writing novels.

REFERENCES

Prof. Ian F.C. Smith, Lab. Director, ++41 (0)21 693 52 42, ian.smith@epfl.ch
Dr. François Fleuret, M.Sc. Diploma Supervisor, ++41 (0)27 721 77 39, fleuret@idiap.ch
Dr. Bernd Domer, Industry contact, ++41 (0)58 787 87 44, bernd.domer@iss.ch