Stellingen

behorende bij het proefschrift
Fuzzy Probabilistic Learning and Reasoning
door G.C. van den Eijkel

1. De vage kansrekening, gebaseerd op "de kans op een vage gebeurtenis", is:

   a. een onzekerheidscalculus waarin vaagheid en willekeurigheid beide bestaan,

   b. een basis voor een efficiënte niet-parametrische kansdichtheidsschatter,

   c. een generalisatie van de kansrekening,

   d. een specificatie van de vage logica,

   en is daarmee een bruikbare synthese van de vage logica en de kansrekening.

   Dit proefschrift, Hoofdstuk 4

2. Leren is niet alleen het verminderen van de beslisonzekerheid maar ook het verminderen van de informatie welke nodig is om het geleerde zinvol te representeren.

   Dit proefschrift, Hoofdstuk 5

3. Redeneren met regels volgens de vage kansrekening leidt tot genuanceerde uitspraken waarvoor toch een algemene uitleg kan worden gegeven.

   Dit proefschrift, Hoofdstuk 5 en 6

4. De Quantummechanica staat de interpretatie toe dat vooralsnog de natuur van continue aard is maar zich tot nog toe discreet naar ons opstelt; dit impliceert dat het kleinste deeltje nooit gevonden zal worden.

5. Als wetenschap is de Artificial Intelligence al even ijdel als de Geneeskunde omdat intelligentie, evenals geneeskracht, niet het vermogen is van een recept maar van een zelforganiserend proces.

6. De Stirling-motor, ofschoon eerder ontwikkeld, is geavanceerder dan de zonnecel.

7. Naar de meest waardevolle informatie in een publicatie, de tijdgeest, kan meestal niet worden verwezen omdat hij tussen de regels rondwaart.

8. Wetenschap is de mythische voortzetting van onze mystieke ervaringen. Zij is dus niet meer of minder waar dan andere religies.

9. Het rekeningrijden zal, tengevolge van het doorberekenen van de rekening in goederen en diensten, het toonbeeld worden van een Haagse mop: "maximaal overhead project".

10. De wet Modernisering Universitaire Bestuursorganisatie (MUB) resulteert in een tragikomedie op alle bestuurlijke niveaus; je wordt er nooit echt bij betrokken maar je ogen tranen van het lachen.

11. De zitzak illustreert dat wrijving een passende indruk achterlaat.

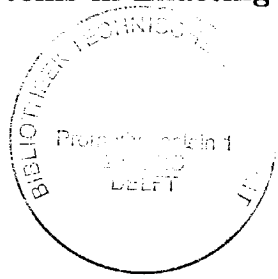
12. Diepgang komt uit de breedte.

Fuzzy Probabilistic Learning and Reasoning

Rule Induction for Decision-Support Systems in Exacting Environments

Fuzzy Probabilistic Learning and Reasoning

Rule Induction for Decision-Support Systems in Exacting Environments



Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College voor Promoties aangewezen,
op maandag 18 januari 1999 te 13:30 uur

door

Gerard Cornelis VAN DEN EIJKEL

technisch natuurkundig ingenieur
geboren te Alphen aan den Rijn

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. E. Backer.

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. E. Backer,	Technische Universiteit Delft, promotor
Dr.ir. J.C.A. van der Lubbe,	Technische Universiteit Delft, toegevoegd promotor
Prof.dr.mr.dr. B.A.M. de Mol,	Academisch Medisch Centrum Amsterdam & Technische Universiteit Delft
Prof.ir. H.B. Verbruggen,	Technische Universiteit Delft
Prof.dr. H. Koppelaar,	Technische Universiteit Delft
Prof.dr. A.K. Jain,	Michigan State University, USA
Dr.ir. R.P.W. Duin,	Technische Universiteit Delft

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

van den Eijkel, Gerard Cornelis

Fuzzy Probabilistic Learning and Reasoning/
G.C. van den Eijkel.- Delft : Technische Universiteit Delft,
Faculteit der Informatietechnologie en Systemen. - Ill.
Thesis Technische Universiteit Delft. - With ref. - With summary in Dutch.
ISBN 90-407-1805-9
Subject headings: AI/Pattern Recognition/Machine Learning/Fuzzy Sets/Fuzzy Logic.

Published and distributed by:
Delft University Press
Mekelweg 4
2628 CD Delft
The Netherlands
Telephone: +31 (0)15-2783254
Telefax: +31 (0)15-2781661
E-mail: DUP@DUP.TUDelft.NL

Aan Carolien
Aan mijn ouders

Contents

Summary xi

1 Introduction 1
- 1.1 Learning, Reasoning, and Uncertainty . . . 3
  - 1.1.1 Data, Information, and Knowledge . . . 3
  - 1.1.2 Uncertainty . . . 5
- 1.2 Synthesis . . . 6
- 1.3 Outline . . . 8

2 Rule Induction 11
- 2.1 Main Problem: Data Fit vs. Mental Fit . . . 12
- 2.2 Elements of Rule Induction . . . 15
  - 2.2.1 Domain Representation . . . 16
  - 2.2.2 Knowledge Representation . . . 17
  - 2.2.3 Search Process . . . 19
  - 2.2.4 Classification Method . . . 22
- 2.3 Information-Theoretic Approach . . . 23
  - 2.3.1 Information and Uncertainty . . . 23
  - 2.3.2 Example of Tree Induction . . . 25
  - 2.3.3 Information-Theoretic Rule Induction . . . 28
- 2.4 Conclusion . . . 32

3 Kernel-Based Density Estimation 33
- 3.1 Parzen Windows . . . 35
- 3.2 Double-Kernel Estimator . . . 38
  - 3.2.1 Basic Principle . . . 38
  - 3.2.2 Analysis . . . 41
  - 3.2.3 Choosing the Kernels and their Distribution . . . 46
- 3.3 Non-Uniform Distributions . . . 48
- 3.4 Experiments . . . 55
  - 3.4.1 Uniform vs. Non-uniform DK estimator . . . 55
  - 3.4.2 r-Optimal Non-uniform DK Estimator . . . 57
  - 3.4.3 Classification on Sparse Data Sets . . . 57
- 3.5 Conclusion . . . 61

4 Fuzzy Probability **63**
4.1 Knowledge and Uncertainty 64
4.1.1 On Knowledge 64
4.1.2 On Certainty 66
4.2 Types of Uncertainty 66
4.2.1 Probability 66
4.2.2 Fuzziness 69
4.3 A Framework for Fuzzy Probability 72
4.3.1 Considerations for Co-existence 73
4.4 Fuzzy Probabilistic Algebra 74
4.5 Fuzzy Probabilistic Logic 78
4.6 Derivation of the Double-Kernel Estimator 79
4.7 Classification using Fuzzy Probability 80
4.7.1 Example 82
4.8 Conclusion 83

5 Fuzzy Probabilistic Rule Induction **87**
5.1 Overview of FPI 89
5.2 Reference Frame 90
5.3 Information and Knowledge 91
5.4 Generalization 94
5.4.1 Hypothesis Generation 96
5.4.2 Non-Disjoint Rule Induction 97
5.4.3 Disjoint Rule Induction 98
5.5 Classification and Explanation 100
5.6 Coping with High Dimensionality 102
5.7 Experimental Results 103
5.7.1 Experimental Setup 103
5.7.2 Data Sets 104
5.7.3 Results 105
5.8 Conclusion 110

6 The Intelligent Anesthesia Monitor **111**
6.1 Monitor Design 112
6.2 Analysis Stage 115
6.2.1 Alarms vs. Triggers 115
6.2.2 Learning Triggers 116
6.3 Feasibility Study 117
6.3.1 Goal 117
6.3.2 Fuzzy Sets 118
6.3.3 Rule Induction Method 118
6.3.4 Data Acquisition 119
6.3.5 Results and Conclusions 120
6.4 Case Study 121
6.4.1 Database 122
6.4.2 Selection of Interventions and Features 122

6.4.3 Selection of Examples 124
6.4.4 Results 124
6.4.5 Feature Selection 126
6.4.6 Performance on the Remaining Operations 127
6.4.7 Preliminary Conclusion 128
6.4.8 Discussion with an Expert Panel 129
6.5 Conclusion 130

7 Discussion **131**
7.1 Reflections 132
7.2 Further Extensions 133
7.3 Future Research 134

Bibliography **137**

A Proof of Convergence **147**

B A Generalized Class of DK estimators **153**

C Allowed Kernels **159**

D Order Independence **165**

E MISE Analysis **167**

F Proof of Equivalence **169**

G Example Rule Base **171**

Leren en Redeneren op basis van de Vage Kansrekening **175**

Acknowledgments **177**

Curriculum Vitae **179**

Summary

This thesis deals with the problem of knowledge acquisition for decision-support systems in exacting environments. In exacting environments it is necessary to obtain explicit knowledge by which the right decisions can be made *and* explained. Such knowledge can be used in a decision-support system in order to improve the decision-making process of experts. An example of an exacting environment is patient monitoring in anesthesia, which has been subject of research in the Intelligent Anesthesia Monitor project of the Delft University of Technology in collaboration with the Academic Medical Center in Amsterdam. This project has stimulated much of the work which is presented in this thesis.

In this thesis knowledge is acquired by learning a set of rules from examples. It is argued that the result of learning should be a rule base that fits both the actual data (data fit) and the user's frame of reference (mental fit). The reason is that both aspects are necessary in order to make and explain the right decisions. To this purpose a synthesis is made between probability density function estimation, which has an emphasis on data fit, and fuzzy rule induction, which has an emphasis on mental fit. To realize this synthesis a general framework for uncertainty calculus is developed: the fuzzy probabilistic framework. The fuzzy probabilistic framework is based on the probability of a fuzzy event, and is highly suitable for learning and reasoning with uncertainty. This framework is one of the main contributions of this thesis.

One of the validations for the fuzzy probabilistic framework is that a new and efficient kernel-based density estimator can be derived: the double-kernel estimator. It is shown how this estimator is mathematically related to the well-known Parzen Windows technique. Experiments show that in decision problems (e.g. classification) the double-kernel estimator can obtain a higher accuracy with fewer kernels than the Parzen Windows technique. The double-kernel estimator is one of the accessory contributions of this thesis.

Another main contribution of this thesis is a new rule induction algorithm: fuzzy probabilistic rule induction. This algorithm, based on the fuzzy probabilistic framework, follows the covering paradigm in rule induction. The rules are selected on the basis of the J-information measure, which is closely related to the mutual information used in decision trees. Experiments, in which an implementation of this algorithm was used called FILER, show that FILER can obtain highly accurate classifications in comparison with other algorithms. Further, these classifications can be explained by using only a small number of

general rules. The remaining problem is that the covariance of the data cannot be taken into account in the generalized rules. Without generalization the covariance can be taken into account, but in that case the fuzzy probabilistic rule induction degenerates to the double-kernel technique.

The final contribution of this thesis is the application of fuzzy probabilistic rule induction in anesthesia monitoring. Anesthesia monitoring is an example of an exacting environment where many sources of complex information have to be processed in a relatively short time. A complicating factor in anesthesia monitoring is the time-varying nature of the physiological signals. The approach followed in this thesis is representing the changes in time by several trend parameters. On the basis of these trends and other features rules can be learned from examples of interventions ("alarm" situations) given by anesthetists. With these rules a decision-support system can reason in such a way that it can (1) trigger the anesthetists, and (2) explain the cause for such a trigger. A case study is presented where on the basis of about a thousand examples, obtained with permission of the University of Groningen, a rule base of about 40 rules was generated. Using cross-validation, it was estimated that the rule base could recognize almost 80% of the unobserved examples correctly, and give a meaningful explanation as well. An expert panel confirmed that this would be the expected performance of an anesthetist, and agreed with many of the general rules that had been induced. On the basis of these results it is concluded that fuzzy probabilistic rule induction is useful for a decision-support system in anesthesia. However, the performance of the system ultimately depends on the quality of the examples provided by an expert.

Chapter 1

Introduction

In everyday life countless decisions need to be made. Often some source of information can be used to make a rational, well-considered decision. Suppose we had to decide whether we should travel by car or by bike. We could simply flip a coin to make an irrational decision or we could listen to the traffic announcement on the radio to make a more rational decision. For decision problems like classification, estimation, prediction, forecasting, and control, we often need to rely on experience, or examples, to make the right decision. More often than not such experience manifests itself as an "intuition", or a "gut-feeling" that inclines us towards a particular decision. This typically seems to occur in exacting environments: (demanding) environments where multiple sources of complex information need to be processed in relatively short time. As many of our most profound abilities that partially take place on a subconscious level as well such as walking, talking, object recognition, and learning, intuition is unquestionably useful for arriving at a decision. However, as we are a species driven by curiosity and capable of communication and reasoning, we find it hard to justify decisions by something so inexplicable and implicit as intuition. Especially if others make a decision that is contradictory to the decision that we ourselves would make, we require an explanation for further discussing or reasoning. This thesis is devoted to the development of a system that can support the decision-making process in exacting environments by suggesting an appropriate decision, which can also be explained to the expert.

An example of an exacting environment that has stimulated the work in this thesis is patient monitoring by anesthetists during surgery. Exacting environments are characterized by decision-making processes where:

- multiple sources of numerical information need to be processed,
- information changes rapidly,
- peoples' lives are at stake,
- multiple interdependent decision levels are required to arrive at the final decision,

- specialized terminology (reference frame) and knowledge guide communication and reasoning,

- justification of decisions may be required by legal authorities.

Other examples include monitoring patients at the intensive care, condition monitoring in the industry, system control by operators in chemical or nuclear plants, and to a certain extent also financial management and marketing for large industries. In such environments it is imperative that, on the basis of the available information, a decision-support system makes the right decision *and* is expressive. The latter means that the system should be able to explicitly state the reason(s) for, and the uncertainty related to the decision. It should be noted that such a system does not replace our decision-making process. Quite often, not all the information is available to the system. However, what it should do is make maximum use of the available information to suggest the right decision, accompanied by an explanation. In exacting environments, systems like these can support our lower cognitive and often time-consuming tasks like data and information processing that are necessary for making rational decisions.

Apart from clarifying an intuitive decision-making process (improving the quality of the decision), there is a variety of reasons for using decision-support systems:

- to reduce human errors due to fatigue, distraction or stress,

- to increase the work flow,

- to support non-expert decision makers,

- to educate and train (non-)experts,

- to increase the consistency of the decision making,

to name but a few.

The general problem in developing decision-support systems is the acquisition of the right knowledge. Usually the decision-making process is so complex and requires so much domain knowledge that it cannot be modeled by a set of equations like the ones used in modeling physical processes. Further, the knowledge of expert decision makers comes from years of experience and is not easily acquired due to its implicit and intuitive nature. In literature on knowledge-based systems, this problem is known as the knowledge-acquisition bottleneck. Early solutions to this problem were based on interviewing experts by using formal knowledge structures, like the well-known KADS-system [16]. However, in the 90's it became clear that the system itself should actually learn from experience, see for example [99]. In this thesis we adopt this view and focus on the process of learning in order to acquire useful knowledge for decision-support systems.

1.1 Learning, Reasoning, and Uncertainty

Learning from experience plays an important role in tasks like walking, talking, etc.. It comes naturally to many beings and it is sometimes done almost "mindlessly". We sometimes learn without knowing that we learn, and only afterwards we notice a change, an improvement, in our behavior. If we are aware that we have learned something, then we often do not know exactly what and cannot find the words to express it. It is therefore surprising that we even can learn, to a certain extent, how to learn.

To make systems, i.e. computers, that learn from experience, it is necessary to make the learning process explicit in an algorithm of some kind. Before turning to a general paradigm for such algorithms, we will have to clarify the terms learning, reasoning, and uncertainty.

Learning from experience is known (in computer science) as *inductive learning*, sometimes also denoted as learning from examples. Other types of learning exist such as learning by being told, learning by analogy or analytical learning (theorem proving), but these are outside the scope of this thesis. The product of learning is knowledge, which is necessary for the decision-making process. This knowledge is used to obtain a decision for instances (cases) that we have not experienced before. This use of knowledge is known (in computer science) as *deductive reasoning*. We will usually refer to inductive learning and deductive reasoning as learning and reasoning, respectively.

1.1.1 Data, Information, and Knowledge

In Figure 1.1 we have visualized a general paradigm for learning and reasoning in the decision-making process: the Data-Information-Knowledge (DIK) paradigm, see also [9] for the seminal work on this paradigm. The DIK paradigm consist of three layers: the data layer, the information layer and the knowledge layer. The data layer describes the decision-making process in terms of some measurements and/or observations, the information layer describes the decision-making process in terms of examples and the knowledge layer describes the decision-making process in terms of a meaningful partitioning of the instance-space. Another element in the DIK paradigm is the *meta-knowledge*, which guides the processes of learning and reasoning. Because the DIK paradigm plays an important role in this thesis, we will discuss it in somewhat more detail.

Data

The data consists of all the observations (i.e. measurements) obtained from our (past) experiences (i.e. by instruments or senses). The types of observation determine and span the data space. In principle the data is unstructured, may contain redundancy and can be erroneous or noisy.

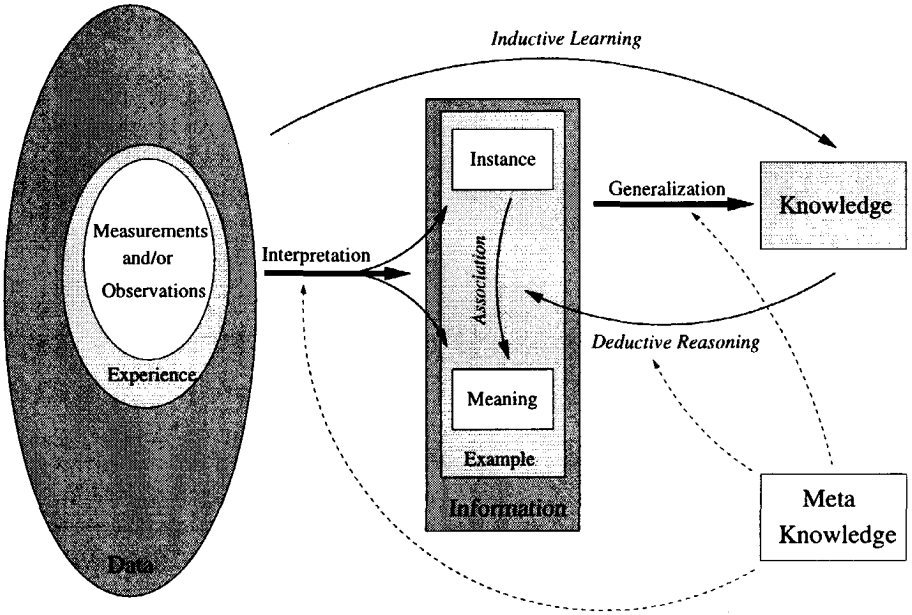


Figure 1.1: *The DIK-paradigm visualized.*

Information

Information (meaningful data) is obtained by interpretation of the data. By interpretation it is determined what data is of interest and what data is not. Often some preprocessing takes place on the data such as feature extraction, feature selection and filtering. Further, the interpretation of the data is essential for ordering the data into instances (outcomes) and associated meanings, i.e. examples. An instance can be thought of as the "conditional" information, and a meaning can be thought of as the associated "conclusive" information.

The instance space (outcome space) is spanned by the features (attributes), the instance space is usually denoted as feature space and an instance is usually denoted as a feature vector. The meaning associated to an instance can be obtained from events, classes, decisions or labels: *concepts*. In general, the meaning is obtained from a set of concepts which may be observed, assumed or defined. An example is an association of an instance to a meaning (meaningful instance).

We finally note that sometimes the feature values are given a meaning as well, irrespective of the meaning associated to the instance. This is usually referred to as a discretization of the feature-space. The thus obtained discrete values of the features are sometimes denoted as qualifications or modalities.

Knowledge

Knowledge is essentially a division of the instance space in regions which are associated to a meaning. This knowledge is obtained from generalizing (extending) the meaning of a few instances to an entire region in the instance space. Knowledge can be represented in many forms such as networks, functions or rules, to name but a few popular representations. Whatever the exact representation of the knowledge, the essence of knowledge is that it provides a description of the instance space (a synopsis) by which new (unobserved) instances can be given a meaning.

1.1.2 Uncertainty

When acquiring knowledge by experience in order to apply this knowledge to new instances, we essentially employ an empiricist epistemology. In the beautiful words of the 18th-century Scottish empiricist philosopher David Hume [64]:

"If reason determin'd us, it wou'd proceed upon that principle, that instances, of which we have had no experience, must resemble those of which we have had experience, and that the course of nature continues always uniformly the same"

However, Hume also said:

"There can be no demonstrative arguments to prove that, those instances, of which we have had no experience, resemble those, of which we have had experience."

Hume's statement implies that we can never be completely certain that the knowledge we have acquired by experience is a general truth which holds for all new instances. This is generally stated as the problem of induction, and Hume was one of the first to recognize it. One may think that a way around the problem is to collect all possible instances so that we have total experience. However, such an approach must be rejected from both a practical and theoretical point of view. We usually do not have the time to collect all the possible instances, and we can almost never be sure that we have collected all possible instances. The approach that we will follow is to estimate the uncertainty of our knowledge. We assume that, before having had any experience, we are completely ignorant and, hence, completely uncertain about the decision to make, and that after some experience we are less uncertain then before. Therefore we state that the goal of learning is to reduce the uncertainty in our knowledge, and thus in our decisions, as much as possible.

The field of Information Theory studies uncertainty measures and their application in for example communication technology. In Information Theory the uncertainty is usually quantified through the use of probability theory. However, there are two major paradigms in science that claim to quantify uncertainty:

- Probability Theory,
- Fuzzy Sets.

There have been - and still are - numerous debates on the necessity of having two paradigms for uncertainty. We will enter this debate by advocating the view that both paradigms capture a valuable but different type of uncertainty. Further, we will show in this thesis how both types of uncertainty can be combined in a single - synthesized - paradigm for uncertainty. In this paradigm - denoted as Fuzzy Probability - both types of uncertainty co-exist.

1.2 Synthesis

The discipline of computer science studying learning and reasoning is Artificial Intelligence (AI), which acquired the status of a discipline in the 50's. There are two mainstream fields studying decision processes from a different perspective. On the one hand there is the field of Pattern Recognition, which mainly deals with numerical, functional, and algebraic methods for learning and reasoning. On the other hand there is Machine Learning, which mainly deals with symbolic, logical, and heuristic methods. Roughly speaking, Pattern Recognition concentrates on the reasoning process, whereas Machine Learning is mainly concerned with the learning process. Since learning cannot go without reasoning and vice-versa, the line between Pattern Recognition and Machine Learning is rather thin. Therefore it is sometimes more useful to look at the individual techniques that are used in both fields. Each field can be specified into three individual techniques, roughly denoted by statistical, neural or fuzzy techniques. Here we regard the classical (Boolean) logic and set theory as special cases of fuzzy techniques. In a review of statistical, fuzzy, and neural techniques, Jim Bezdek concluded in 1993 [12]:

Indeed it is our expectation and contention that synthesis between the statistical, fuzzy and neural approaches to problems in this domain [pattern recognition] *will continue to grow - perhaps this integration will be the single most important horizon for our research*

However, in his review Bezdek aimed at the synthesis of neural techniques with either statistical or fuzzy techniques. Although we agree with the conclusion on the synthesis of techniques, we hold a slightly different view on which techniques to synthesize. We state that, for our purpose, the fuzzy and statistical techniques should be synthesized. This view will be motivated from a characterization of the techniques on a global level.

In terms of the DIK paradigm the techniques can be characterized as in Table 1.1. On the basis of this global characterization, we state that neural techniques are essentially a computational paradigm, whereas statistical and

Table 1.1: *Characterization of techniques.*

Technique	information	knowledge	generalization	reasoning
neural	input/output vectors	network	weight estimation	propagation
statistical	features + classes	functions	parameter estimation	probabilistic reasoning
fuzzy	sets + decisions	rules	conceptual partitioning	fuzzy logic

fuzzy techniques are paradigms for reasoning with - a specific type of - uncertainty. We further state that in statistical and neural techniques the meaning is mainly obtained from the conclusive information, whereas fuzzy techniques obtain their meaning mainly from the sets and the conceptual partitioning. The latter statement is based on the following reasons:

- the sets are usually obtained from clustering or from experts, hence, the sets have meaning irrespective of the decision made,
- the rules form - for each possible decision - a conceptual partition (obtained by an algorithm or provided by experts) of the instances associated to the same decision, hence, a rule provides an explanation for a particular decision.

The meaning as obtained from these two reasons will be referred to as the "mental fit" to the decision-making process. In contrast, the meaning obtained from the conclusive information will be referred to as the "data fit" to the decision-making process. The difference in emphasis the techniques make on either data fit or mental fit can also be observed in the emphasis on the error rate - sometimes referred to as *predictive accuracy* or *accuracy* for short - of the decision-making process. Fuzzy techniques usually have to accept the error rate in decision making as a result of their mental fit. As such, a low error-rate is an indirect validation of the mental fit of a fuzzy technique. Statistical and neural techniques, which depend on the data fit for a meaning, are directly designed and optimized for minimum-error-rate decision making.

For an accurate *and* expressive decision-support system, both a good data fit and a good mental fit is essential. To this purpose, we conclude that we should synthesize statistical techniques with fuzzy techniques. Currently this synthesis mainly takes place in a (sub)field of Machine Learning called rule induction, see also [102]. The reason for synthesis in this field is to improve the data fit of the rule induction techniques. However, usually statistical techniques are integrated with classical logic and crisp set techniques, which dominate this field. We will take the synthesis one step further and integrate a statistical technique with a fuzzy technique, see Figure 1.2, by using the fuzzy probabilistic framework.

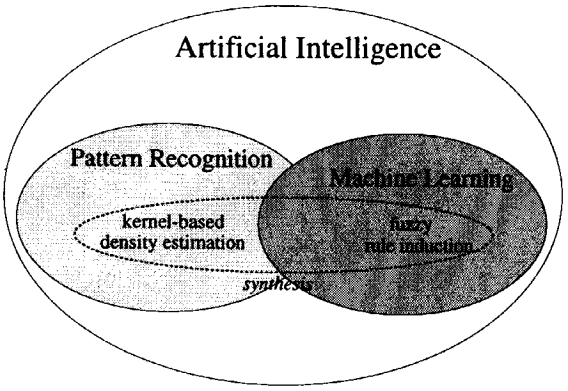


Figure 1.2: *The AI-field and some of its subfields as used in this thesis. The synthesis in this thesis is formed between kernel-based density estimation and fuzzy rule induction.*

1.3 Outline

This thesis presents the synthesis of density estimation with fuzzy rule induction. To synthesize these techniques we had to develop a more general framework for learning and reasoning; a framework that combined fuzziness and probability. This has resulted in the Fuzzy Probabilistic framework for Learning and Reasoning. The reason for the synthesis is no less than the ambition to develop a decision-support system that is suitable for exacting environments. Further, such a system should be able to surpass existing approaches in accuracy and expressiveness by providing a good data fit as well as a good mental fit. In Figure 1.3 the outline of this thesis is depicted as well as the interdependencies between the chapters. Although the techniques presented in this chapter are suitable for many types of decision problems, we will mainly demonstrate their use in classification problems.

Chapter 2 gives an overview of rule induction, emphasis is made on approaches using information theory. The discrepancy between data fit and mental fit in rule induction is discussed. References are made to Machine Learning and (fuzzy) rule induction. This chapter has been published in [42].

Chapter 3 introduces a general technique for density estimation: the double-kernel estimator. Although its roots lie in the well-known Parzen Windows technique, it is more efficient in decision making, i.e. classification. The statistical properties of the double-kernel estimator are extensively studied. References are made to other kernel-based techniques and experimental results are provided. Several ideas developed in this chapter are used again in other chapters. Essentially the double-kernel estimator eases the conception of the fuzzy-probabilistic framework. This chapter is based on [40, 41]

Chapter 4 presents the fuzzy probabilistic framework on the basis of some

fundamental assumptions. It is demonstrated how the double-kernel estimator can be derived from this framework. It is also demonstrated how the framework can be used for learning and reasoning in a more transparent way than by using the double-kernel estimator. References are given to related work on "fuzzy probabilities". Some ideas in this chapter have been expressed in [35],[36].

Chapter 5 presents the final rule induction method which utilizes the fuzzy probabilistic framework and the information-theoretic approaches presented in chapter two. The chapter concludes with experimental results on the basis of five publicly available data sets used in the Statlog project, a project in which 24 algorithms for classification have been compared. Early work related to this chapter has been presented in [37], [38].

In Chapter 6 the developed approach is evaluated for use in the Intelligent Anesthesia Monitor project. A feasibility study and a case study are presented and discussed. Part of this chapter has been published in [29] and earlier results have been presented in [39].

Finally, Chapter 7 discusses the main results and suggests directions for further research.

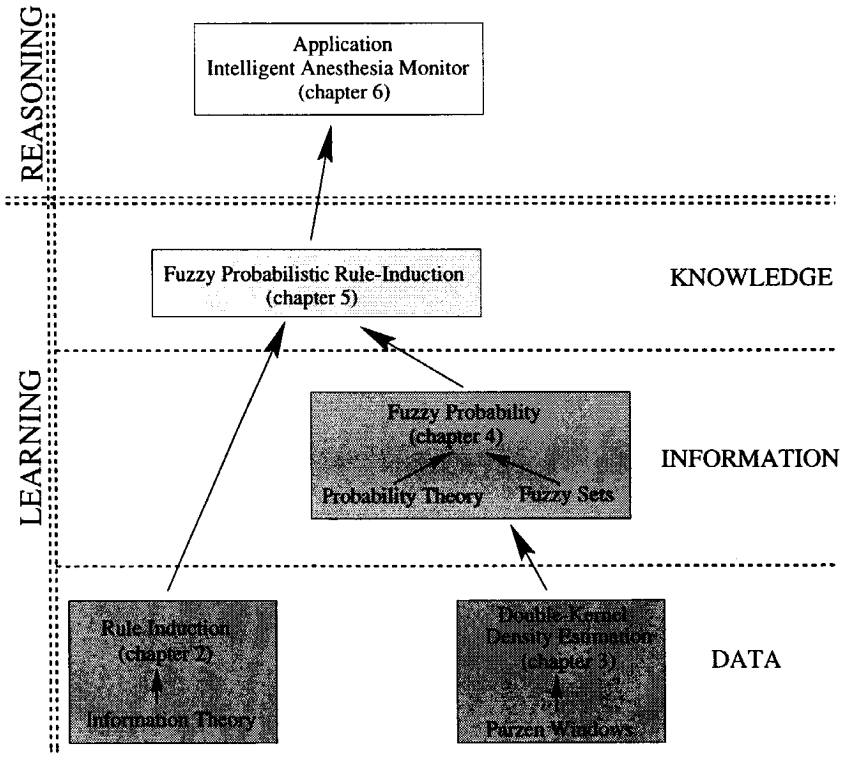


Figure 1.3: *Outline of this thesis in its chapters.*

Chapter 2

Rule Induction

Rule induction has been researched for some decades within the larger field of Machine Learning. Machine Learning in its turn is a part of the Artificial Intelligence (AI) discipline which achieved recognition as a discipline in the early 50's. The AI objective is to understand human intelligence and to develop intelligent systems. Machine Learning (ML) focuses on the ability of learning and gained momentum in the early 80's with rule induction (also known as concept learning) for which it is still well known. Early successful applications of Machine Learning include: discovering rules of chemistry using Meta-Dendral [19], discovering laws of physics using Bacon [78] and soybean disease diagnosis using AQ11 [89]. Apart from rule induction, other popular paradigms of the Machine Learning field are neural nets, genetic algorithms, case-based learning and analytic learning (theorem proving). Some early tutorials of the Machine Learning field can be found in [91, 92], more recent overviews can be found in [34, 58, 79, 122, 123]. Nowadays, several distinct reasons for using and studying Machine Learning can be observed:

- understanding human learning,
- developing computational learning,
- solving decision-making problems (e.g. classification),
- acquiring knowledge for expert systems,
- discovery of knowledge (data mining).

In this chapter we intend to familiarize the reader with rule induction. As we motivated in Chapter 1 of this thesis, we regard learning as the reduction of uncertainty in our knowledge. To this purpose we focus on information-theoretic approaches, because, as we will show, information theory provides an extensive framework for measuring the reduction of uncertainty. Instead of an in-depth study of one rule induction algorithm, we try to identify and clarify the main problem of rule induction and the issues involved.

The outline of this chapter is as follows. First, a motivation for using rule induction is given and the main problem of partitioning is stated. Second, some key elements of rule induction are discussed in order to clarify the partitioning problem. Finally, existing information-theoretic approaches to the induction problem are outlined by using a simple example.

2.1 Main Problem: Data Fit vs. Mental Fit

The main advantage that rule induction offers for decision-making problems is what is sometimes called a *mental fit* to the problem (see also [44]). Many techniques, like statistics or neural nets, partition the feature space into as many regions as there are classes by using some kind of discriminant function (e.g. a posterior probabilities, linear discriminant functions etc.). These techniques provide a *data fit*, in the sense that these techniques' sole goal is to optimize the accuracy of the classification, i.e. the prediction over unseen instances (predictive accuracy). Such techniques can be called black-box techniques. Unlike these black-box techniques, rule induction techniques partition the feature space into multiple regions (see Figure 2.1), where a region is represented in a (logical) symbolic way and associated with a class[1]. This way of partitioning the feature space can be regarded as generating "explicit" knowledge describing the data, it will be referred to as conceptual partitioning.

As an example from the medical domain consider a system that simply classifies a patient as being "ill" on the basis of blood pressure, ECG and other physiological measurements. Even if this system was an excellent classifier, it would not be of much use as a decision-support system. For proper decision support it is necessary to know whether the patient under consideration is ill because of a high blood pressure or a low temperature (or even because of both) in order to treat him properly. Rule induction is concerned with finding such explicit reasons, often referred to as *concept descriptions* or *rules*.

A second advantage that rule induction techniques offer comes from the partitioning process as well. Many rule induction techniques are searching for simple (general) concept descriptions, which often leads to dimensional reduction or feature selection. Feature selection is a preprocessing step in many other classification techniques, where a subset of features is selected to reduce the dimensionallity[2] of the classification problem. The intrinsic feature-selection property of rule induction makes it also possible to use "the best part" of each

[1]There exists an extensive nomenclature. In rule induction an outcome is usually called an instance, whereas an event is usually called a class or (target) concept. Further, sub-classes are often clusters in classes, and simply called concepts in rule induction. Features are sometimes referred to as attributes, and the feature space and attribute space are often used as synonyms for outcome space or instance space. However, the instance space is actually a subset of the feature space.

[2]One important reason for reducing the dimensionality is the so called dimensionality problem; the higher the number of dimensions, the better the classification can be if sufficient data is available. Unfortunately the number of data is usually small and it has been frequently observed that the dimensionality has to be decreased to obtain a better classification.

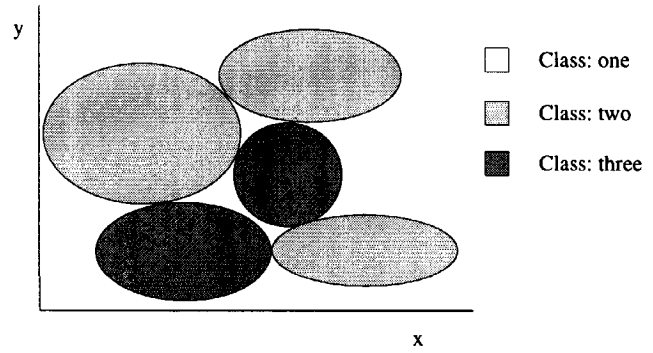


Figure 2.1: *Example of a non-overlapping partition for a three-class problem. The two-dimensional feature space, spanned by features x and y, is divided in several regions. Each region is associated with a class. A region associated with a class forms a concept description, the combination of all concept descriptions for a class forms a class description. Note that the third class is essentially described by an* else *rule.*

feature in a rule such that a better classification over unseen instances can be obtained than with a subset of features, while still the dimensionality per rule can be low.

As an example of a basic approach to partitioning consider Figure 2.2. Clearly the two-class data contains some structure. Suppose we chose to represent a region in the two-dimensional feature space, spanned by x and y, by logical conjunctions forming a rectangle of arbitrary shape, a region can then be written as:

If $(th_1 < x < th_2)$ **And** $(th_3 < y < th_4)$ **Then** class is $+$

Here, th_i denotes a threshold. A basic approach now consists of finding a small set of rectangles such that all examples are explained correctly. This set can be found for example by starting with the most general rule (also known as the empty rule, or Null-hypothesis, since it covers all examples (but not correctly!)):

If $(-\infty < x < \infty)$ **And** $(-\infty < y < \infty)$ **Then** class is $+$

Clearly this hypothesis needs specification (refinement), i.e. the boundary of the rectangle needs to be narrowed in order to exclude the negative examples. A possible algorithm is the following :

- step 0. Initialize a general hypothesis by the Null-hypothesis,
- step 1. randomly select a positive example from the data as a seed,
- step 2. compare the seed with a negative example and minimally narrow

the Null-hypothesis rectangle such that it excludes the negative example but covers the seed,

- step 3. repeatedly reduce the rectangle until all negative examples are excluded; the rectangle is then a maximally general concept,
- step 4. if all positive examples are covered by the maximally general concept, then go to step 6,
- step 5. remove the covered positive examples from the data and repeat step 1 to step 5,
- step 6. the disjunction of all maximally general concepts is a complete and consistent (target) concept description of the positive examples; end.

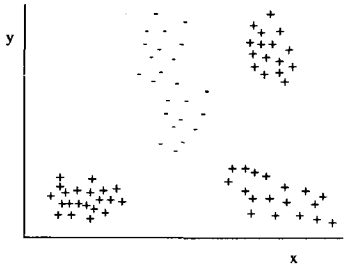


Figure 2.2: *Example of a set of observations consisting of two classes (positive examples and negative examples) in a two-dimensional feature space, with features x and y.*

This algorithm leads to a description of the positive examples and the negative examples can be described implicitly by a final Else-rule or explicitly by repeating the algorithm for the negative examples. This algorithm is in fact a simplified version of the well known AQ-family of algorithms introduced by Michalski, one of the founders of rule induction. This particular version of the algorithm has some drawbacks. First, the final partition (concept descriptions) depends on the order in which the examples are processed (see Figure 2.3), second, in case of overlapping classes, the description becomes too restrictive for the positive examples and/or will lead to an incomplete description, third, rectangles may not be the best representation for all problems (compare ellipsoidal regions).

What the above algorithm illustrates is that by defining a *representation* for the descriptions, the partitioning can be regarded as a search through a space of descriptions for the "optimal" partitioning [98]. Usually "optimal" is defined by a set of preferences or criteria. Quite often many possible partitions of a feature space exist and it is not obvious which of these partitions is optimal. The problem being the selection of a partition which is both a mental fit as well as a data fit. This problem is still not completely solved since it is difficult to

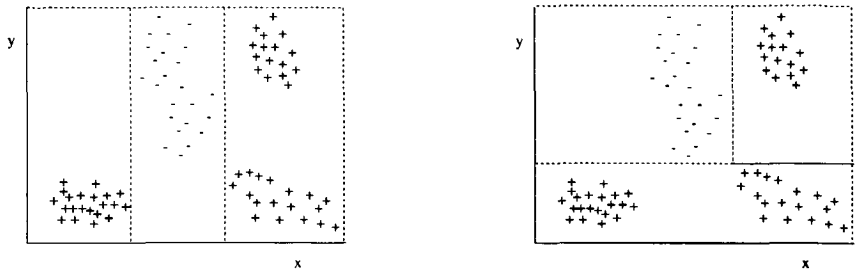


Figure 2.3: *Examples of possible non-overlapping partitions for the two-class problem. The two-dimensional feature space, with features x and y, is divided in several regions. Each region is associated with a class from the event space; a region associated with a class forms a concept description.*

measure the mental fit. Further it is still a topic of research how it should be weighted against data fit. Quite often a choice for mental fit (i.e. more general partitioning) is a choice against data fit (i.e. less accurate classification). This chapter will frequently return to the problem of data fit vs. mental fit since it plays such an important role in many aspects of rule induction.

2.2 Elements of Rule Induction

The goal of rule induction can be stated as follows [79];

- Given a set of examples,
- Find a set of understandable concept descriptions of the examples that, to the extent possible, correctly classifies novel instances.

The requirements for the partitioning, forming a set of concept descriptions, are in general twofold. First, all concept descriptions together should completely cover (explain) the instance space: *completeness*, second, these concept descriptions should be as simple as possible: *Occam's razor* [124]. The rationale behind Occam's razor is that a simpler explanation is more likely to capture the essence of the problem. It is difficult to satisfy these requirements perfectly, while keeping the goal of the partitioning in mind. In fact it can be stated that the partitioning problem in rule induction is to maximize the classification performance while minimizing the complexity of the descriptions. Many algorithms have emerged that all make an effort to approximate the above goal and requirements. Several key elements can be distinguished, which play an important role in induction. Key elements that will be discussed in this section are:

- Domain Representation,
- Knowledge Representation,

- Search Strategy,
- Classification Method.

For other taxonomies see also [85, 91]. Although much has been and can be said about these key elements, this section will merely touch upon these elements for reasons of brevity. However, where appropriate, we will discuss in some more detail how these key elements can influence both the data fit as well as the mental fit.

2.2.1 Domain Representation

The first problem faced in induction is the representation of the domain by a set of instances. The set of all possible instances forms the instance space. In general the instance space is represented as a multidimensional space, also called the feature space, in which instances are represented by a vector of feature values. Further, each instance is given a label according to the event (class) to which it belongs; in this way examples are obtained. The set of all events, also known as the set of classes, is usually denoted as the event space.

In general two types of domains are considered: discrete and continuous domains, depending on the type of feature value. The benefit of a discrete domain is that there exists a finite number of values that completely describes the feature space in detail. This finite number of values forms the most detailed partition and is simply the discrete feature space itself. In continuous domains, such a partition does not exist or is at best infinite. In discrete domains, the search for the best partition is therefore naturally restricted by the finite feature space. In continuous spaces such a natural restriction does not exist and in theory the number of different partitions is truly infinite (although also in discrete cases the number of possible partitions can be very large). Early induction algorithms, such as Michalski's AQ [88] and Quinlan's ID3 [112] focused on discrete domains. Nowadays many algorithms provide techniques for continuous domains as well.

To deal with continuous domains there exist three different approaches. One way of dealing with continuous domains is to use an a priori clustering of the feature values, which results in a discrete feature space formed by the clusters. Separating the discretization of the domain and the actual search process has the advantage that for example an expert may perform the clustering, possibly leading to a better understanding of the actual rules (mental fit). A second, more dynamic, approach is through the use of threshold concepts, i.e. $x > \mathit{threshold}$ as in C4.5 [114]. These approaches do not use a priori discretization, but make the discretization process a part of the search process. The advantage of this approach over an a priori approach is that the concepts can be more precisely tuned to the "true" decision boundaries (improved data fit), since in an a priori approach these boundaries are essentially predefined. However, it is not clear how suitable threshold concepts are for obtaining a mental fit. In essence threshold concepts provide (discriminative) decision boundaries rather than (characteristic) descriptions, much in the same way as statistical techniques

like discriminant analysis. Consider our medical example again: to know that a patient is ill because the blood pressure is above 140.5 mmHg may not aid his treatment. It may be necessary to know that there exist two cases (clusters) of high blood pressure, one around 150 mmHg and one around 175 mmHg, which may require different actions, even though in both cases the class is "ill". Hence, the representation and discovery of "sub-classes" is difficult when using threshold concepts. A third approach for dealing with continuous domains is by *fuzzification* of the feature values, rather than by *discretization.* This approach, by using fuzzy sets [146], tries to obtain the mental fit of a priori clustering and the data fit of threshold concepts. The following reasons for the use of fuzzification in rule induction can be found in literature (although not always stated explicitly).

- Fuzzy sets and fuzzy logic provide means for dealing with the combination of numerical information and linguistic information, which could provide a good basis for both data fit and mental fit [138].
- Fuzzy rule-based controllers using expert rules perform very well. Since knowledge acquisition from experts is difficult, because knowledge is often implicit, there is a clear need for automatic induction of fuzzy rules [59].
- Crisp regions are only rarely good approximations of the actual decision boundaries in cases where the data contains noise or overlapping classes. Introducing fuzzy regions may therefore lead to a better approximation, and thus a higher accuracy. [22]

2.2.2 Knowledge Representation

One of the most influential elements in both mental fit as well as in data fit is the representation of the concepts (regions) generated during the partitioning. If, for example, the representation consists of predetermined hyper-cubes, then the decision boundaries will have a block-like shape and will strongly bias the classification if the true decision boundary is not block-like. The flexibility of the concepts, i.e. the degrees of freedom in the representation of the concept, not only influences the data fit but also determines the explanation capability and, hence, the mental fit. However, a too flexible description may in theory provide an optimal partition, but the computational costs to arrive at such partition (if at all) may be very high. In that sense, the representation also guides the search, in that it can restrict the search space by allowing only a subset of all possible partitions. So, the choice for knowledge representation is often a compromise between the flexibility of the representation and the computational cost of the search process.

A partition can be represented in several ways. Three representations are traditionally distinguished: decision trees, decision lists and production rules. A decision tree is a directed, a-cyclic graph of nodes and arcs. At each node a simple test is made, leading to a next node; at the end-nodes (leaves) decisions are made with respect to the class labels see Figure 2.5. Decision trees were

introduced in the ML community by Quinlan with his ID3 algorithm [112, 113] for discrete domains, which was extended to C4.5, that is also applicable in continuous domains through the use of threshold concepts [114]. C4.5 can be said to be one of today's mainstream decision tree algorithms. A second mainstream decision tree algorithm is CART described in [15], which is regarded as the seminal statistical work on decision trees. Other examples of popular algorithms using decision trees are Assistant [20] (which introduced improvements on dealing with missing values, attribute splitting and pruning), and Cal5 [101] which was specifically designed for dealing with continuous domains. Recent tree algorithms using fuzzification can be found in [22, 27], but an excellent fuzzy-tree algorithm is found in [62].

A decision list is an ordered set of rules of the form:

If test **Then** class **Else If** test...

An example of an algorithm using such lists is CN2 [26].

Production rules, or simply "rules", are a set of unordered If-Then rules of the form:

If test **Then** class

It is generally agreed in the ML community that rules provide the best mental fit to the data. They are easy to understand since each rule is a complete relation between a region in the feature space and a class (a complete concept description). In decision trees and decision lists, in contrast, the concept descriptions are distributed over the separate tests in the nodes and the else part, respectively, in order to minimize the number of tests necessary to obtain the classification. The disadvantage of production rules is that some test has to be performed several times before the right rule is found for classification. Since trees and lists are more economical and rules have a better mental fit, many algorithms obtain the best of both representations by providing means to convert trees into rules and vice versa. Examples of algorithms focusing on production rules are the AQ family of algorithms introduced by Michalski [88, 90]. Other examples are the CN2 algorithm [26][3], the minimal entropy approach [110] and ITrule [129]. Production rule algorithms using fuzzification can be found in [1, 2, 50, 59, 80, 103, 130, 138]. Despite the difference in knowledge representation, all of the above forms represent (more or less obvious) partitions of the feature space by (a conjunction or disjunction of) logical tests like $>, <=, \in$.

An integral part of knowledge representation, that we like to discuss briefly, is the representation of uncertainty. The uncertainty model guides the decision-making process (reasoning process) of associating a class with a region in the feature space. Many algorithms use probability to assign a class to a region, i.e. the most frequent class in a region determines the class associated with

[3] CN2 has two modes: generating an ordered decision-list or an unordered set of production rules

the region. In approaches using fuzziness, the assignment of classes is based on similarity, the example that fits the region best (prototype example) determines the class. If the data set contains no noise or overlapping classes, then the fuzzy or probabilistic approaches do not differ in class assignment for nearly equal regions; one may even claim that in this case there is no need for representing uncertainty. Unfortunately, many data sets cannot be perfectly described such that fuzzy and probabilistic approaches may lead to different class assignments for nearly equal regions. How uncertainty is represented and handled is therefore an important issue in problems containing noise and/or class-overlap.

2.2.3 Search Process

The search process in rule induction can be characterized as a hypothesize-and-test cycle, which is recognized in psychology as a typical way of learning by human beings [18]. The search process typically consists of three mechanisms, which are often interwoven: hypotheses generation, search strategy and selection.

Hypothesis generation

The search process provides a mechanism for hypothesis generation, that can be model-driven or data-driven. In a model-driven approach, the hypotheses are generated according to some predefined scheme (e.g. in an exhaustive search); in a data-driven approach, it is the data itself that induce the hypotheses (e.g. by generalizing a specific instance to an entire region).

Especially in data-driven approaches, the mechanism for hypothesis generation has to use generalization or specialization techniques, which lead to more general or more specific hypotheses. Generalization (and it's counterpart specification) depend on a single principle; increasing (decreasing) the scope of the logical test in the hypothesis. The scope of a test, or the *coverage* of a hypothesis, are both terms that are used to describe the generality of the hypothesis, e.g. the larger the scope or coverage the more general a hypothesis is. As an example we will use a production rule to illustrate some ways of generalization. Given a production rule:

If blood pressure = high **And** heart rate = low **Then** alarm

The most common way of generalization is by use of the *dropping condition* principle, i.e.:

If blood pressure = high **Then** alarm

Less common is to change the conjunction into a disjunction:

If blood pressure = high **Or** heart rate = low **Then** alarm

which is known as *turning conjunction into disjunction* principle. A different way of generalizing is the *adding alternative* principle, i.e.:

If blood pressure = high **Or** very high **And** heart rate = low **Then** alarm

Suppose we had a concept in our knowledge representation language that denotes the value "high or very high" by "above normal", we also might have used this concept in order to obtain the same effect. This way of generalization is known as the *extending reference* principle, i.e:

If blood pressure = above normal **And** heart rate = low **Then** alarm

Many other schemes have been introduced to describe the specific ways of generalization/specification, for more details see [91] for a classical overview. If fuzzy concepts are used then there are even more ways to change the scope of the test, e.g. by using fuzzy hedges, see [148] for the seminal work on fuzzy hedges and see [70] for a good textbook.

The hypothesis mechanism also provides the direction of the search; it determines whether the learning starts with the most specific concept descriptions allowed in the knowledge representation (often the domain representation itself) and moves toward more general descriptions, or vice versa. The type of search in the example algorithm at the beginning of the chapter is a data-driven general-to-specific search. Model-driven and specific-to-general search strategies also exist. The famous version-space algorithm introduced by Mitchell [97], even combines several search strategies in order to keep track of all consistent partitions. Several traditional types of mechanisms exist: exhaustive search, specific-to-general search, general-to-specific search, beam search, recursive search etc. It goes beyond the scope of this chapter to describe these individual mechanisms, a good overview can be found in [79].

Apart from the direction of the search, the hypothesis mechanism determines how the data set is searched, either in a single batch process which allows optimization over the data set, or incrementally where each new example provides new evidence, or by combinations of batch and incremental processes (an incremental batch-process). Incremental learning is often motivated by the way humans learn: humans seem to learn sequentially from each new example without having to refer to an explicit database and without beginning from scratch. Incremental versions of induction algorithms exists for rules [94] as well as for trees [134].

Search strategy

The search process needs a strategy to arrive at the learning goal. There are several paradigms for rule induction, of which the most popular are:

- combinatorial paradigm: illustrative but hardly used outside textbooks,
- divide-and-conquer paradigm: used in tree-based algorithms of which the

prototype is ID3,

- covering paradigm: used in many rule-based algorithms of which AQ can be said to be the prototype.

The combinatorial approach can be considered to be the most naive approach to learning. In this approach many possible partitions are evaluated on the basis of classification performance and simplicity of the partition. It involves lots of "number-crunching", is hardly applicable in high-dimensional problems and is often guided by optimization procedures such as genetic algorithms. It is rarely used except occasionally for optimization over possible clusterings for discretization, as in [59].

In the more sophisticated divide-and-conquer paradigm one starts with selecting a single feature as a node and tests all of its values. A test forms a (sub-)partition and is evaluated according to some quality criterion. Those tests that fail the criterion are further specified by adding another feature-node and so on. Those tests that pass the criterion form the end nodes (leaves) of the tree. As an example see Figure 2.5.

The covering paradigm uses a recursive procedure to obtain the final rule base. In each recursion one tries to find the best rule according to some selection criterion. The examples that are covered by the selected rule are then removed from the training set and the next recursion starts. In general, this process stops when the feature space is completely partitioned or when all the examples are covered.

Selection

In order to select possible partitions or rules from candidate hypotheses, a quality measure, often referred to as a preference criterion, is used. A well-designed measure combines two aspects in a single *quality* measure, mental fit and data fit. The following are important elements in mental fit.

- Completeness: the instance space should be described completely.
- Coverage (also referred to as generality): the regions of a partition should be powerful, as measured by the (absolute or relative) total number of examples covered by a region (irrespective of the classes). Related to these measures are the density and sparseness of a region.
- Simplicity: the description length of a partition should be as small as possible. It can be measured by the size of a decision tree, the number of rules, the number of tests in a partition or in the conditional part of a rule. Note that coverage also influences the simplicity.
- Explainability: the concept description should be understandable to a user. This is a more-qualitative measure, rules are usually preferred over trees, and qualifications ("high", "low") are preferred over thresholds. Depending on the application, non-disjoint rules can be preferred

over disjoint rules, conjunctions can be preferred over combined conjunctions/disjunctions etc..

The following are important elements in data fit.

- Consistency (also referred to as specificity, certainty or discriminative power, predictive accuracy): the amount of (un)certainty with respect to the classes within a region, often measured by a conditional probability or entropy on the basis of the examples in a training-set.
- Classification Error: the actual error, usually measured as the average probability of error, i.e. *error rate*, on a test-set or by using cross-validation (leave one out, leave whole out, etc.).

A motivation for these elements can be found in [17, 57, 129]. Many measures have been proposed for combining data fit and mental fit, not necessarily equally weighted. The most successful of these measures rely on information-theoretic measures such as entropy or some other statistic. For an overview of (statistical) measures for evaluation see [53, 102]. In addition to using the quality as a relative measure for selecting some partition over others, it is not uncommon to let a user define a quality under-bound for accepting partitions or rules. The CN2 algorithm [26], which improves upon the basic AQ algorithm using an information measure to cope with noise and class-overlap, uses such a minimum quality; the algorithm recursively searches for the best rule that at least satisfies the minimum under-bound. Such a search *heuristic* or *bias* is often necessary to prevent an algorithm from getting trapped in a local optimum.

2.2.4 Classification Method

The final key element in rule induction that we would like to touch upon is the method for classification. In decision trees or decision lists, classification is straightforward since the partitions are disjoint; that is the partitions consists of non-overlapping regions. A new instance is therefore a member of only one partition and can be classified accordingly. However, if an instance is covered by several rules, then it is not obvious according to which rule it should be classified. There are several ways to deal with this problem of multiple coverage. The simplest is to *order* the rules according to a quality measure and classify the new instance according to the first rule that covers it. Another approach is to use a weighted classification using all rules that cover the instance, as is frequently used in fuzzy algorithms. Finally, one can try to correct for the multiple coverage by forming a disjoint partition for reasoning. This can be done by literally forming a new disjoint rule base or by a recalculation that leads to the same effect (a technique sometimes referred to as *backtracking*). On itself this correction is a sensible approach, since it somewhat decouples the functionality of data fit and mental fit. It can be questioned, however, what the integrity is of such an approach, since in essence a different (general) partition is used for classification than originally derived during learning. Again, this is an example of how a choice for data fit can be a choice against mental fit.

2.3 Information-Theoretic Approach

This section will outline the information-theoretic approach to induction. The goal is to arrive at a measure that can select some partition (or even a complete rule base) over other partitions (rule bases). In the information-theoretic approach, the measure used in both tree induction as well as in rule induction is based on mutual information. In order to explain this measure, it is necessary to clarify the notion of information, often referred to as entropy or uncertainty [135].

2.3.1 Information and Uncertainty

Suppose somebody picks randomly a number ξ out of the set $N = \{1, 2, 3, 4\}$. Suppose further that it is our task to find out which number this is with a minimum number of yes/no questions. As a first guess we may think that we can always find the answer with at most three questions, since there are only four possible numbers. If we are lucky we may have the answer right the first time, but if we are unlucky we need three questions. However, the answer is that we can always find the answer with at most two questions; see Figure 2.4. The trick is to note that an answer to a question reduces the uncertainty we have with respect to ξ. If we do not ask any question, then there are four possible numbers, all being equally likely. The uncertainty $H(N)$ in this discrete case is now defined as the *discrete information* and is usually expressed in bit(s):

$$\begin{aligned} H(N) &= \sum_{n=1}^{4} -P(n) \log_2 P(n) \\ &= \sum_{n=1}^{4} -0.25 \log_2 0.25 = 2 \; bits \end{aligned} \tag{2.1}$$

where $P(n)$ is the probability that ξ equals n. Suppose we ask the question: "Is ξ equal to three or four?", then the answer (either yes or no) leaves only two equally likely numbers. The uncertainty about ξ that we have after obtaining the answer "yes" to our question is then defined as the discrete *conditional information* $H(N|yes)$:

$$\begin{aligned} H(N|yes) &= \sum_{n=1}^{4} -P(n|yes) \log_2 P(n|yes) \\ &= 0 + 0 + 0.5 + 0.5 = 1 \; bit \end{aligned} \tag{2.2}$$

where $P(n|yes)$ is the probability ξ equals n *given* that the answer is "yes"; a conditional probability. This quantity can be viewed as the uncertainty in the answer or as the information still necessary to find the final answer. Likewise

for $H(N|no)$:

$$\begin{aligned} H(N|no) &= \sum_{n=1}^{4} -P(n|no)\log_2 P(n|no) \\ &= 0.5 + 0.5 + 0 + 0 = 1\ bit \end{aligned} \tag{2.3}$$

It is clear that discrete information nicely predicts the remaining number of yes/no questions which we still have to ask in order to obtain a *certain* answer. However, since we do not know the answer to the question beforehand, we can also calculate the *expectation* of the remaining uncertainty. This quantity is also known as the discrete *average conditional information* $H(N|Q)$, where $Q = \{yes, no\}$.

$$\begin{aligned} H(N|Q) &= \sum_{q=yes}^{n} o \sum_{n=1}^{4} P(q)\ P(n|q)\log_2 P(n|q) \\ &= \sum_{n=1}^{4} -P(yes)\ P(n|yes)\log_2 P(n|yes) + \\ &\quad + \sum_{n=1}^{4} -P(no)\ P(n|no)\log_2 P(n|no) \\ &= 0.5 * 1 + 0.5 * 1 = 1\ bit \end{aligned} \tag{2.4}$$

We may now ask how much uncertainty the answer to this question may reduce, or what the *expected information gain* is upon receiving the answer. Since we started with an uncertainty of 2 bits and expect to have a single bit of uncertainty left having asked the question, we may expect that the reduction of uncertainty equals 1 bit. In general this quantity is known as the discrete *mutual information* and is defined as:

$$I(N;Q) = H(N) - H(N|Q) \tag{2.5}$$

Suppose we would like to immediately ask the question "is ξ equal to four" instead of asking is "is ξ equal to 3 or 4", then we can immediately calculate that the expected information gain of this question equals:

$$\begin{aligned} I(N;Q) &= H(N) - H(N|Q) \\ &= 2 - \sum_{n=1}^{4} -P(yes)\ P(n|yes)\log_2 P(n|yes) + \\ &\quad + \sum_{n=1}^{4} -P(no)\ P(n|no)\log_2 P(n|no) \\ &= 2 - (0 + 1.1996) = 0.8004 bit \end{aligned} \tag{2.6}$$

Hence, the expected information gain of this question is less than the gain found for the question "is ξ equal to 3 or 4" (which was 1 bit), therefore *on average*

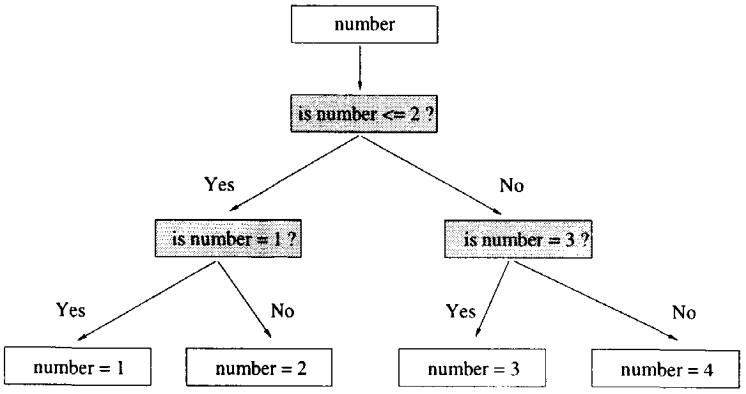


Figure 2.4: *Example of finding a number between 1 and 4 with only two questions. Start from the top and work your way down to the leaves.*

we are better of with asking "is ξ equal to 3 or 4" (and only after we obtain a yes to this question it is useful to ask: "is ξ equal to four").

Mutual information helps in selecting the proper questions. In much the same way it can help in selecting the proper partition, since a partition is essentially formed by conditions put on the instance of a test (question). In the information-theoretic approach it is now assumed that a partition should provide an information gain, if it is to be useful. The partition that provides the highest gain is then selected over others.

Suppose we have a partition R into n disjoint regions of the instance space $R = \{R_1, ..., R_j, ..., R_n\}$, where each region can be associated with an event (class) C_i from the event space $C = \{C_1, ..., C_i, ..., C_k\}$ by a conditional probability $P(C_i|R_j)$. The discrete information of the event space is then defined as:

$$H(C) = E[-\log_2 P(C_i)] = \sum_{i=1}^{k} -P(C_i) \log_2 P(C_i) \tag{2.7}$$

the discrete conditional information as:

$$H(C|R) = E[-\log_2 P(C_i|R_j)] = \sum_{j=1}^{n} \sum_{i=1}^{k} -P(C_i, R_j) \log_2 P(C_i|R_j) \tag{2.8}$$

and the mutual information as:

$$I(C;R) = H(C) - H(C|R) \tag{2.9}$$

2.3.2 Example of Tree Induction

As an example of tree induction, consider the following set of examples of eight patients classified as healthy or ill on the basis of their heart rate and mean

Table 2.1: *Examples of patients.*

Patient no.	Heart Rate	Blood Pressure	Class
1	irregular	normal	ill
2	regular	normal	healthy
3	irregular	abnormal	ill
4	irregular	normal	ill
5	regular	normal	healthy
6	regular	abnormal	ill
7	regular	normal	healthy
8	regular	normal	healthy

blood pressure, that is shown in Table 2.1. Suppose our problem is to find out, with a minimal number of questions, when a patient is ill or healthy. Or in other words, to divide these examples in groups of "ill" patients and "healthy" patients.

By using the mutual information, the first question leading to a partition is easily found. We note that the uncertainty with respect to the class without partitioning equals:

$$\begin{aligned} H(Class) &= -P(healthy)\log_2 P(healthy) - P(ill)\log_2 P(ill) \\ &= 0.5 + 0.5 = 1\ bit \end{aligned}$$

where the a priori probabilities are calculated from the examples (each being 0.5 since there are 4 ill and 4 healthy patients). Now we have several possible ways to partition the examples. We could partitioning the examples according to the heart rate, but we could also partition the examples according to the blood pressure. In order to choose between these possibilities, we simply calculate the information gain (mutual information) of each. For the heart rate, we note that it can take on the values "regular" and "irregular", we obtain:

$$\begin{aligned} &H(Class|Heart\ Rate) = \\ &= -P(irregular)\ P(healthy|irregular)\log_2 P(healthy|irregular) + \\ &\quad -P(regular)\ P(healthy|regular)\log_2 P(healthy|regular) + \\ &\quad -P(irregular)\ P(ill|irregular)\log_2 P(ill|irregular) + \\ &\quad -P(regular)\ P(ill|regular)\log_2 P(ill|regular) \\ &= -0.375 * 0 - 0.625 * 0.8\log_2 0.8 - 0.375 * 0 - 0.625 * 0.2\log_2 0.2 \\ &= 0.45\ bit \end{aligned}$$

Whereas we have for the blood pressure:

$$\begin{aligned} &H(Class|Blood\ Pressure) = \\ &= 0.25 * 0 + 0.75 * 0.66\log_2 0.66 + 0.25 * 0 + 0.75 * 0.33\log_2 0.33 \\ &= 0.69\ bit \end{aligned}$$

hence the information gain for these partitions is:

$$\begin{aligned} I(Class; Heart\ Rate) &= 1 - 0.45 = 0.55\ bit \\ I(Class; Blood\ Pressure) &= 1 - 0.69 = 0.31\ bit \end{aligned}$$

Because it gives a larger information gain, we choose to partition according to the heart rate. Now the group for which the heart rate is "irregular" is perfect in the sense that the three patients that have an irregular heart rate are indeed all "ill". Hence, we can decide with certainty that if the heart rate is irregular, then the patient is ill. However, the other group is less clear-cut: of the five patients that have a regular heart rate, four patients are healthy but one is ill. Therefore, we cannot decide without uncertainty what the patient class is if the heart rate is regular, so we need to refine this group. Normally, this refinement entails a recursion: generate possible (sub-)partitions for the group to be refined and select the one having highest information gain. In this example problem, however, we only have one possibility for refinement: the blood pressure. If it is normal, then the patients who have a regular heart rate are indeed healthy, but if the blood pressure is abnormal than the patient is "ill". In this way we have arrived at a scheme depicted in Figure 2.5, which can be used for other patients which where not in the data base. What we have arrived at is a decision tree, and the type of induction that we performed was identical to that introduced by Quinlan in his famous ID3 tree induction algorithm [112, 113].

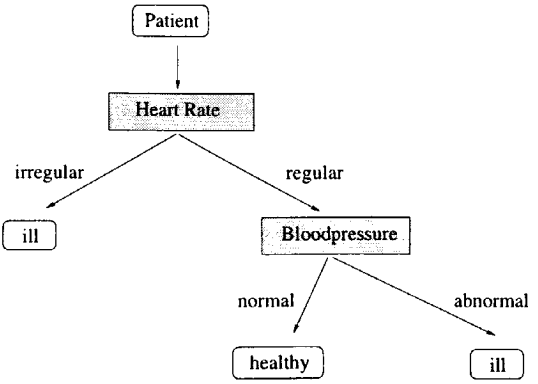


Figure 2.5: *Example of a simple decision tree.*

In view of this powerful and general tree induction algorithm, the reader might wonder why so many other algorithms exist, even for the information theoretic approach. The reason is that real-world problems that are more complicated than our simple, noise-free database without overlapping classes, demand better methods. If noise is present the tree may start to fit the noise or become too specific to deal with the class overlap, a condition called "over-specification" or "over-fitting". The solution to that problem is to build in an appropriate stopping criterion or a tree pruning algorithm, by which irrelevant

sub-partitions can be prevented or removed, respectively. Irrelevance can also be tackled by information theory and many other statistical measures [102]. Other reasons for the diversity in algorithms stem from the type of discretization, hypothesis generation, search strategy etc., as discussed in section 2.2.

2.3.3 Information-Theoretic Rule Induction

Although decision trees provide a powerful means for generating useful partitions, there is a different way of approaching the problem. If we take a good look at the examples in Table 2.1, it is clear that the patient is ill if the heart rate is irregular or if the blood pressure is abnormal; otherwise he is healthy. Hence, we can form the following *rule base*, that gives the same decision as the decision tree in Figure 2.5:

- **If** Heart Rate is irregular **Then** Patient is ill
- **If** Blood Pressure is abnormal **Then** Patient is ill
- **If** Heart Rate is normal **And** Blood Pressure is normal **Then** Patient is healthy

The construction of such a rule base is not trivial, even though it seems easy. A way to construct a rule base from the tree of Figure 2.5 is by "walking along the path of the tree". We then get the following rule base:

- **If** Heart rate is irregular **Then** Patient is ill
- **If** Heart rate is regular **And** Blood Pressure is abnormal **Then** Patient is ill
- **If** Heart rate is regular **And** Blood Pressure is normal **Then** Patient is healthy

As can be seen through comparison of the rule bases, the second rule is somewhat more specific than the second rule of the first rule base, and is therefore somewhat more complicated to explain. For more complex problems in high-dimensional feature spaces, explanations become much more complicated. In such cases a rule base with *overlapping* rules provides a simpler explanation and, thus, a better mental fit than a decision-tree or disjoint rule-base.

In this final section we will outline an information-theoretic approach to rule induction that uses the J-information measure, first introduced by Smyth and Goodman [128]. With this approach it is possible to construct rule bases with overlapping rules.

Data, information, and knowledge

A useful paradigm for rule induction is the Data-Information-Knowledge paradigm as outlined in the introduction of this thesis. Suppose we have a continuous n-dimensional domain which we would like to partition. The first step to be

taken is to transform the continuous data in discrete information which consists of specific regions, formed by qualifications, and an associated decision. The search for possible rules is then performed on the information and the selected rules are stored in a rule base, called the knowledge. The knowledge is used for classification and explanation of new instances. In this way we have obtained three-layers, a data layer, an information layer and a knowledge layer. The paradigm is depicted in Figure 2.6.

Since the rules are expressed in qualifications obtained from a discretization, it can be advantageous to let an expert determine the discretization. In that case the qualifications can be viewed as a reference frame in which the classification problem should be cast. In decision-support systems, such a reference frame is useful for explaining results to an expert "in his own words" [28]. However, the transformation from data to information can be provided by any appropriate discretization method, such as K-means clustering [4]. For an overview of clustering, see [9]. As in the covering paradigm mentioned in 2.2.3, we require that the information layer completely covers the data layer and that the knowledge layer completely covers the information layer.

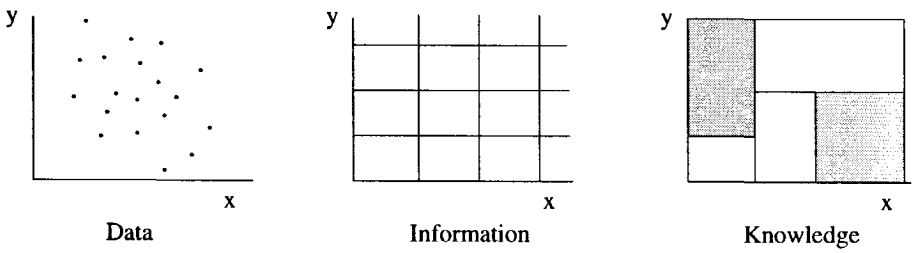


Figure 2.6: *Figure depicts the Data-Information-Knowledge paradigm. The information is a discrete representation of the data, as indicated by the regions. The knowledge consists of general rules for each class (event), which is indicated by the shading.*

The J-information measure

The formation of the knowledge layer from the information layer is essentially guided by mutual information, just as in tree induction. However, the main advantage of the J-measure over mutual information is that it allows the evaluation of a single rule (a region associated with a class) rather than a complete partition. To see this, we note that, according to [13], the mutual information

$I(C;R)$ can be written as an expectation:

$$\begin{aligned} I(C;R) &= H(C) - H(C|R) \\ &= E_{C_i,R_j}[\log_2 \frac{P(C_i|R_j)}{P(C_i)}] \\ &= \sum_{j=1}^{n}\sum_{i=1}^{k} P(R_j)P(C_i|R_j)\log_2 \frac{P(C_i|R_j)}{P(C_i)} \end{aligned} \tag{2.10}$$

The j-information measure is defined as:

$$j(C;R_j) = \sum_{i=1}^{k} P(C_i|R_j)\log_2 \frac{P(C_i|R_j)}{P(C_i)} \tag{2.11}$$

such that:

$$I(C;R) = E_{R_j}[j(C;R_j)] \tag{2.12}$$

Here, $j(C|R_j)$ expresses the goodness-of-fit of the region R_j with respect to the classes; this a measure for the data fit of the rule that covers region R_j. The J-measure is an extension of the j-measure with a mental fit part:

$$J(C;R_j) = P(R_j)j(C;R_j) \tag{2.13}$$

The larger the probability $P(R_j)$, the better is the mental fit of the rule that covers region R_j. Since a rule essentially gives information over a single class C_m (the majority class) and its complement (not C_m), the J-measure according to:

$$\begin{aligned} J(C;R_j) &= P(R_j)P(C_m|R_j)\log_2 \frac{P(C_m|R_j)}{P(C_m)} + \\ &\quad + P(R_j)(1-P(C_m|R_j))\log_2 \frac{1-P(C_m|R_j)}{P(C_m)} \end{aligned} \tag{2.14}$$

is more suitable for the specific task of rule induction, see [102].

Since the J-measure compares the a priori probability with the a posteriori probability, it is also referred to as information gain. This J-measure has been introduced by Smyth and Goodman [128] and is considered to be one of the most promising measures [71] for rule induction.

Example

As an example of rule induction using the J-measure, we will use the previous problem of the patient database. If we allow overlapping rules then the following initial hypotheses can be formed either data-driven or model-driven (we have written the J-values behind the hypotheses and ordered the list for convenience):

- **If** Heart Rate is regular **And** Blood Pressure is normal **Then** Patient is healthy (0.5 bit)

- **If** Heart Rate is irregular **Then** Patient is ill (0.375 bit)
- **If** Blood Pressure is abnormal **Then** Patient is ill (0.25 bit)
- **If** Heart Rate is regular **Then** Patient is healthy (0.18 bit)
- **If** Blood Pressure is normal **Then** Patient is healthy (0.06 bit)
- ...

Clearly the first hypothesis has the highest J-value (0.5 bit) and we select this one for our rule base. All patients that are healthy are described (covered) by this rule. However, there are still patients left which are not yet covered by this rule base. Reviewing our list of hypotheses, we select the second hypothesis as the next rule to add to our rule base. Now only one patient remains uncovered, the ill patient with a regular heart rate and an abnormal blood pressure. For this patient we choose the third hypothesis to add to the rule base. Our rule base (or knowledge) does now cover all the examples and consists of the following rules:

- **If** Heart Rate is regular **And** Blood Pressure is normal **Then** Patient is healthy
- **If** Heart Rate is irregular **Then** Patient is ill
- **If** Blood Pressure is abnormal **Then** Patient is ill

Note that both the second and third rule cover patient number three of the database. It is said that the rules overlap or are non-disjoint. In this case they both have the same conclusion, and hence we have no conflict. However, conflicting rules make it sometimes necessary to form only disjoint rules. A disjoint scheme of rule induction essentially follows the same procedure. It iteratively generates hypotheses and selects the one having the highest J-value *and* that is disjoint with all existing rules in the rule base. In this case we would have obtained for the patients:

- **If** Heart Rate is regular **And** Blood Pressure is normal **Then** the Patient is healthy
- **If** Heart Rate is irregular **Then** the Patient is ill
- **If** Heart rate is regular **And** Blood Pressure is abnormal **Then** the Patient is ill

Note that these are exactly the same rules as we have obtained from the tree induction algorithm (after transformation to rules). In general, this is usually not the case; it is only due to the small number of examples and features present in this synthetic database.

Like tree induction, rule induction may also suffer from over-fitting. This is especially the case when disjoint rules are generated. Hence, the problem of

determining the (ir)relevance of a rule is also a major concern in rule induction algorithms. In the original ITrule algorithm of Smyth and Goodman, this problem was solved by letting the user specify the number n of rules to be generated, and the algorithm returns the nth best rules.

The ITrule algorithm has not been specifically designed for the classification task, although it can be used for it. The main motivation for ITrule was the problem of finding relations in a database (data mining)[4]. In rule induction approaches, such a relation can be easily evaluated by using the J-information measure. In tree induction approaches such a relation can not be directly evaluated, since a complete tree has to be constructed in order to evaluate the formed relations. For an overview of issues involving data mining we refer to [58, 95].

2.4 Conclusion

The main motivation for using rule induction is to obtain a mental fit to the decision-making problem. Such a mental fit makes an explanation of a decision possible. We have discussed the key elements in rule induction, where we paid special attention to the problem of data fit and mental fit. As a result we conclude that from a mental-fit point of view production rules are to be preferred over decision trees. Further, the rules should be based on qualifications rather than threshold concepts, and should be as simple as possible. Such simple rules can for example be obtained by using overlapping rules. From a data-fit point of view, however, the qualifications as formed by logical tests (sets) may be too restrictive. Further, a too simple rule base may ignore subtleties in the data and can be less accurate than a somewhat more complex rule base.

We ended this chapter by discussing existing information-theoretic approaches to rule induction. We showed that the J-measure is closely related to mutual information used in many of the successful decision-tree algorithms. The main advantage of the J-measure is that it allows the evaluation of a single rule. Another advantage of the J-measure is that the aspects of data fit as well as mental fit are represented by separate factors, which allow weighting between data fit and mental fit. Many other measures for rule-evaluation exist, but the J-measure is one of the few measures backed by information theory. The latter being of importance since it nicely fits the view that learning is a process of reducing the uncertainty in knowledge.

[4]It is often thought that data mining involves dealing with large databases, missing values, corrupted data etc. Although, these are all relevant topics in data mining, the essence of data mining is the discovery of relations between *any* of the features present in the database. In classification problems, the relations searched are between the features and the event space, in data mining any feature (or even multiple features) may form the event space. This sincerely increases the set of all possible hypotheses. In the previously used patient database, relations like: if heart rate is regular then blood pressure is normal, might have been also interesting.

Chapter 3

Kernel-Based Density Estimation

The statistical approach towards pattern-recognition problems is founded on the Bayes Decision Theory. In this theory it is assumed that all the class-conditional probability density functions $p(\mathbf{x}|C_i)$, $i \in \{1, 2, ...m\}$ in a d-dimensional feature space $\Re^d$ for feature vectors $\mathbf{x} \in \Re^d$ are known, as well as the a-priori probabilities $P(C_i)$. The *Bayes Rule* then states that the a posteriori probability is given by (in case of equal costs):

$$P(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)P(C_i)}{\sum_{i=1}^{m} p(\mathbf{x}|C_i)P(C_i)} \tag{3.1}$$

In order to minimize the probability of misclassifications, a class C_{max} is associated to the feature vector (pattern $\mathbf{x}$) that maximizes the a posteriori probability $C_{max}(\mathbf{x}) = \max_i\{P(C_i|\mathbf{x})\}$ (also known as: minimum-error-rate classification). More general, a set *discriminant functions* $g_i(\mathbf{x})$ can be defined such that $C_{max}(\mathbf{x}) = \max_i\{g_i(\mathbf{x})\}$. The problem is that the probability density functions (pdf's) or the discriminant functions (df's) are usually unknown, the basic approach is then to construct the pdf's or the df's from the data at hand. These approaches are traditionally considered to be conceptually different and are referred to as *density estimation* and *discriminant functions*, respectively[1]. Discriminant functions are considered to be a more direct approach, since it is not necessary to estimate the density functions first. Due to the "minimum-error-rate" criterion, a set of (parameterized) functions g_i can be optimized by directly minimizing the error of misclassifications without estimating density functions. Hence, discriminant functions lead to a characterization of the class-boundaries, but do not provide information on the within-class distribution, such that little

[1]This conceptual difference is vague because of two reasons. First, a density function can be considered as a special kind of discriminant function. Second, especially in the nonparametric case, there is a tendency to directly optimize the estimation of the pdf with respect to the minimum-error-rate criterion just like in the discriminant functions approach.

is known about the probability of the actual decision made by using discriminant functions. In some applications it is desirable to somehow indicate the quality (or certainty or probability) of the decision related to pattern $\mathbf{x}$. Clearly, the most important advantage of density estimation over discriminant functions, is that it does provide *locally* a quality of the decision related to the specific pattern. Several popular techniques exist for estimating the pdf or df, these can be subdivided in *parametric* techniques and *nonparametric* techniques. The difference between these two techniques is that the first assumes an underlying (parameterized) function, whereas the second is more general; it does not make explicit assumptions concerning the shape of the function to be estimated (although all techniques do make the implicit assumption that the function exists and that it is continuous). To our knowledge, all nonparametric techniques are somehow related to kernel-based techniques. For a classical overview of Pattern Recognition approaches using Bayes Decision Theory, the reader is referred to [31]. Recently there is a renewed interest in discriminant functions because of the *support vector algorithms* which, among other statistical techniques, can be found in [136].

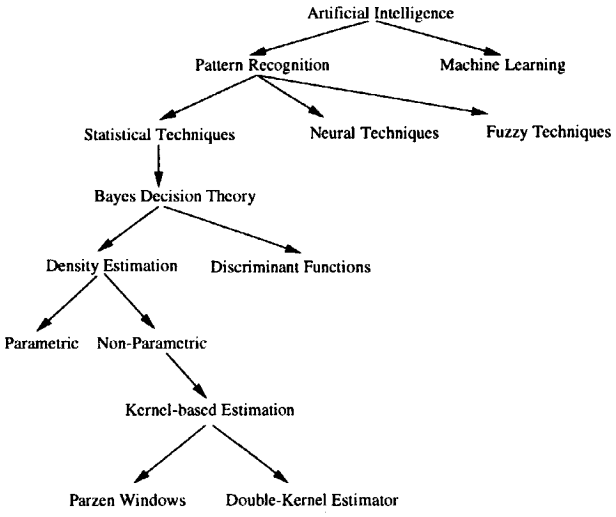


Figure 3.1: *Partial classification tree of Pattern Recognition.*

This chapter presents a novel approach to nonparametric density estimation: the double-kernel estimator, which is based on the well-known Parzen Windows technique. Before turning to the double-kernel estimator, we will first give a short overview of the Parzen Windows technique as an introduction to kernel-based density estimation and discuss some of its shortcomings.

3.1 Parzen Windows

Parzen Windows was introduced by Parzen [108] for the one-dimensional case and extended for the multivariate case by Murthy [100] and others. A good overview of Parzen Windows (PW) is given in [31].

In PW a density function $f(\mathbf{y})$ is used for which: $f(\mathbf{y}) > \mathbf{0}$, $\int f(\mathbf{y})d\mathbf{y} = 1$ and $\mathbf{y} \in \mathcal{R}^d$. Having d-dimensional data $\mathbf{x}_i \in I$, where $I \subset \mathcal{R}^d$, then the estimated pdf $\hat{p}(\mathbf{x})$ is obtained from:

$$\hat{p}(\mathbf{x}) = \frac{1}{h^d n} \sum_{i=1}^{n} f(\frac{\mathbf{x} - \mathbf{x}_i}{h}) \tag{3.2}$$

Where n is the total number of data, h is the "smoothing parameter" or window width, and the density function f is sometimes called the window. An example of a PW estimate is given in Figure 3.2.

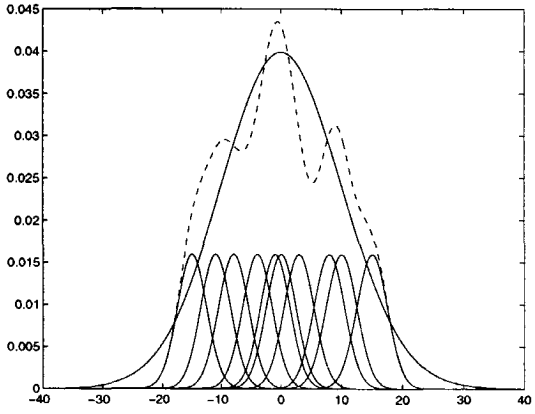


Figure 3.2: *Parzen Windows estimation of a Gaussian density. The large Gaussian is the theoretical density function. The estimation is based on 15 observations, each inducing a kernel (small Gaussians). The dashed line is the estimate that is obtained from a summation of the kernels.*

It can be shown that the mean and variance of the estimator for an unknown density $p(\mathbf{x})$ satisfy, see [31, 100, 108]:

$$\begin{aligned} E[\hat{p}(\mathbf{x})] &= \bar{\hat{p}}(\mathbf{x}) = \frac{1}{h^d} \int f(\frac{\mathbf{x} - \mathbf{v}}{h}) p(\mathbf{v}) d\mathbf{v} = \frac{1}{h^d} f(\frac{\mathbf{x}}{h}) * p(\mathbf{x}) \\ Var[\hat{p}(\mathbf{x})] &\leq \frac{\sup_{\mathbf{x}}\{f\}\bar{\hat{p}}(\mathbf{x})}{nh^d} \end{aligned} \tag{3.3}$$

where we use $*$ to denote a convolution. If we make the additional requirement (which holds for nearly all density functions f that may be used as a window):

$$\lim_{h \to 0} \frac{1}{h^d} f(\frac{\mathbf{x} - \mathbf{x}_i}{h}) = \delta(\mathbf{x} - \mathbf{x}_i) \tag{3.4}$$

where we used δ for the Dirac delta function, and if we also require that $h = h(n)$ is a function of n such that:

$$\begin{aligned} \lim_{n\to\infty} h^d(n) &= 0 \\ \lim_{n\to\infty} nh^d(n) &= \infty \end{aligned} \tag{3.5}$$

then the mean of the estimator converges to the exact pdf and the variance converges to zero (convergence in mean square) for an increasing number of data. It may also be observed that the best estimate for $p(\mathbf{x})$ is always a smoothed (filtered) version of the real pdf due to the convolution with the window, which should therefore have an as small as possible width. Unfortunately this is only possible if we have a sufficient number of data n such that the variance remains small.

In a somewhat different but more general notation we may write for (3.2):

$$\hat{p}(\mathbf{x}) = \frac{1}{V_\phi |H_\phi| n} \sum_{i=1}^{n} \phi(\frac{\mathbf{x} - \mathbf{x}_i}{H_\phi}) \tag{3.6}$$

Where we have used a (squared-integrable) d-dimensional separable kernel $\phi(\mathbf{x})$, and where $\mathbf{x} = (x_1, x_2, ..., x_d)$. For the kernel holds that :

$$\begin{gathered} \phi(\mathbf{x}) = \phi(x_1)\phi(x_2)...\phi(x_d) \\ \sup\{\phi\} = \phi(\mathbf{0}) = 1 \\ 0 \leq \phi(\mathbf{x}) \leq 1 \\ V_\phi = \int \phi(\mathbf{x}) d\mathbf{x} \\ 0 \leq V_\phi < \infty \end{gathered} \tag{3.7}$$

Further, we use H_ϕ as a general smoothing matrix (which is a function of n) allowing different smoothing in all dimensions, and we write $|H_\phi|$ for the determinant of H_ϕ. Clearly:

$$\int \phi(\frac{\mathbf{x}}{H_\phi}) d\mathbf{x} = V_\phi |H_\phi| \tag{3.8}$$

Also, to simplify the notation, we will use $\bar{p}$ instead of $\bar{\hat{p}}$ to denote the mean of the estimator $\hat{p}$. We will use the above notational conventions for the remainder of this chapter. With these conventions (3.3) becomes:

$$\begin{aligned} \bar{p}(\mathbf{x}) &= \frac{1}{V_\phi |H_\phi|} \mu(\frac{\mathbf{x}}{H_\phi}) * p(\mathbf{x}) \\ Var[\hat{p}(\mathbf{x})] &\leq \frac{\sup\{\phi\}\bar{p}(\mathbf{x})}{n V_\phi |H_\phi|} \end{aligned} \tag{3.9}$$

And the requirements for convergence become:

$$\begin{aligned} \lim_{H_\phi \to 0} \frac{1}{V_\phi |H_\phi|} \phi(\frac{\mathbf{x} - \mathbf{x}_i}{H_\phi}) &= \delta(\mathbf{x} - \mathbf{x}_i) \\ \lim_{n\to\infty} |H_\phi(n)| &= 0 \\ \lim_{n\to\infty} n|H_\phi(n)| &= \infty \end{aligned} \tag{3.10}$$

Although PW is a theoretically well understood technique, it has three "shortcomings".

First, in practice the number of data is limited and therefore the kernel can not be taken too small (otherwise "holes" and "spikes" appear in the estimated pdf). The optimal choice for the kernel and its width (often referred to as the smoothing-parameter) is a topic of research in itself [32, 61, 87, 125], however, in practice the Gaussian kernel is used and a width is chosen which optimizes the classification performance [115].

A second shortcoming is the *fixed-bandwidth* limitation of the PW approach; *fixed-bandwidth* refers to the single kernel-type by which the density is estimated. Especially if the density to be estimated can be considered as a mixture of several simple density functions it can be advantageous to use *locally adaptive* kernels, which allow for different kernels at different positions in the feature space. A few examples of these estimators can be found in literature, such as the balloon-estimator (more commonly known as the k-nearest-neighbor estimator) [84] and the sample-point estimator [14]. For some recent discussion on locally adaptive density estimation we refer to [118]. Also in locally-adaptive estimation the main problem is the choice of the kernels.

The third shortcoming of the PW approach, on which we will mainly focus in the remainder of this chapter, is the computational time and storage for estimating a pdf value. From equation (3.2) it is clear that for each pdf value at $\mathbf{x}$, all n kernels have to be evaluated and summed since the kernels are distributed according to the data (each datum "carries" its own kernel). A number of solutions have been proposed to reduce the computational load of which the majority comes down to a simple principle: reduce the number of kernels to be evaluated. In [47] the number of data is reduced by extracting a suitable subset, resulting in less data and, hence, less kernels to be evaluated. In [6, 140] the data are clustered and a weighted kernel estimator is used. In [43, 63, 120, 125] the pdf is reconstruction from a limited number of samples using an equidistant grid of kernels such as the binned-kernel estimator and the linear weighted estimator. Finally, it has been suggested to sample the PW estimate *afterwards*, but [63] shows that this is an inferior approach with respect to using an equidistant grid of kernels *prior* to estimation.

In this chapter we will describe a theoretical sound technique for designing kernel-based estimators that use an equidistant distribution of kernels prior to estimation of the pdf. It is shown that many approaches described in literature for reducing the number of kernels can be regarded as special cases. We will refer to this technique as the *Double-Kernel estimator* (uniform DK estimator, or DK estimator for short). Apart from describing and understanding existing approaches in more detail, the main advantage is that it leads to a new estimator which is based on a non-equidistant distribution of kernels. This *non-uniform Double-Kernel Estimator* leads to a large reduction of kernels without losing accuracy. In our analysis we will focus, without loss of generality, on the fixed-bandwidth kernel-based estimators, which form the majority of the kernel-based techniques described in literature.

3.2 Double-Kernel Estimator

The basic idea in the Double-Kernel technique is to distribute some points (samples) in the feature space, and then to obtain an estimated pdf from using kernels on these samples only. As will be shown, the DK approach can also be thought of as a re-distribution of the data into the samples using the first kernel followed by an estimation on the basis of the samples using the second kernel (hence a double-kernel technique). In this section we will examine an uniform distribution of kernels and in the next section we will examine the non-uniform case.

3.2.1 Basic Principle

Consider an equidistant lattice L of samples $\{\mathbf{x}_1 ... \mathbf{x}_s ... \mathbf{x}_m\}$, $L \subset \mathcal{R}^d$, where L can be described by the $d \times d$ sampling matrix S and a set $N \subset \mathcal{Z}^d$ of d-dimensional coefficients $\mathbf{n}_1, ... \mathbf{n}_s, ..., \mathbf{n}_m$ such that:

$$\mathbf{x}_s = S\mathbf{n}_s \tag{3.11}$$

The sampling matrix can be regarded as a matrix containing the (linear independent) basis vectors of the lattice L, and where N specifies the linear combinations in order to obtain all lattice points. Although S is not unique for a given lattice, the sample-volume Δ_S given by the determinant of sampling matrix $|S|$, is unique. (note that *different* lattices can have the same sample-volume). In case of one-dimensional sampling, S becomes a scalar and the volume is equal to this scalar (for a treatise on multidimensional sampling we refer to [86]). An example of a two-dimensional lattice is given in Figure 3.3.

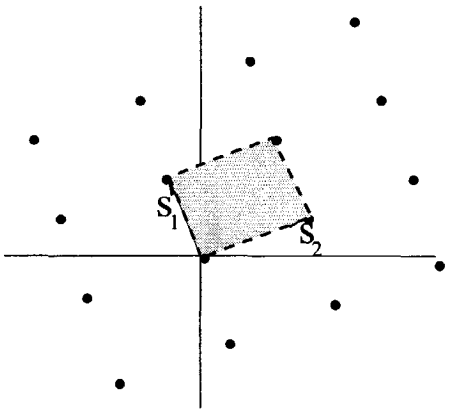


Figure 3.3: *A two-dimensional sampling lattice with basis vectors* $\mathbf{s}_1$ *and* $\mathbf{s}_2$. *The sample-volume is indicated by the rectangle.*

We can write from equation (3.6) the PW estimate for a sample $\mathbf{x}_s$ by using

a kernel μ (not necessarily different from the previously used kernel ϕ):

$$\hat{p}(\mathbf{x}_s) = \frac{1}{V_\mu |H_\mu| n} \sum_{i=1}^{n} \mu(\frac{\mathbf{x}_s - \mathbf{x}_i}{H_\mu}) \tag{3.12}$$

This is the summation of kernels μ positioned at $\mathbf{x}_i$ for a value at $\mathbf{x}_s$. If the kernels are symmetric, then:

$$\sum_{i=1}^{n} \mu(\frac{\mathbf{x}_s - \mathbf{x}_i}{H_\mu}) = \sum_{i=1}^{n} \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}) \tag{3.13}$$

This can be interpreted as a summation of kernel values of one kernel positioned at $\mathbf{x}_s$ instead of a summation of n kernels at position $\mathbf{x}_i$ (e.g. Figure 3.4). Although the result for $\hat{p}(x_s)$ from the left and right-hand side of equation (3.13) is exactly the same, the interpretation is different.

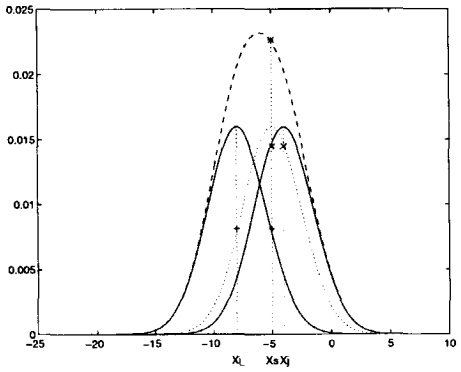


Figure 3.4: *One kernel at x_s (dotted kernel), or two kernels at x_i and x_j (left and right) lead to the same summed estimate at x_s. The dashed line is the summation of the left and right kernel.*

If we have a total of m different samples x_s of which $\hat{p}(\mathbf{x}_s)$ is known, then we may interpret this as a new data set. This new data set is in fact a re-distribution of the original data into a set of m different "virtual observations" $\mathbf{x}_s$, each not only observed once but several times, which can be estimated with a fractional number. This fractional number will be called the kernel-count (bin-count) and denoted by: $\hat{c}(\mathbf{x}_s)$. The kernel-count for a virtual observation is calculated by using the Parzen Windows estimate with kernel μ :

$$\hat{c}(\mathbf{x}_s) = n V_\mu |H_\mu| \hat{p}(\mathbf{x_s}) = \sum_{i=1}^{n} \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}) \tag{3.14}$$

Here, we multiplied with $nV_\mu|H_\mu|$ to obtain the absolute, de-normalized, kernel-count. Further, we also used (3.13) to obtain the last equality. The total number

of data m' in the new data set is then the sum of all the absolute kernel-counts:

$$m' = \sum_{s=1}^{m} \hat{c}(\mathbf{x}_s) \tag{3.15}$$

An estimation for $\hat{p}(\mathbf{x})$ can then be obtained from the m *weighted* or *modulated* observations by applying equation (3.6) again, but this time on the new data set with a kernel function ν. The uniform DK estimator then becomes:

$$\begin{aligned}
\hat{\hat{p}}(x) &= \frac{1}{V_\nu |H_\nu| m'} \sum_{s=1}^{m} \nu\left(\frac{\mathbf{x} - \mathbf{x}_s}{H_\nu}\right) \hat{c}(\mathbf{x}_s) \\
&= \frac{1}{V_\nu |H_\nu| m'} \sum_{s=1}^{m} \nu\left(\frac{\mathbf{x} - \mathbf{x}_s}{H_\nu}\right) \sum_{i=1}^{n} \mu\left(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}\right) \\
&= \frac{1}{V_\nu |H_\nu| m'} \sum_{s=1}^{m} \sum_{i=1}^{n} \nu\left(\frac{\mathbf{x} - \mathbf{x}_s}{H_\nu}\right) \mu\left(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}\right)
\end{aligned} \tag{3.16}$$

This is an intuitive way of deriving the basic (uniform) DK estimator. By using a distribution of kernels according to some samples in the feature space (a sampling distribution), it is possible to rewrite the PW approach into a summation of m weighted kernels. However, instead of choosing one kernel function, a user is now burdened with specifying two kernel functions and a sampling distribution. It will be shown in the next sections that, if the kernels and their distribution are chosen according to some criterions, the DK estimator is an equally well estimator as the PW estimator. An example of a DK estimation is given in Figure 3.5.

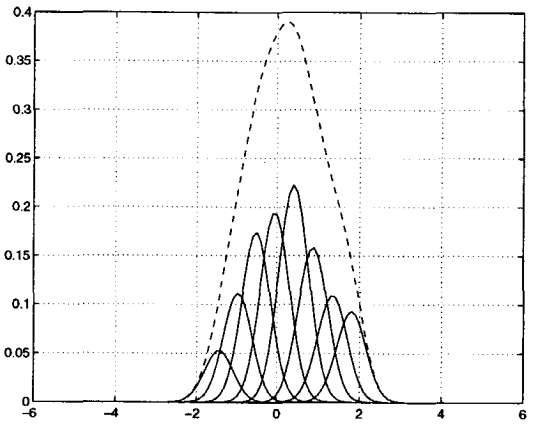


Figure 3.5: *A Double-Kernel estimation (dashed line) of a Gaussian using Gaussian kernels.*

3.2.2 Analysis

In order to arrive at the criterions mentioned in the previous section, some properties such as normalization, effects of sampling, mean and variance of the estimator need to be considered. In this section these properties will be studied and in the next section we will turn to the problem of choosing kernel functions and samples by using the properties studied here. The reader is referred to Appendix B for a more general class of DK estimators which also provides a basis for designing *locally adaptive* DK estimators.

A well-defined pdf estimator should itself be a density function. Hence, the uniform DK estimator should integrate to one. Integration of (3.16) gives:

$$\begin{aligned}\int \hat{\hat{p}}(x)d\mathbf{x} &= \int \frac{1}{V_\nu|H_\nu|m'}\sum_{s=1}^{m}\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\hat{c}(\mathbf{x}_s)d\mathbf{x} \\ &= \frac{1}{V_\nu|H_\nu|m'}\sum_{s=1}^{m}\int \nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})d\mathbf{x}\hat{c}(\mathbf{x}_s) \\ &= \frac{1}{V_\nu|H_\nu|m'}\sum_{s=1}^{m}V_\nu|H_\nu|\hat{c}(\mathbf{x}_s) \end{aligned} \tag{3.17}$$

where (3.8) has been used. By using (3.15), it follows:

$$\int \hat{\hat{p}}(x)d\mathbf{x} = \frac{V_\nu|H_\nu|}{V_\nu|H_\nu|m'}\sum_{s=1}^{m}\hat{c}(\mathbf{x}_s) = \frac{m'}{m'} = 1 \tag{3.18}$$

To understand the effects of the number of samples, two limit-cases are considered. Suppose only a single sample $\bar{\mathbf{x}}$ is used then (3.16) becomes:

$$\begin{aligned}\hat{\hat{p}}(\mathbf{x}) &= \frac{1}{V_\nu|H_\nu|m'}\sum_{s=1}^{1}\sum_{i=1}^{n}\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\mu(\frac{\mathbf{x}_i-\mathbf{x}_s}{H_\mu}) \\ &= \frac{1}{V_\nu|H_\nu|m'}\nu(\frac{\mathbf{x}-\bar{\mathbf{x}}}{H_\nu})\sum_{i=1}^{n}\mu(\frac{\mathbf{x}_i-\bar{\mathbf{x}}}{H_\mu}) \\ &= \frac{1}{V_\nu|H_\nu|\sum_{i=1}^{n}\mu(\frac{\mathbf{x}_i-\bar{\mathbf{x}}}{H_\mu})}\nu(\frac{\mathbf{x}-\bar{\mathbf{x}}}{H_\nu})\sum_{i=1}^{n}\mu(\frac{\mathbf{x}_i-\bar{\mathbf{x}}}{H_\mu}) \\ &= \frac{1}{V_\nu|H_\nu|}\nu(\frac{\mathbf{x}-\bar{\mathbf{x}}}{H_\nu}) \end{aligned} \tag{3.19}$$

Thus, for one sample only, the uniform DK estimator reduces to a biased *parametric* density estimator, e.g. a Gaussian estimate can be obtained by choosing a Gaussian kernel, with covariance matrix H_ν and choosing the mean of the data as the only sample $\bar{\mathbf{x}} = \bar{\mathbf{x}}_i$. On the other hand, in the case of an infinite sampling frequency, i.e. (see also (3.11))

$$\begin{aligned} m &\to \infty \\ \Delta_S &\to 0 \end{aligned} \tag{3.20}$$

then the DK estimate becomes, using (3.16):

$$\begin{aligned}\lim_{m\to\infty,\Delta_S\to 0}\hat{\hat{p}}(\mathbf{x}) &= \lim_{m\to\infty,\Delta_S\to 0}\frac{1}{V_\nu|H_\nu|m'}\sum_{s=1}^{m}\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\hat{c}(\mathbf{x}_s)\\ &= \lim_{m\to\infty,\Delta_S\to 0}\frac{1}{V_\nu|H_\nu|m'\Delta_S}\sum_{s=1}^{m}\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\hat{c}(\mathbf{x}_s)\Delta_S\end{aligned} \tag{3.21}$$

observing that (by using (3.14) and (3.15)):

$$\begin{aligned}\lim_{m\to\infty,\Delta_S\to 0} m'\Delta_S &= \sum_{i=1}^{n}\int\mu(\frac{\mathbf{x}_i-\mathbf{x}_s}{H_\mu})d\mathbf{x}_s\\ &= nV_\mu|H_\mu|\end{aligned} \tag{3.22}$$

and:

$$\begin{aligned}\lim_{m\to\infty,\Delta_S\to 0}\sum_{s=1}^{m}\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\hat{c}(\mathbf{x}_s)\Delta_S &= \int\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\hat{c}(\mathbf{x}_s)d\mathbf{x}_s\\ &= \nu(\frac{\mathbf{x}}{H_\nu})*\hat{c}(\mathbf{x})\end{aligned} \tag{3.23}$$

(3.21) then becomes:

$$\begin{aligned}\lim_{m\to\infty,\Delta_S\to 0}\hat{\hat{p}}(\mathbf{x}) &= \frac{1}{V_\nu|H_\nu|nV_\mu|H_\mu|}\nu(\frac{\mathbf{x}}{H_\nu})*\hat{c}(\mathbf{x})\\ &= \frac{1}{V_\nu|H_\nu|nV_\mu|H_\mu|}\sum_{i=1}^{n}\nu(\frac{\mathbf{x}}{H_\nu})*\mu(\frac{\mathbf{x}-\mathbf{x}_i}{H_\mu})\end{aligned} \tag{3.24}$$

Thus in the limit case, the DK estimator becomes a Parzen Windows estimator with kernel:

$$\phi(\frac{\mathbf{x}}{H_\phi}) = \nu(\frac{\mathbf{x}}{H_\nu})*\mu(\frac{\mathbf{x}}{H_\mu}) \tag{3.25}$$

This implies that, if the PW estimator converges to the unknown density estimator, then also the DK estimator using an unlimited number of samples will converge. In more detail this is proven for the one-dimensional case in Appendix A. Hence, here we have the first important result that, in the limit-case of infinite sampling, the uniform DK estimator converges. However, the results obtained for the effects of sampling in the limit-cases are practically of little interest, since we do not want to use a biased estimator nor an infinite number of samples. Therefore the effects of sampling in between the two limit-cases have to be studied. To this purpose the mean and variance of the uniform DK estimator are studied (the analysis is based on the analogy of the analysis of Parzen Windows).

For obtaining the mean and variance, the data $\mathbf{x}_i$ are treated as i.i.d.: independent random variables identically distributed according to a data density $p(\mathbf{x})$, whereas the samples $\mathbf{x}_s$ are treated as random variables distributed according to a sampling density $p_S(\mathbf{x})$, which is required to be uniform, i.e.:

$$p_S(\mathbf{x}) = \frac{1}{m\Delta_S} \tag{3.26}$$

where Δ_S is the sample-volume. Further, the sampling density is independent of the data density such that the joint density $p_J(\mathbf{x}_s, \mathbf{x}_i)$ may be written as $p_S(\mathbf{x})p(\mathbf{x})$. Using (3.16) and requiring that m' is a constant for all $\mathbf{x}_i, \mathbf{x}_s$, the mean of the uniform DK estimator is then:

$$\begin{aligned} \bar{\hat{p}}(\mathbf{x}) &= E_{(\mathbf{x}_s,\mathbf{x}_i)}[\hat{p}(x)] \\ &= \frac{1}{V_\nu |H_\nu| m'} \sum_{s=1}^{m} \sum_{i=1}^{n} E_{(\mathbf{x}_s,\mathbf{x}_i)} \left[\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu}) \mu(\frac{\mathbf{x}_i-\mathbf{x}_s}{H_\mu}) \right] \end{aligned} \tag{3.27}$$

substituting $\mathbf{v}$ for $\mathbf{x}_s$ and $\mathbf{w}$ for $\mathbf{x}_i$, using the fact that kernel μ is symmetric and using $*$ to denote a convolution:

$$\begin{aligned} \bar{\hat{p}}(\mathbf{x}) &= \frac{mn}{V_\nu |H_\nu| m'} \int\int \nu(\frac{\mathbf{x}-\mathbf{v}}{H_\nu}) \mu(\frac{\mathbf{w}-\mathbf{v}}{H_\mu}) p(\mathbf{w}) p_S(\mathbf{v}) d\mathbf{w} d\mathbf{v} \\ &= \frac{mn}{V_\nu |H_\nu| m' m \Delta_S} \int\int \nu(\frac{\mathbf{x}-\mathbf{v}}{H_\nu}) \mu(\frac{\mathbf{w}-\mathbf{v}}{H_\mu}) p(\mathbf{w}) d\mathbf{w} d\mathbf{v} \\ &= \frac{n}{V_\nu |H_\nu| m' \Delta_S} \nu(\frac{\mathbf{x}}{H_\nu}) * \mu(\frac{\mathbf{x}}{H_\mu}) * p(\mathbf{x}) \end{aligned} \tag{3.28}$$

This equation shows that the mean (or expectation) of the uniform DK estimator is a double convolution between the unknown density $p(\mathbf{x})$ and the two kernels ν and μ. In order to normalize the expectation, i.e. that it integrates to one, Δ_S should be chosen such that:

$$m'\Delta_S = nV_\mu |H_\mu| \tag{3.29}$$

which can be satisfied by requiring:

$$\sum_{s=1}^{m} \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}) = \frac{V_\mu |H_\mu|}{\Delta_S} \; \forall \; \mathbf{x}_i \tag{3.30}$$

To see this, substitute (3.30) in $m'\Delta_S$:

$$\begin{aligned} m'\Delta_S &= \sum_{i=1}^{n} \sum_{s=1}^{m} \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}) \Delta_S \\ &= \sum_{i=1}^{n} \frac{V_\mu |H_\mu|}{\Delta_S} \Delta_S \\ &= nV_\mu |H_\mu| \end{aligned} \tag{3.31}$$

which indeed satisfies (3.29). Using (3.29) in (3.28), the expectation becomes:

$$\bar{\hat{p}}(\mathbf{x}) = \frac{1}{V_\nu |H_\nu| V_\mu |H_\mu|} \nu(\frac{\mathbf{x}}{H_\nu}) * \mu(\frac{\mathbf{x}}{H_\mu}) * p(\mathbf{x}) \tag{3.32}$$

As in the case of unlimited sampling, it is observed that the uniform DK estimator is in principle equivalent with a Parzen Windows estimator with a kernel function ϕ satisfying (3.25). Due to the convolutions, a smoothed version of the unknown density $p(\mathbf{x})$ is obtained. Only if for ν and μ δ-functions are chosen will $\bar{\hat{p}}(\mathbf{x})$ equal $p(\mathbf{x})$. Since this is only true *on average* (as expressed by the expectation), it is necessary to consider the variance, using (3.16) :

$$Var[\hat{\hat{p}}(\mathbf{x})] = Var\left[\sum_{s=1}^{m}\sum_{i=1}^{n} \frac{1}{V_\nu |H_\nu| m'} \nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu}) \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu})\right] \tag{3.33}$$

Since this is the variance of the sum of functions of statistically independent random variables (the data), it equals to the sum of the variance of the separate terms:

$$\begin{aligned} Var[\hat{\hat{p}}(\mathbf{x})] &= \sum_{i=1}^{n} E_{(\mathbf{x}_s,\mathbf{x}_i)}\left[\left(\sum_{s=1}^{m} \frac{1}{V_\nu |H_\nu| m'} \nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu}) \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}) - \frac{1}{n}\bar{\hat{p}}(\mathbf{x})\right)^2\right] \\ &= nE_{(\mathbf{x}_s,\mathbf{x}_i)}\left[\frac{1}{V_\nu |H_\nu| m'} \sum_{s=1}^{m} \nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu}) \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu})\right]^2 - \frac{1}{n}\bar{\hat{p}}^2(\mathbf{x}) \end{aligned} \tag{3.34}$$

neglecting the second term,

$$\begin{aligned} Var[\hat{\hat{p}}(\mathbf{x})] &\leq nE_{(\mathbf{x}_s,\mathbf{x}_i)}\left[\frac{1}{V_\nu |H_\nu| m'} \sum_{s=1}^{m} \nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu}) \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu})\right]^2 \\ &\leq nE_{(\mathbf{x}_s,\mathbf{x}_i)}\left[M \frac{1}{V_\nu |H_\nu| m'} \sum_{s=1}^{m} \nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu}) \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu})\right] \end{aligned} \tag{3.35}$$

where:

$$M = \frac{1}{V_\nu |H_\nu| m'} \sum_{s=1}^{m} \nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu}) \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}) \tag{3.36}$$

We can further analyze the variance by using the following well-known inequalities:

$$\int g(x)f(x)dx \leq \sup_x\{g\} \int f(x)dx \tag{3.37}$$

$$\sum_i^n y_i x_i \leq \sup_i\{y\} \sum_i^n x_i \tag{3.38}$$

where with $\sup_x\{g\}$ we denote the supremum of the set of values obtained from $g(x)$ by varying x: an upperbound (maximum) which is not necessarily an element of $\{g\}$.

Applying the integral inequality to the variance (bounding M), and substituting v and w gives:

$$\begin{aligned} Var[\hat{\bar{p}}(\mathbf{x})] &\leq \sup_{(\mathbf{x}_s,\mathbf{x}_i)}\{M\} nE_{(\mathbf{x},\mathbf{x}_s,\mathbf{x}_i)}\left[\frac{1}{V_\nu|H_\nu|m'}\sum_{s=1}^{m}\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\mu(\frac{\mathbf{x}_i-\mathbf{x}_s}{H_\mu})\right] \\ &\leq \frac{\sup_{(\mathbf{x},\mathbf{x}_s,\mathbf{x}_i)}\{M\}nm}{V\nu|H_\nu|m'}\int\int\nu(\frac{\mathbf{x}-\mathbf{v}}{H_\nu})\mu(\frac{\mathbf{w}-\mathbf{v}}{H_\mu})p(\mathbf{w})p_S(\mathbf{v})d\mathbf{w}d\mathbf{v} \end{aligned} \tag{3.39}$$

using (3.28):

$$Var[\hat{\bar{p}}(\mathbf{x})] \leq \sup_{(\mathbf{x},\mathbf{x}_s,\mathbf{x}_i)}\{M\}\bar{\bar{p}}(\mathbf{x}) \tag{3.40}$$

since, by using the summation inequation of (3.38),

$$\begin{aligned} \sup_{(\mathbf{x},\mathbf{x}_s,\mathbf{x}_i)}\{M\} &\leq \sup_{(\mathbf{x},\mathbf{x}_s,\mathbf{x}_i)}\left\{\frac{1}{V_\nu|H_\nu|m'}\sum_{s=1}^{m}\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\mu(\frac{\mathbf{x}_i-\mathbf{x}_s}{H_\mu})\right\} \\ &\leq \sup_{\mathbf{x}}\{\nu\}\frac{1}{V_\nu|H_\nu|m'}\sum_{s=1}^{m}\mu(\frac{\mathbf{x}_i-\mathbf{x}_s}{H_\mu}) \\ &\leq \sup_{\mathbf{x}}\{\nu\}\frac{1}{V_\nu|H_\nu|n} \end{aligned} \tag{3.41}$$

where the last inequality comes from using condition (3.30) and its implication (3.29). Hence, (3.40) finally becomes:

$$Var[\hat{\bar{p}}(\mathbf{x})] \leq \sup\{\nu\}\frac{1}{V_\nu|H_\nu|n}\bar{\bar{p}}(\mathbf{x}) \tag{3.42}$$

This variance upper-bound is in principle equivalent to the variance upper-bound of the Parzen Windows estimator (3.9). So by requiring:

$$\begin{gathered} \textstyle\sum_{s=1}^{m}\mu(\frac{\mathbf{x}_i-\mathbf{x}_s}{H_\mu}) = \frac{V_\mu|H_\mu|}{\Delta_S}\ \forall\ \mathbf{x}_i \\ \lim_{n\to\infty}|H_\mu(n)| = 0 \\ \lim_{n\to\infty}|H_\nu(n)| = 0 \\ \lim_{n\to\infty} n|H_\nu(n)| = \infty \end{gathered}$$

it is said that the uniform DK estimator converges to the unknown density function (convergence in mean square [107]). Since under these conditions the mean given in (3.32) and the variance given in (3.42) become:

$$\begin{aligned} E[\hat{\bar{p}}(\mathbf{x})] &\to p(\mathbf{x}) \\ Var[\hat{\bar{p}}(\mathbf{x})] &\to 0 \end{aligned} \tag{3.43}$$

On first sight one may think that this convergence holds *independent* of the number of samples. This is not true, since one of the requirements is that:

$$\sum_{s=1}^{m} \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}) = \frac{V_\mu |H_\mu|}{\Delta_S} \tag{3.44}$$

As we will see in the next section, this requirement can only be fulfilled by placing conditions on:

- the distribution of the kernels (samples),
- the type of kernels.

3.2.3 Choosing the Kernels and their Distribution

Several estimators can be constructed from equation (3.16), of which some are well-known in literature, such as the binned-kernel estimator and the linear-weighted kernels estimator. However, they have never been presented in the DK form. To see how these kernel-based estimators are constructed we will focus our discussion on DK estimators having an expectation and variance equivalent to the Parzen Windows estimator.

To construct an estimator having an equivalent expectation and variance, two conditions have to be satisfied:

$$\sum_{s=1}^{m} \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}) = \frac{V_\mu |H_\mu|}{\Delta_S} \; \forall \; \mathbf{x}_i \tag{3.45}$$

$$\frac{1}{V_\nu |H_\nu| V_\mu |H_\mu|} \nu(\frac{\mathbf{x}}{H_\nu}) * \mu(\frac{\mathbf{x}}{H_\mu}) = \frac{1}{V_\nu |H_\nu|} \nu(\frac{\mathbf{x}}{H_\nu}) \tag{3.46}$$

Using these conditions then (3.28) becomes, by noting that (3.29) holds,

$$\frac{n}{V_\nu |H_\nu| m' \Delta_S} \nu(\frac{\mathbf{x}}{H_\nu}) * \mu(\frac{\mathbf{x}}{H_\mu}) * p(\mathbf{x}) = \frac{1}{V_\nu |H_\nu|} \nu(\frac{\mathbf{x}}{H_\nu}) * p(\mathbf{x})$$

which is the expectation of the PW estimator (3.9) with kernel ν. As we have seen in the previous section, the variance (3.40) is also equivalent to the variance of the PW estimator (3.9) if the first condition (3.45) holds. The question now becomes, for which kernels and distributions can these two conditions be satisfied? Let us focus on condition (3.45) first.

The first condition basically states that the summation of over all the distributed kernels should be constant. As we show in detail in Appendix C, if the sampling matrix S is chosen as $S = V_\mu^{\frac{1}{d}} H_\mu$ then

$$\Delta_S = |S| = V_\mu |H_\mu| \tag{3.47}$$

and for *any* reasonable kernel it holds that:

$$\sum_{s=1}^{m} \mu(\frac{\mathbf{x}_s - \mathbf{x}_i}{H_\mu}) \approx 1 \quad \forall \quad \mathbf{x}_i \tag{3.48}$$

and condition (3.45) is satisfied. Figure (3.6) shows one-dimensional examples of the simple and popular "bin" and "linear-weighted" or triangle kernels with $V_\mu = 1$ and $\Delta_S = H_\mu = (\mathbf{x}_{s+1} - \mathbf{x}_s)$. With the choice of the "bin" the binned-kernel estimator is obtained and with the choice of the triangular kernel the linear-weighted estimator is obtained. Estimators of this kind have been described by several authors [43, 54, 63, 120, 125].

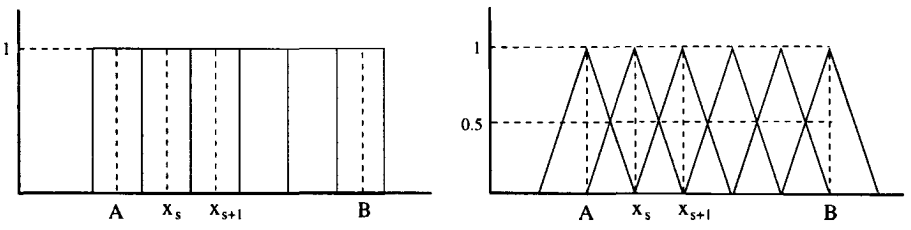


Figure 3.6: *Two kernel types μ, a bin and a triangular kernel satisfying conditions (3.48),(3.47) for* $\mathbf{x_i} \in [A, B]$.

The second condition (3.46) simply means that filtering ν with μ has no effect on ν (perfect filtering). Hence, the Fourier-transform of μ should be uniform wherever the Fourier-transform of ν exists. As is shown in Appendix C μ has a bandwidth D_ω which equals $\frac{2\pi}{S}$. If we denote D_{ω_c} as the cut-off bandwidth of ν then for perfect filtering D_{ω_c} should be completely contained within $\frac{2\pi}{S}$:

$$D_{\omega_c} \leq \frac{2\pi}{S} \tag{3.49}$$

We have now formalized the two conditions (3.47 and 3.49) under which the uniform DK estimator results in an equivalent estimator as the Parzen Windows estimator. Given a kernel ν with which we want to obtain an estimate equivalent to the Parzen Windows estimate we need to choose S according to the equality sign of (3.49) in order to obtain the minimum number of samples. The sampling matrix then becomes:

$$S = \frac{2\pi}{D_{\omega_c}} \tag{3.50}$$

Which is in complete accordance with multidimensional sampling theory, and this frequency is known as the Nyquist-density for the function ν [86] (note that we have used here the cut-off bandwidth which equals twice the cut-off frequency). We might also have reasoned that, since the uniform DK estimator is a linear combination of the kernel ν, the highest frequency present in the

estimate of $p(\mathbf{x})$ is the cut-off frequency $0.5D_{\omega_c}$ of the kernel ν. Therefore, for reconstruction from samples, it is sufficient to sample with the Nyquist-density of ν. Along this line of thought, kernel μ can be regarded as a *pre-filter* for ν in order to prevent aliasing.

Concluding, a Parzen Windows estimate with kernel ν of an unknown density $p(\mathbf{x})$ is obtained if we choose the sampling matrix according to (3.50) and a kernel-width matrix H_μ according to (3.47):

$$H_\mu = \frac{S}{V_\mu^{\frac{1}{d}}} \tag{3.51}$$

This is a useful result, since we now have related the kernel and smoothing-parameter of the Parzen Windows estimator to:

- a kernel ν,
- a set of smoothing-parameters for the kernels ν, μ,
- a sampling matrix S.

such that the uniform DK estimator is always equivalent to the Parzen Windows estimator. The only degree of freedom that remains is the choice of kernel μ. In principle it does not really matter which kernel we choose as long as the duration of μ is equal to the sampling matrix. However, there is a practical restriction to the choice of kernel μ. A close analysis of condition (3.48) (see Appendix C) reveals that this holds for reasonable kernels *only if* the samples are taken everywhere where $\mu(\frac{\mathbf{x}_i - \mathbf{x}}{H_\mu})$ exists. Suppose we would allow to choose for kernel μ a perfect pre-filter (a sinc-function) then from the pre-filter point of view this is an optimal choice, however since in the spatial-domain the sinc-function $\mu(\frac{\mathbf{x}_i - \mathbf{x}}{H_\mu})$ exists everywhere we need to sample the complete spatial-domain which leads to an infinite number of samples. On the other hand, if we would choose a block-function such that we only need a limited number of samples (since the block-function only exists for its duration around x_i) then from the pre-filtering point of view we do not have a perfect pre-filter (only in a first order approximation as shown in Appendix C). Therefore, we look for a kernel μ having both a perfect finite bandwidth and a perfect finite duration. The only function which satisfies both conditions equally well is the Gaussian kernel. For this reason, the Gaussian kernel seems to be a logical choice for μ. With this final choice, our quest for the optimal uniform DK estimator is completed.

3.3 Non-Uniform Distributions

The uniform DK estimator with settings as discussed in the previous section leads to an excellent approximation of the Parzen Windows estimator, as can also be observed in the experiments. However, it requires an equidistant sampling grid which in a small number of dimensions does not take a lot of samples but may become astronomical in higher dimensions. Consider for example an

equidistant grid in one dimension containing $m = 5$ samples which is not a lot of samples. The same grid in $d = 5$-dimensions would require $d^m = 3125$ samples which is a lot. Typically, the number of data in more-dimensional problems is often sparse. The problem of keeping the number of samples small even in high-dimensional problems with sparse data is addressed in this section.

Before turning to a general *non-equidistant sampling* scheme, consider the uniform DK estimator again where μ is chosen to be a bin-function $\beta(\frac{\mathbf{x}}{H_\beta})$, a binned-kernel estimator, such that:

$$\beta_j(\frac{\mathbf{x}}{H_\beta}) = \begin{cases} 1 & for \quad |\frac{(\mathbf{x})}{H_\beta}| <= 1 \\ 0 & otherwise \end{cases} \tag{3.52}$$

Clearly, if a bin positioned at $\mathbf{x}_s$ does not contain any data then $\beta(\mathbf{x}_s) = 0$ and we may discard the sample. In this way, the number of samples can be reasonably reduced. Since we accept only samples if their bin is non-empty, then in the case of very small bins we can maximally accept n samples (n equal to the number of data), and in the case of one large bin we accept only one sample. Hence, the number of samples is always smaller than or equal to the number of data ($m \leq n$). Using equidistant sampling, the binned-kernel estimator leads to the minimum number of samples possible, and despite its rather poor pre-filter properties, leads to good results. For an even lower number of samples we must look at non-equidistant sampling schemes.

To arrive at a non-equidistant sampling scheme, consider first the kernel χ defined as:

$$\chi(\mathbf{x} - \mathbf{x}_i) = \frac{\mu(\frac{\mathbf{x}-\mathbf{x}_i}{H_\mu})}{\sum_{s=1}^{m} \mu(\frac{\mathbf{x}_s-\mathbf{x}_i}{H_\mu})} \tag{3.53}$$

Note that χ is not strictly a kernel according to our kernel definition in (3.7) since in general:

$$\sup\{\chi\} \leq 1 \tag{3.54}$$

However, we have mainly used the kernel definition for ease of analysis and we will derive the properties for χ in the following. Since χ is a function of its local position $\mathbf{x}_i$, χ can be thought of as a locally adapted kernel μ. Clearly, if the samples are placed on an equidistant grid then $\chi = \mu$ due to (3.48). However, for *any* sampling grid it holds that:

$$\sum_{i=1}^{n}\sum_{s=1}^{m} \chi(\mathbf{x}_s - \mathbf{x}_i) = \sum_{i=1}^{n}\sum_{s=1}^{m} \frac{\mu(\frac{\mathbf{x}_s-\mathbf{x}_i}{H_\mu})}{\sum_{s=1}^{m} \mu(\frac{\mathbf{x}_s-\mathbf{x}_i}{H_\mu})} = \sum_{i=1}^{n} 1 = n \tag{3.55}$$

Also, for any sampling grid it holds that:

$$\begin{aligned}\Delta_{\mathbf{x}} &= \frac{1}{sup\{\chi\}}\int \chi(\mathbf{x}-\mathbf{x}_i)d\mathbf{x} \\ &= \frac{\sum_{s=1}^{m}\mu(\frac{\mathbf{x}_s-\mathbf{x}_i}{H_\mu})}{\mu(\mathbf{0})}\int \frac{\mu(\frac{\mathbf{x}-\mathbf{x}_i}{H_\mu})}{\sum_{s=1}^{m}\mu(\frac{\mathbf{x}_s-\mathbf{x}_i}{H_\mu})}d\mathbf{x} \\ &= V_\mu|H_\mu| \end{aligned} \tag{3.56}$$

This is a useful result since it means that the duration-density of χ always equals the duration-density of μ. Implying that, in a first order approximation, χ is an equally well pre-filter for ν as μ is.

Consider now a sampling density function $p_S(\mathbf{x})$, then we can estimate this sampling density by applying a Parzen Windows estimator on m *random* samples $\mathbf{x}_s$:

$$\hat{p}_S(\mathbf{x}) = \frac{1}{mV_\mu|H_\mu|}\sum_{s=1}^{m}\mu(\frac{\mathbf{x}_s-\mathbf{x}}{H_\mu}) \tag{3.57}$$

The kernel χ may now be written by using the estimate $\hat{p}_S(\mathbf{x})$ at the locations $\mathbf{x}_i$ (hence, $\hat{p}_S(\mathbf{x}_i)$) as:

$$\chi(\mathbf{x}-\mathbf{x}_i) = \frac{\mu(\frac{\mathbf{x}-\mathbf{x}_i}{H_\mu})}{mV_\mu|H_\mu|\hat{p}_S(\mathbf{x}_i)} \tag{3.58}$$

Using the kernel χ to obtain the expectation of the non-uniform DK estimator we get:

$$\begin{aligned}\bar{\bar{p}}(\mathbf{x}) &= E_{(\mathbf{x}_s,\mathbf{x}_i)}[\hat{\bar{p}}(x)] \\ &= \frac{1}{V_\nu|H_\nu|n}\sum_{s=1}^{m}\sum_{i=1}^{n}E_{(\mathbf{x}_s,\mathbf{x}_i)}\left[\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\frac{\mu(\frac{\mathbf{x}_s-\mathbf{x}_i}{H_\mu})}{mV_\mu|H_\mu|\hat{p}_S(\mathbf{x}_i)}\right] \end{aligned} \tag{3.59}$$

substituting $\mathbf{v}$ for $\mathbf{x}_s$ and $\mathbf{w}$ for $\mathbf{x}_i$, using the fact that kernel μ is symmetric and using $*$ to denote a convolution:

$$\begin{aligned}\bar{\bar{p}}(\mathbf{x}) &= \frac{mn}{V_\nu|H_\nu|n}\int \nu(\frac{\mathbf{x}-\mathbf{v}}{H_\nu})\int \frac{\mu(\frac{\mathbf{w}-\mathbf{v}}{H_\mu})}{mV_\mu|H_\mu|\hat{p}_S(\mathbf{w})}p(\mathbf{w})d\mathbf{w}p_S(\mathbf{v})d\mathbf{v} \\ &= \frac{1}{V_\nu|H_\nu|V_\mu|H_\mu|}\nu(\frac{\mathbf{x}}{H_\nu}) * \left(\left[\mu(\frac{\mathbf{x}}{H_\mu}) * \frac{p(\mathbf{x})}{\hat{p}_S(\mathbf{x})}\right]p_S(\mathbf{x})\right) \end{aligned} \tag{3.60}$$

changing the order of the convolution and the product (both being linear operators; see Appendix D):

$$\begin{aligned}\bar{\bar{p}}(\mathbf{x}) &= \frac{1}{V_\nu|H_\nu|V_\mu|H_\mu|}\nu(\frac{\mathbf{x}}{H_\nu}) * \mu(\frac{\mathbf{x}}{H_\mu}) * \frac{p(\mathbf{x})}{\hat{p}_S(\mathbf{x})}p_S(\mathbf{x}) \\ &= \frac{1}{V_\nu|H_\nu|V_\mu|H_\mu|}\nu(\frac{\mathbf{x}}{H_\nu}) * \mu(\frac{\mathbf{x}}{H_\mu}) * p(\mathbf{x})\frac{p_S(\mathbf{x})}{\hat{p}_S(\mathbf{x})} \end{aligned} \tag{3.61}$$

Under the assumption that $\hat{p}_S(\mathbf{x})$ is a good estimator for the sampling density over the entire space where $p(\mathbf{x})$ exists:

$$\hat{p}_S(\mathbf{x}) \approx p_S(\mathbf{x}) \tag{3.62}$$

then we obtain:

$$\bar{\hat{p}}(\mathbf{x}) = \frac{1}{V_\nu|H_\nu|V_\mu|H_\mu|}\nu(\frac{\mathbf{x}}{H_\nu}) * \mu(\frac{\mathbf{x}}{H_\mu}) * p(\mathbf{x}) \tag{3.63}$$

which is the mean we already obtained in the previous section. This is the most important result obtained for the DK estimator. Basically it states that no matter how the sampling is performed, we can always obtain the expectation of the Parzen Windows estimator as long as we *adapt* the kernel μ with the estimate of the sampling density such that the effects of the non-uniform sampling density are canceled. Clearly, efficient adaptation can only be obtained if $\hat{p}_S(\mathbf{x})$ is a good estimator for the sampling density. To complete the analysis, the variance can be calculated by using (3.40) where M is given by:

$$M = \frac{1}{V_\nu|H_\nu|n}\sum_{s=1}^{m}\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\chi(\mathbf{x}_i-\mathbf{x}_s) \tag{3.64}$$

such that:

$$\begin{aligned}
\sup_{(\mathbf{x}_s,\mathbf{x}_i)}\{M\} &\leq \sup_{(\mathbf{x}_s,\mathbf{x}_i)}\left\{\frac{1}{V_\nu|H_\nu|n}\sum_{s=1}^{m}\nu(\frac{\mathbf{x}-\mathbf{x}_s}{H_\nu})\chi(\mathbf{x}_i-\mathbf{x}_s)\right\} \\
&\leq \sup\{\nu\}\frac{1}{V_\nu|H_\nu|n}\sum_{s=1}^{m}\chi(\mathbf{x}_i-\mathbf{x}_s) \\
&\leq \sup\{\nu\}\frac{1}{V_\nu|H_\nu|n}
\end{aligned} \tag{3.65}$$

where the last inequality is obtained by using (3.55). The variance then becomes the variance of the Parzen Windows estimator:

$$Var[\hat{\hat{p}}(\mathbf{x})] \leq \sup\{\nu\}\frac{1}{V_\nu|H_\nu|n}\bar{\hat{p}}(\mathbf{x}) \tag{3.66}$$

The conditions under which both the expectation as well as the variance remain unchanged are:

- the samples $\mathbf{x}_s$ are independent of $\mathbf{x}_i$ (I)
- $\hat{p}_S(\mathbf{x}) = p_S(\mathbf{x})$ (II)

In order to see when the second condition holds we have studied the Mean Integrated Square Error (MISE), which is a common measure to compare the

true distribution with the estimated distribution [118]. In Appendix E we find that the MISE is minimized if:

$$\begin{aligned} \bar{p}_S(\mathbf{x}) &\approx p_S(\mathbf{x}) \\ MISE &\leq \sup\{\hat{p}_S(\mathbf{x})\} - \frac{1}{V} \end{aligned} \tag{3.67}$$

where V is the total volume over which the MISE is calculated. Note that for samples on an equidistant grid where condition (3.48) holds, condition I holds:

$$\bar{p}_S(\mathbf{x}) = p_S(\mathbf{x}) \tag{3.68}$$

due to the uniformity of p_S filtering has no effect. Also condition II holds because:

$$MISE \leq \sup\{\hat{p}_S(\mathbf{x})\} - \frac{1}{V} \leq \frac{1}{mV_\mu|H_\mu|} - \frac{1}{mV_\mu|H_\mu|} = 0 \tag{3.69}$$

Thus the non-uniform analysis is more general since it also describes the results of the previous section. A general way to minimize the MISE is to let $p_S(\mathbf{x})$ be a smooth distribution w.r.t. μ such that $\bar{p}_S(\mathbf{x}) \approx p_S(\mathbf{x})$ and by *non-equidistant* sampling such that the volume V is decreased as much as possible. However, note that the volume V is the space where p_S exists, since p_S should exist at least everywhere where the pdf p exists, V should at least *cover* all the data. Note that even if some data are not covered in V, they can still contribute to the estimator. Due to the kernel function χ, all data that is not covered in V is completely re-distributed towards it's nearest neighbor sample. The total volume V can be decreased in at least two ways:

- choosing the samples *locally* closer to another,
- reducing the number of sample-volumes.

In general a sampling more packed than equidistant sampling where (3.48) holds, leads to larger values of $\sup\{\hat{p}_S(\mathbf{x})\}$ but also larger values of $\frac{1}{V}$. Thus as long as $\frac{1}{V}$ increases at a rate higher than or equal to $\sup\{\hat{p}_S(\mathbf{x})\}$ we may expect that the MISE remains small. In order to have comparable rates of increment a triangular-shaped kernel should be chosen, such as a Gaussian. An example of equidistant sampling and two examples of non-equidistant sampling using a (local) sample-density equal to the Nyquist-density of the kernel is given in Figure 3.7. For all three types of sampling it holds that the MISE equals zero because of:

$$\begin{aligned} \sup\{\hat{p}_S(\mathbf{x})\} &= \sup\{\frac{1}{mV_\mu|H_\mu|}\sum_{s=1}^{m}\mu(\frac{\mathbf{x}_s - \mathbf{v}}{H_\mu})\} \\ &= \frac{1}{mV_\mu|H_\mu|} \\ &= \frac{1}{V} \\ \Rightarrow MISE &= 0 \end{aligned} \tag{3.70}$$

where in the left image $m = 8$ and in the right image a non-equidistant sampling is given for which holds the same with $m = 2$. The center image consists of locally more densely packed volumes, where m is roughly equal to 4 if triangular-shaped kernels would be used. Note that the coverage of the data is maintained in all three cases. Clearly, we can reduce the number of samples significantly if the data is sparse, especially if *clusters* are present in the data. If the data is dense without clear clusters, then the reduction through non-equidistant sampling may be expected to be small. Concluding, in order to minimize the number of samples while remaining the condition that $\hat{p}_S(\mathbf{x}) \approx p_S(\mathbf{x})$ over the entire volume where the distribution $p(\mathbf{x})$ exists (from now on called the outcome space[2]. This comes to down to finding the smallest set (partitioning) of sample-volumes that completely covers the outcome space with a minimum of overlap, *optimal partitioning* as in the right image of Figure 3.7.

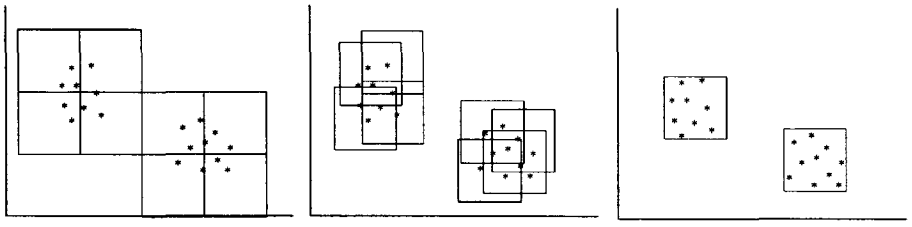


Figure 3.7: *Three types of sampling, using bins of volume* $\frac{1}{V_\mu |H_\mu|}$. *The data is denoted by "*", the samples can be thought of as lying in the centers of the bins.*

A way to achieve an *approximately* optimal partitioning is by introducing *uniform-mean* sampling. Consider samples $\mathbf{x}_s$ distributed according to

$$\mathbf{x}_s = S(\mathbf{n}_s + \mathbf{y}_s) = S\mathbf{n}_s + S\mathbf{y}_s \tag{3.71}$$

where $\mathbf{y}_s$ is a i.i.d. random variable for which holds that:

$$|\mathbf{y}_s| \leq \frac{\mathbf{1}}{\mathbf{2}} \tag{3.72}$$

Suppose there is a number of k data $\mathbf{x}_i$ in each sample-bin of width Δ_S with center $S\mathbf{n}_s$ then we may choose $S\mathbf{y}_s$ according to

$$S\mathbf{y}_s = \frac{1}{k} \sum_{\mathbf{x}_i \in S\mathbf{n}_s}^{k} \mathbf{x}_i - S\mathbf{n}_s \tag{3.73}$$

$$\Rightarrow \quad S\mathbf{y}_s + S\mathbf{n}_s = \frac{1}{k} \sum_{\mathbf{x}_i \in S\mathbf{n}_s}^{k} \mathbf{x}_i$$

$$\Rightarrow \quad \mathbf{x}_s = \frac{1}{k} \sum_{\mathbf{x}_i \in S\mathbf{n}_s}^{k} \mathbf{x}_i \tag{3.74}$$

[2]Often the outcome space is the same as the feature space, but essentially the outcome space is only a sub-space of the feature space: that part where the outcomes exist

if k is zero then the sample is discarded. In other words, we replace the initial $\mathbf{x}_s$ by the mean of the data captured in the sample-bin if the bin is non-empty else we discard the sample. We will refer to this type of sampling as: *uniform-mean* sampling. Since $\mathbf{y}_s$ is a function of the data $\mathbf{x}_i$, where $\mathbf{x}_i$ are i.i.d. random variables, then also $\mathbf{y}_s$ is i.i.d. and thus we may assume that the samples are also independent of the data-distribution such that the condition (I) is satisfied.

In order to look for the smallest number of sample-volumes covering the data we adopt the following scheme using a *reduction parameter* r:

- step 0. Start with an equidistant grid with sampling matrix rS_0, where $r = 1$ and S_0 equals the Nyquist-density of the kernel μ with which $p_S(\mathbf{x}_s)$ is to be estimated,
- step 1. replace the samples $\mathbf{x}_s$ according to (3.74) with $S = rS_0$,
- step 2. gradually increase the *reduction parameter* r as long as every $\mathbf{x}_i$ is contained within one of the updated (hence repositioned) sample-volumes $|S|$.

Instead of checking whether each data-point is contained within a sample-volume we may also monitor an error function of the DK estimator vs. the Parzen Windows estimator or, in classification problems, directly monitor the classification error. The reason for starting with an equidistant grid is a heuristic in order to reduce the overlap of the individual volumes. If, due to the particular distribution and the offset of the grid, we get a partition like the center-image in Figure 3.7 then the MISE is not influenced *if* we use triangular-shaped kernels. Therefore the expectation and variance of the estimator remain constant throughout the sampling process until coverage of the data is lost. In practice some *coverage loss*[3] can be allowed leading to even more reduction, but then a classification-error function should be monitored to see how much loss is acceptable. In some cases all data are important and coverage loss will induce errors. In other cases the data may suffer from noise and coverage loss may reduce classification errors.

We observe that uniform-mean sampling can be thought of as a form of K-means clustering [4], where each sample-volume defines a seed and limits the "walk" of the center to it's volume due to the single update. Due to this single update the uniform-mean scheme is much less expensive than K-means clustering, which needs at least several updates. For a recent treatise on clustering we refer to [8]. The uniform-mean approach of clustering may not lead to the absolute optimal partitioning, more intelligent approaches may be found. The main reason for using the uniform-mean approach is that it is simple (computationally not expensive) and guarantees that the conditions for optimal partitioning hold as long as no coverage loss occurs.

Estimators that use (computationally more expensive) hierarchical or partitional clustering for the kernel-positions are not new, several cluster-based

[3]With coverage is meant the coverage of the data, instances, by the sample-volume of the kernel which equals its duration (spatial width). So for Gaussian kernels it can happen that an instance is not covered but still contributes to the kernel-count.

binned-kernel estimators have been proposed , see also [140]. In our experiments we will compare our method with the weighted Parzen Windows approach reported in [6]. The reported analysis for these methods states that only in the case of taking as many clusters as data, the Parzen Windows estimate is obtained, all other cases may lead to inferior results. We can understand this, since clustering in general does not guarantee that the optimal partitioning, as derived above, is found, nor that the conditions for optimal partitioning are satisfied.

3.4 Experiments

Experiments have been carried out on synthetic and real data to demonstrate the use of DK estimators. In all the experiments Gaussian kernels were used for both the Parzen Windows estimators as well as the DK estimators. The covariance-matrix H of the Gaussian kernels for ϕ in the PW estimator and ν in the DK estimator was chosen to be a $H_\nu = (h\Sigma)^{0.5}$. Here, Σ is the (multidimensional, class-conditional) empirical covariance-matrix of the data and h is a smoothing parameter. According to (3.50) for equidistant sampling the sampling matrix was chosen as $S = (h\Sigma)^{0.5}$. Finally, the covariance-matrix for the kernel μ in the DK estimator was always chosen according to (3.51): $H_\mu = (\frac{h\Sigma}{2\pi})^{0.5}$

3.4.1 Uniform vs. Non-uniform DK estimator

For visual comparisons we experimented on one-dimensional synthetic data to show the difference between uniform and non-uniform DK estimator. For the non-uniform DK estimator we used a reduction-parameter $r = 1$. The synthetic data set consists of n random draws from a standard Gaussian with mean zero and variance one. The density was estimated for several smoothing values of h and several draws n. As a reference for numerical comparison the true mean-absolute error (mae) between the Parzen Windows estimate and the true density was calculated over approximately 200 values of $p(x)$, see Figure 3.8. The mae is calculated as the root from the mean-square error. Further, the mae between the PW estimate for the uniform and non-uniform DK estimator, denoted by ϵ_u and ϵ_n respectively, were also calculated. As can be seen from these three figures, the mae between the PW estimate and the true distribution is two orders of magnitude larger than the errors between the DK estimates and the PW estimate. Hence, we see that the DK estimates are equally accurate in estimating the true density as the PW estimate. A visual example is given in Figure 3.9.

We also calculated the number of kernels that were used by both the uniform and non-uniform DK estimators and expressed them as a reduction-ratio $\frac{m}{n}$, where m is the number of kernels and n the number of data, see Figure 3.10. Clearly, the smoother the kernel the smaller the number of kernels used, which is a direct implication of sampling with the Nyquist-density. As can be seen from

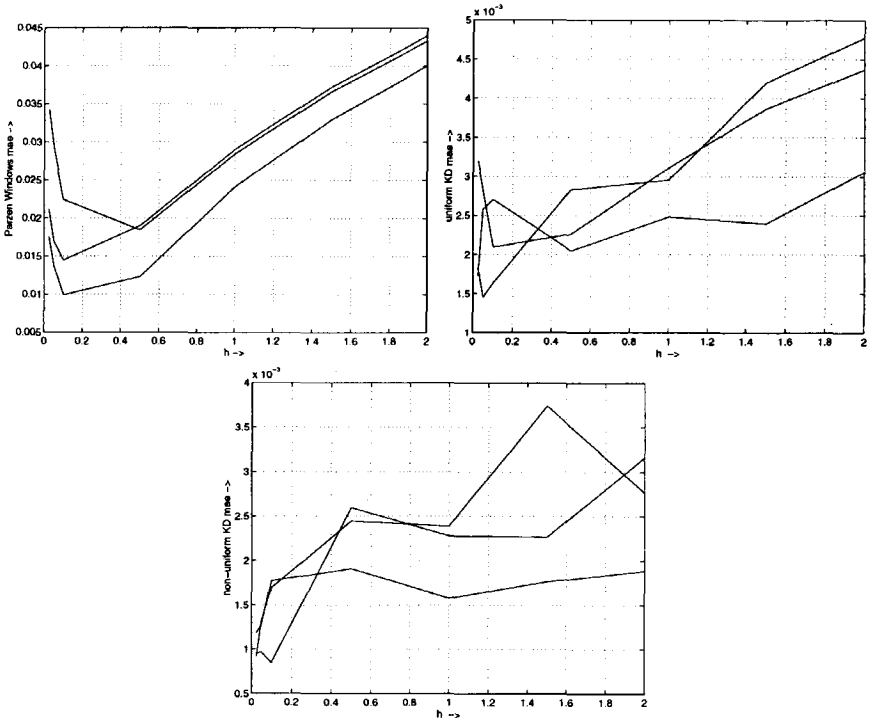


Figure 3.8: *Mean-absolute error of the Parzen Windows estimator (top left), the uniform DK estimator (top right) and the non-uniform DK estimator (bottom) with the true density as a function of the smoothing-parameter h. Curves corresponding to draws: 25,50,100 (in descending order of the mae).*

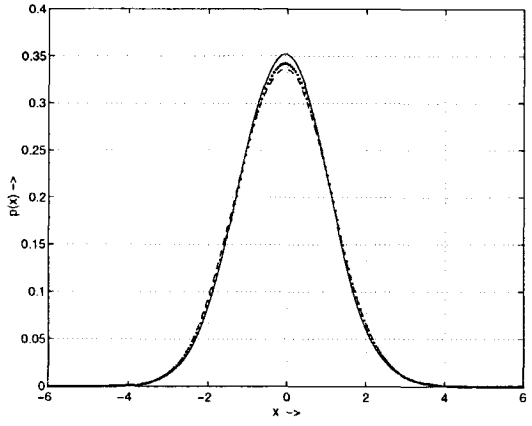


Figure 3.9: *Three estimates of a normal distribution, the solid line is the Parzen Windows estimate, the dotted line is the uniform DK estimate, the dashed line is the non-uniform DK estimate. The estimates are based on 50 draws, uniform DK needed 9 kernels and non-uniform needed 8 kernels.*

comparisons between the figures, the non-uniform DK estimate, even without an optimal partition due to the fixed $r = 1$, needs less kernels than the uniform DK estimate. Also note that for more dense data (e.g. 100 draws and more), the reduction ratio becomes very small also for the uniform DK estimator.

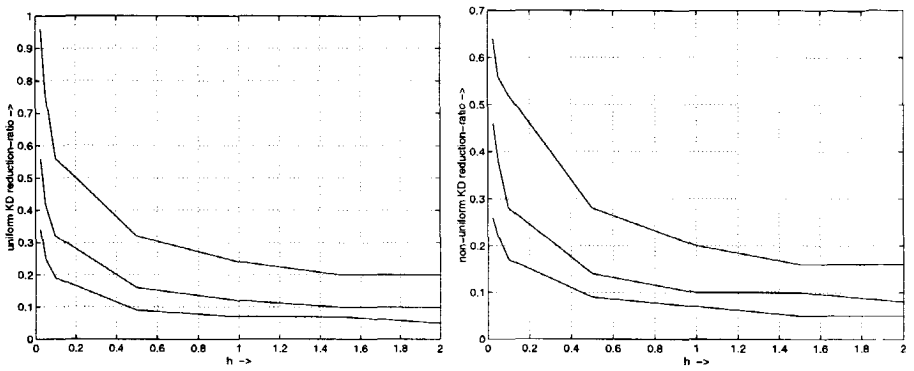


Figure 3.10: *The reduction ratio's for the uniform DK estimate (left) and the non-uniform DK estimate (right) as a function of the smoothing parameter h. Curves corresponding to (from up to down): 25,50,100 draws.*

3.4.2 r-Optimal Non-uniform DK Estimator

We repeated the above experiment for the non-uniform DK estimator, where the reduction parameter r was optimized. The optimization was done by using the *non-equidistant* sampling scheme, in which r is increased as long as all samples are covered. Results of the mae and the reduction-ratio are given in Figure 3.11. When compared to the results in Figure 3.8 it is clear that further reduction is indeed possible, while still keeping the mae an order of magnitude smaller than the true mae. A visual example is given in Figure 3.12.

Finally we allowed a 10% loss of coverage of the data and optimized r again as long as 90% of the data was covered. Results of the mae and the reduction-ratio are given in Figure 3.13. A visual example is given in Figure 3.12. Also from these figures it is clear that the mae is still an order of magnitude smaller than the true mae. However further reduction may lead to mae in the same order of the true mae.

3.4.3 Classification on Sparse Data Sets

To demonstrate the use of the non-uniform DK estimator in high-dimensional classification problems on sparse data we used two popular data sets: IRIS and IMOX. IRIS consist of 150 patterns from 3 classes in 4 dimensions [46]. IMOX consist of 192 patterns [61] from 4 classes in 8 dimensions. For classifier design,

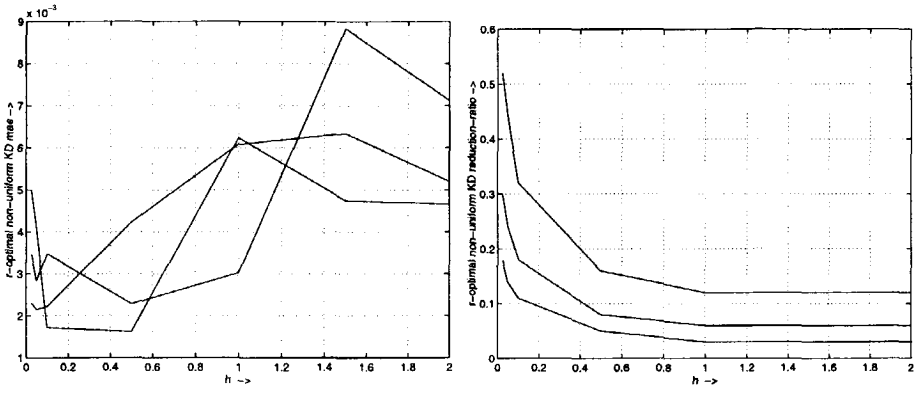


Figure 3.11: *Mean-absolute error (left) and reduction ratio's (right) of the non-uniform DK estimator as a function of the smoothing-parameter h with optimal reduction parameter r. Curves corresponding to draws: 25,50,100.*

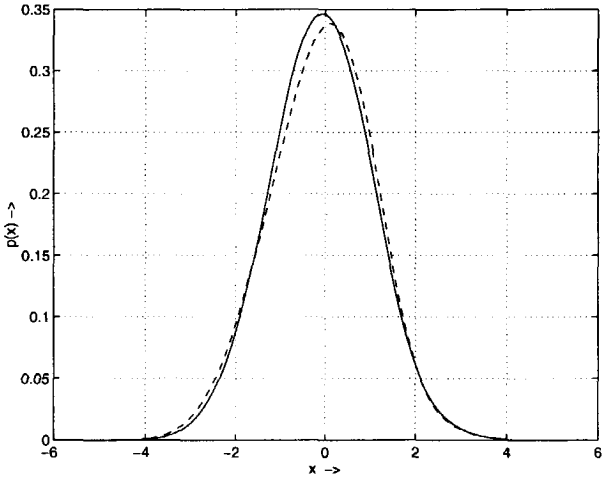


Figure 3.12: *Two estimates of a normal distribution, the solid line is the Parzen Windows estimate, the dashed line is the non-uniform DK estimate with optimal* $r = 2$. *Estimates are based on 50 draws, the non-uniform DK estimate needed 5 kernels.*

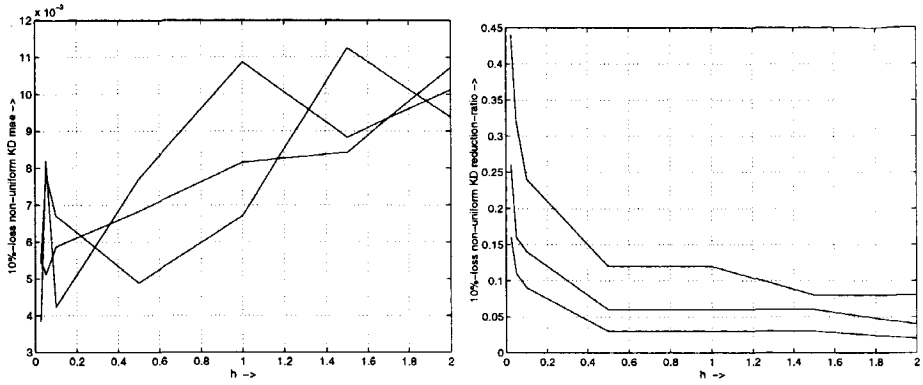


Figure 3.13: *Mean-absolute error (left) and reduction ratio's (right) of the non-uniform DK estimator as a function of the smoothing-parameter h with optimal reduction parameter r. Curves corresponding to draws: 25,50,100.*

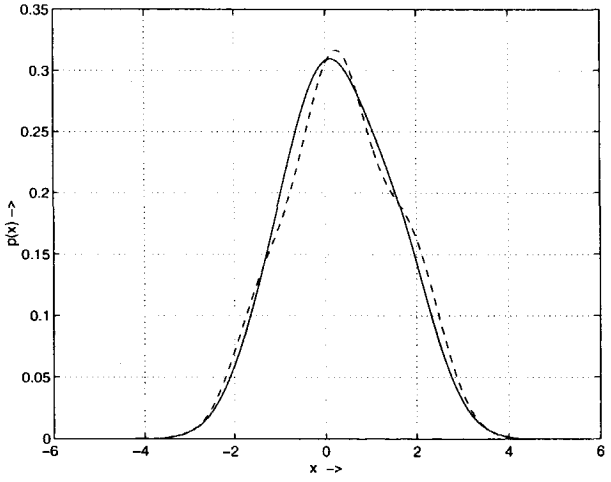


Figure 3.14: *Two estimates of a normal distribution, the solid line is the Parzen Windows estimate, the dashed line is the non-uniform DK estimate in which up to 10% of the data was allowed to be uncovered. Estimates are based on 50 draws, the non-uniform DK estimate needed 3 kernels.*

several strategies exist for several decades, the most popular being the leave-one-out method and the bootstrap method [52, 61, 77]. In our experiments we used the leave-one out method to compare our results with the weighted PW approach [6], which is also a method which can be thought of as an estimation using a non-uniform distribution. In itself the weighted PW approach bears large resemblance with the mixture approach of West [140]. Both approaches are known for their large reductions.

First, the smoothing parameter h of the Parzen Windows estimate was optimized by using the leave-one-out method. From literature on the two data sets it is well-known that the smoothing parameters $h^{0.5}$ for the PW classifier, in case of minimum leave-one-out error, are near 1.4 for the IRIS data set and near 1.1 for the IMOX data set with relative errors of 0.0133 and 0.0521 respectively [6, 61]. Second, having obtained the optimal PW smoothing, the number of kernels were reduced by using the *uniform-mean sampling* scheme and the reduction ratio and the leave-one-out (loo) error were calculated for the non-uniform DK estimator. A calculation of the leave-one-out error on a data set having n patterns, means designing the classifier n times. Therefore, for each design during the leave-one-out procedure, the uniform-mean sampling was performed with a constant coverage-loss threshold and the resulting reduction-ratio was calculated as an average reduction ratio over all n designs. This experiment was repeated for several values of the coverage-loss threshold and the results are summarized in Table 3.1. Note that the coverage-loss threshold is a maximum for the allowed coverage loss in the uniform-mean sampling scheme, the *actual* coverage loss is always smaller then or equal to the threshold. Even if we require that at least 1% should be covered in the partitioning, then at least one kernel will have to be placed in the feature space which will cover *at least* one pattern. Depending on the sparseness of the data-set, this may still lead to "reasonable" *actual* coverage.

As we expected from the analysis, at zero coverage loss the loo error is equal to the loo error of the PW classifier. Even up to 10% coverage loss the loo error remains the same. From the experiments in the previous section this is not surprising since at 10% loss the non-uniform DK estimator still resembled the PW estimator fairly well. However, for higher loss percentages an interesting phenomenon occurred in the IMOX data set. First of all it is striking that the loo error remains constant in a large range of loss-percentages. Second, at nearly ridiculously low reduction-ratios (meaning that only a very few kernels are used) the error *decreases* and becomes smaller than the optimal PW loo error. This is not unique for the IMOX data set since experiments on other (high-dimensional) data sets (both synthetic and real) displayed the same phenomenon.

We do not have a good explanation for this phenomenon. The problem is that the true error rate is very difficult to estimate, and the loo error is only an approximation. It may be that the PW estimator suffers sometimes from to much noise in the data set and since each pattern carries a kernel, PW might over-fit the data; over-fit meaning that the data is described rather than the underlying distribution. In literature there is some discussion on the effectiveness of the PW estimator in higher dimensions [140, 125] which seem

Table 3.1: *Results of classification with the double-kernel estimator on the IRIS and IMOX data sets. Columns show (left to right): data set, smoothing parameter h, coverage-loss, reduction ratio and leave-one-out error.*

data set	h	maximum loss%	Reduction Ratio	leave-one-out error
IRIS	1.4	0	0.3058	0.0133
		5	0.2573	0.0133
		10	0.1961	0.0133
		20	0.1485	0.0333
		30	0.1005	0.0133
		40	0.0518	0.02
		50	0.0201	0.02
		75	0.0201	0.02
		99	0.0201	0.02
IMOX	1.1	0	0.8616	0.0521
		5	0.8237	0.0521
		10	0.6965	0.0521
		20	0.6394	0.0521
		30	0.5343	0.0521
		40	0.4568	0.0521
		50	0.3807	0.0469
		75	0.1719	0.0365
		99	0.0209	0.0521

to support, hence the phenomenon may only be restricted to higher dimensions where the PW estimator is not optimally effective. On the other hand, we may also reason that the DK estimator over-fits the data! Although the non-uniform DK estimator uses only a fraction of the kernels, it is a more flexible density estimator and can be optimized over a second parameter: the reduction ratio. As known in pattern recognition, too much optimization may lead to over-fitting the data as well.

When compared to the weighted PW approach using these data sets and the leave-one-out error for analysis of the reduction ratio then [6] reports an reduction ratio of 0.4 with a loo error of 0.0133 for the IRIS data sets and a reduction ratio of 0.562 with a loo error of 0.0521 for the IMOX data set. Clearly, the non-uniform DK estimator outperforms the reported results by far.

3.5 Conclusion

We discussed a general nonparametric density estimation technique from the field of statistical pattern-recognition based on Parzen Windows. Instead of purely data-driven kernel placement, as in the Parzen Windows technique, a

sample-driven distribution of kernels is used to reduce the number of kernels for estimating the probability density function. The heart of the technique is formed by sampling and reconstruction of continuous functions. The kernel distribution can be obtained from an a priori equidistant sampling grid as well as from a data-driven non-equidistant sampling grid, leading to the uniform and non-uniform double-kernel estimator respectively. The non-uniform double-kernel estimator can be optimized according to two parameters, a smoothing parameter and a reduction parameter. The smoothing parameter determines the width of the kernels and the sampling rate, whereas the reduction parameter essentially determines the sub-sampling rate. It has been proven that the non-uniform KD estimator converges in mean square as long as no coverage loss of the instance space occurs. Experiments on both synthetic data as well as real data show that large reductions in the number of kernels can be obtained even in multi-dimensional spaces, while obtaining equal or even better error rates than with the Parzen Windows technique. Experiments also show that often some coverage loss can be accepted without influencing the error rate. Comparisons with the weighted Parzen Windows technique show that the non-uniform double-kernel estimator can lead to larger reductions and smaller error rates. The main drawback of the double kernel estimator is the time necessary for training due to the reduction parameter. Optimization techniques, like hill-climbing or genetic-algorithms may be more appropriate than the straightforward sequential approach deployed here. However, the storage and time necessary for classification is severely reduced.

The double-kernel technique is interesting because it generalizes several binned-kernel approaches and leads to excellent results. However, we will use it in the remainder of this thesis mainly to ease the conception of the fuzzy probabilistic framework introduced in the next chapters.

Chapter 4

Fuzzy Probability

In the previous chapter we discussed density estimation based on kernel distributions. We presented the double-kernel estimator which was obtained from a mathematical variation on the theme of Parzen Windows. We also demonstrated that this estimator is more efficient for decision-making problems (i.e. classification) than Parzen Windows due to its equal or smaller error rate in classification problems while using a smaller number of kernels. To understand this increased efficiency we will make a conceptual transition. Instead of regarding the distribution of kernels as a mathematical variation, we will regard them as a lattice of *fuzzy sets*, see Figure 3.6. We may now reason that the estimator uses fuzzy sets to estimate a probability density function. We can therefore reasonably assume that the efficiency of the double-kernel estimator in classification tasks arises from its use of both types of uncertainty.

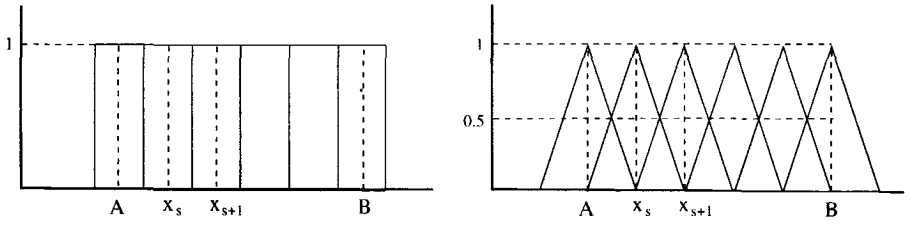


Figure 4.1: A kernel distribution as a lattice of sets on the interval $[A, B]$. A lattice of crisp sets on the left and a lattice of fuzzy sets on the right. A crisp set is regarded as a special kind of fuzzy set.

In this chapter we will first motivate and develop a framework in which fuzziness and probability co-exist. We will show that the usual probabilistic framework and the usual fuzzy logic framework are combined in a single fuzzy probabilistic framework. Further, we will show that the double-kernel estimator can be derived from this framework. Finally we will use the framework to obtain

a rather simple but efficient scheme for decision making - classification - and illustrate it with an example.

4.1 Knowledge and Uncertainty

In the Cartesian view, knowledge can be regarded as a set of statements which are either true or false. Such statements may for example be:

- a characterization,
- a theorem,
- a rule,
- a physical law,
- a mathematical relation.

Often science is regarded as the quest for this knowledge, for finding out what is true and what is not. We adopt a more subtle view on knowledge, one which regards knowledge as statements which are true or false with some degree of certainty. Hence, the concept of certainty, or uncertainty, is identified as an integral part of knowledge. The first issue that we will briefly discuss is where such uncertainty may come from, and second how it can be connected to the truth of a statement.

4.1.1 On Knowledge

We hold the view that we live in (at least) three worlds: the physical world, the observed world and the mental world (see also Figure 4.2). This view is related to the three worlds of Karl Popper [111].

We assume, i.e. belief, that the physical world consists of objects and relations, which are certain and exact. Unfortunately we do not know this physical world other than through our senses and our instruments: the observed world. The discrepancy between the physical world and the observed world comes from the erroneous measurements of our senses and instruments. We only know the physical world to the extent our senses and instruments allow. To make matters worse, we know from Quantum Mechanics that there is a fundamental limitation to the precision with which we observe the physical world due to the interaction of the observer with the physical world. The discrepancy between the physical world and the observed world is maybe best expressed in the words of Werner Heisenberg:

"What we observe is not nature itself, but nature exposed to our method of questioning."

The mental world consists of knowledge about the observations of the physical world. The philosopher Immanuel Kant was one of the first to notice that

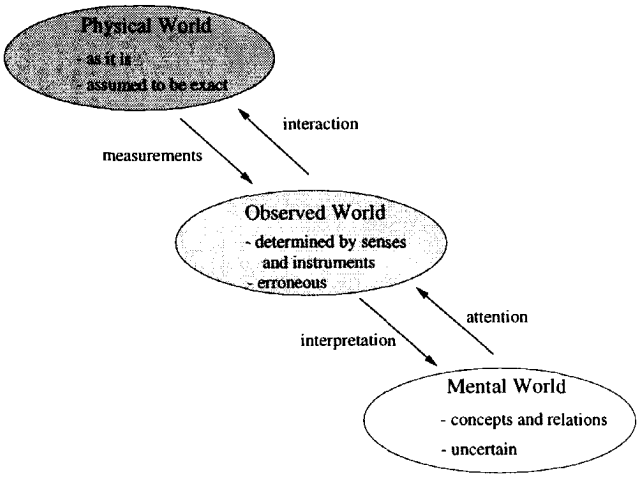


Figure 4.2: Three worlds and their relations.

our mind forms a pair of glasses, a reference frame, *through* which we observe the world and by which we interpret the observations. It should be noted that our observations are closely interwoven with our knowledge (theories), and therefore the difference between observed world and mental world may seem artificial. However, one motivation for making this difference is that often an observation can be interpreted in several different (sometimes even contradictory) concepts or associations, which still all give a valid explanation for the observation.

As a model for knowledge, we assume that knowledge consists of *concepts* and *associations*. Here a concept is an object of the mind, and associations are (supposed) relations between those concepts. Processes like learning, reasoning etc., can be thought of as making new associations, forming new concepts, exploring the relations between the concepts and so on. Our concepts are often not sufficient to describe an observed object exactly, quite often we have to refine our concepts to be consistent with the observations. This may be partly due to our imperfect observations of the physical objects or it may be due to the limited number of concepts we have to describe it. Further, our associations may not be sufficiently accurate or sufficient in number to describe all the observed relations that may exist between the observed objects. Hence, there is a discrepancy, a mismatch, between the observed world and the mental world due to limitations to our knowledge which blur our pair of glasses. To deal with this discrepancy we must become aware of it and accept some uncertainty in our knowledge; our concepts may be not so precise as we would like them to be, and our associations may be correct only most of the time.

It may seem to some that we give up the quest for knowledge, or reduce it to "uncertain" statements. It should therefore be remarked that the drive behind science should be to decrease the uncertainty as much as possible. Uncertainty in our knowledge remains an unwanted guest, but one which cannot be denied existence.

4.1.2 On Certainty

We regard (un)certainty as a property expressing the extent to which a statement[1] is (not) in accordance with our experiences or observations. This implies that we can always refine our statements such that the uncertainty is removed. However, this may ultimately lead to infinite statements, each describing a single observation perfectly. If that would be our knowledge, then we would not be able to make some sense of the world we observe. We would, for example, not be able to make predictions since hardly ever does the *exact* same thing occur twice. Hence, to see some regularity in our observations it seems necessary to allow some uncertainty in our statements.

We may not be certain whether a statement is true (certainly in accordance) or false (certainly not in accordance). Therefore we allow that, due to some underlying *type* of uncertainty, the "truth" maybe graded. That is, a statement may be partly true but also partly false. In the case that we are completely certain that a statement (linguistic or mathematical) is true, the truth value is one, in the case that we are completely certain that a statement is false, the truth value is zero. Many gradations of truth are possible, but always with respect to the certainty; as a way of expressing we could say that certainty modulates the truth, where the certainty takes values between zero and one. We assume that false and true are complementary, and we require that the sum of their truth values always equals one.

On the basis of (un)certainty, we can make a choice, a decision. Usually we make that decision of which we are most certain. Essentially, decision making means reasoning with uncertainty in order to arrive at a choice. However, as soon as the decision is made, the certainty is not relevant anymore since a choice is made by which other possible choices are ruled out. Therefore, we should be very careful with making a decision during reasoning. As an analogy, consider the calculation of a division of two numbers, truncating the numbers before division introduces unnecessary errors in the final result. We think that during reasoning we should take all uncertainty into account, explore the possibilities, and postpone the decision until we are either sufficiently certain or cannot reason any further.

4.2 Types of Uncertainty

In this section we will examine two well-known types of uncertainty; probability and fuzziness.

4.2.1 Probability

The English philosopher John Locke said that : "probability is likeliness to be true" and that "Probability,...,being to supply the defect of our knowledge,...,on propositions we have no certainty..." [82]. Indeed, he connects certainty to the

[1] We think of a statement as an association between concepts.

concept of probability, probability expresses the uncertainty of (our knowledge of) truth about a statement. In this respect it is sometimes noted that in probability theory events are still crisp even before a decision; a statement is either true or false (0 or 1) but we are uncertain whether it is true or false. To clarify the type of uncertainty that probability expresses, we consider the statement: "John goes to church on Sundays". We may question whether this statement is *always* true. If John is tremendously faithful and can not be ill (John is a robot), then we may say that the statement is (always) certainly true. However if John goes to church nearly every Sunday, then the statement is nearly always true. These examples can be extended to "sometimes", "often", "now and then" etc., these words are all related to the concept of occurrence. Here we avoid nomenclatures as "somewhat true", "a little true", etc., since we think that they are not related to uncertainty arising from occurrence.

Probability, or randomness, is usually measured by simply observing John an by determining the frequency with which John goes to church. The certain ("sure") event has a probability normalized to one. The fact that probabilities are measurable and the axiom of identical outcomes given sufficient counts (reproducibility), has led to the idea that probabilities are *objective*. However, even Locke admitted that probability is in at least two ways subjective. First, Locke regards probability as a degree of belief which can be attached to some statement, without experiments, on the basis of the credibility of the speaker claiming the statement. Second, Locke also points out that statements have to be well defined before an objective truth can be attached to it. This has led many to believe that the only statements that are undoubtedly (certainly) true are syllogisms like: "if John goes to church, John goes to church". Indeed, in practice it is often very difficult to define our statements or our events of interest exactly, see also [45]. In our case, suppose John goes to a cathedral, could this be counted as going to church? Did we take cathedral into account in our definition of church? If not, we should add "cathedral" to the class of possible events and hence, going to the cathedral is not counted as going to church. However, where to draw the line? Can the "Main Street Church" also be counted as a church, even though it is not exactly the same as the "St. John's Church", or should we extend the class of possible events to "Main Street Church". We reason that at some point (sooner or later) an a priori subjective judgment will be made whether an outcome belongs to an a priori defined event or not. Unfortunately, probability theory can not take the uncertainty related to this kind of judgment into account. Hence, within probability theory we make an a priori decision even before we reason. This is certainly a defect of probability theory, since we argued that decisions should be postponed until we are either sufficiently certain or cannot reason any further.

Axiomatic probability

Many mathematical frameworks have been developed for probability, and one of the most popular frameworks is the axiomatic approach [45]. Here we will briefly recall the basics of the axiomatic approach. Given a one-dimensional real

feature space $\mathcal{R}$ (known as the outcome space or sample space) with elements x. Further, let us define the σ-field $\mathcal{A}$ as the class of events, where an event is a set of outcomes. Then the probability space is defined as the triplet $(\mathcal{R}, \mathcal{A}, P)$, where P is the *probability* measure over $\mathcal{R}$ for which holds that:

$$P(A) \geq 0 \quad \forall \quad A \in \mathcal{A} \tag{4.1}$$

$$P(\mathcal{A}) = 1 \tag{4.2}$$

$$P(A \cup B) = P(A) + P(B) \quad iff \quad A \cap B = \emptyset \tag{4.3}$$

$$(\forall i A_{i+1} \subset A_i) \ \wedge \ (\cap_i A_i = \emptyset) \Rightarrow \lim_{i \to \infty} P(A_i) = 0 \tag{4.4}$$

Note that $\mathcal{A}$ is a σ-field implies that the operations of union, intersection and complement on sets which are a member of $\mathcal{A}$ results in a set which is again a member of $\mathcal{A}$.

Relative frequency

In technical sciences the concept of probability is often narrowed by combining the axiomatic approach with a relative frequency approach for measuring probabilities [45]. Given an (event) $A \in \mathcal{A}$ and a random variable (r.v.) ξ, the probability that ξ is an element of event A (shortly put as: the probability of event A) can be expressed by the Lebesgue-Stieltjes integral:

$$P(\xi \in A) = P(A) = \int_{x \in A} dF(x) \tag{4.5}$$

where F is the distribution function of r.v. ξ, defined as:

$$F(x) = P(\xi \leq x) \tag{4.6}$$

having the obvious properties:

$$\begin{aligned} F(\infty) &= 1 \\ F(-\infty) &= 0 \end{aligned} \tag{4.7}$$

further, if there exists a density function $p(x)$ for r.v. ξ, such that

$$\frac{dF}{dx} = p(x) \Rightarrow F(x) = \int_{-\infty}^{x} dF = \int_{-\infty}^{x} p(y)dy \tag{4.8}$$

then we can write for the probability of $P(A)$

$$P(A) = \int_{x \in A} dF(x) = \int_{x \in A} p(x)dx \tag{4.9}$$

If we use $f_A(x)$ as being the set function of A, defined in the classical sense of sets: $f_A : x \to \{0, 1\}$, then the usual definition of the probability of an event A can be expressed as:

$$P(A) = E[f_A] = \int f_A(x)p(x)dx \tag{4.10}$$

For an extensive overview of classical probability theory and its properties see [83]. Some excellent essays have been written on probability by the 19th century American Philosopher Charles Sanders Peirce which we recommend for further reading [109]. Peirce elaborated on the basis of Locke's ideas and founded the philosophical base of what now can be regarded as probabilistic reasoning (Peirce called it: "Probable Inference").

4.2.2 Fuzziness

Maybe the first traces of the concept of fuzziness can be found in the philosophy of Plato. According to Plato we live in, what he called, the sensory world. Apart from this sensory world, there is also the world of ideas, a divine world in which we stay before birth and after dead. In the sensory world only shadows of the ideas are found. The "fuzziness" of Plato lies herein, that we can recognize an object in our sensory world because it is an (imperfect) image of the idea of the object in the divine world of ideas. A beautiful illustration of Plato's worlds is expressed in his book "symposium" ($\sigma\upsilon\mu\pi\sigma\sigma\iota\sigma\nu$). In this book, Diotima explains to Socrates the secret of love, how it can guide the beholder in his ascent to the world of ideas. First, he should recognize the beauty of one body, then he should recognize the beauty of two bodies. Along this line he should proceed to the beauty of all bodies, and from there on to the beauty of activities, and then to the beauty of knowledge, untill he finally arrives at beauty itself: the idea of beauty. Once arrived there, he will not want to return to the shadows of beauty, which are present in the sensory world.

Plato's concept of the sensory world may resemble our observed world, but for the world of ideas we do not have a good analogy, since we only deal with the worlds that we live in. However, the world of ideas is maybe most related to our concept of the mental world. Suppose we would adopt the Platonic view in our mental world, in the sense that we would accept the concept of a table exists; the *ideal* table or the *prototype* of a table[2]. Then the statement: "The observed item is a table" can be considered true if the observed item *resembles* the prototypical table. Hence, certainty of the truth may be associated with *similarity*. In this sense, the statement about the table can be said to be certainly true if the item looks exactly like the prototypical table. For measurable features like height, length, volume, etc. similarity functions can be constructed, and its counterpart, *dissimilarity*, provides a practical measure of uncertainty. Typical examples of these dissimilarity functions are error functions like mean-square error or distance functions in general. In cases where the distance of the item to the prototype cannot be calculated, subjective distance functions may be used. Note that also a criterion like the mean-square error on measurable features is in a sense subjective; often one can use many other distance measures as a criterion. Hence, a fuzzy set can be interpreted as a subjective similarity function. It expresses the certainty on the basis of similarity, where perfect similarity (to the prototype) is normalized to one. This is most likely the widest

[2]We do not rule out that we sometimes need a set of prototypes to represent the "ideal"

spread interpretation of fuzzy sets. Numerous schemes of inference using fuzzy sets, or Fuzzy Logic, exist, and a larger number of them are founded by Zadeh [146] [148].

Fuzzy sets

We will recall very briefly the mathematical framework for fuzzy sets. For a more extensive overview of fuzzy sets and fuzzy logic we refer to [70]. The similarity to a set (concept, event) A on the basis of some feature X is given by a membership function $\mu_A(x)$ with $x \in X$ such that:

$$0 \leq \mu_A(x) \leq 1 \quad \forall x \in X \tag{4.11}$$

Hence, the membership function is a mapping of the kind $\mu_A : X \rightarrow [0,1]$. Recall that (normal) crisp-set functions $f_A(x)$ are mappings of the kind $f_A : X \rightarrow \{0,1\}$, that is, f only takes the value 0 (not a member) or 1 (is a member). Instead of defining the similarity on the basis of a single feature, we can also define similarity on the basis of several features, say $x_1, x_2, ..., x_d$. The function $\mu_A(\mathbf{x})$ is then a mapping of $\mu_A : X^d \rightarrow [0,1]$ (and $\mathbf{x} \in X^d$). Some basic notions of fuzzy sets can be regarded as extensions of crisp sets. Let A and B be two fuzzy sets, so we have:

- *empty set* : $\mu_\emptyset(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in X^d$
- *universal set* $\neg$: $\mu_{\neg\emptyset}(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in X^d$
- *equality*: $A = B \Leftrightarrow \mu_A(\mathbf{x}) = \mu_B(\mathbf{x}) \quad \forall \mathbf{x} \in X^d$
- *complement*: $\neg A \Leftrightarrow \mu_{\neg A}(\mathbf{x}) = 1 - \mu_A(\mathbf{x}) \quad \forall \mathbf{x} \in X^d$
- *containment*: $A \subset B \Leftrightarrow \mu_A(\mathbf{x}) \leq \mu_B(\mathbf{x}) \quad \forall \mathbf{x} \in X^d$
- *union*: $A \cup B \Leftrightarrow \mu_{A \cup B}(\mathbf{x}) = u(\mu_A(\mathbf{x}), \mu_B(\mathbf{x})) \quad \forall \mathbf{x} \in X^d$
- *intersection*: $A \cap B \Leftrightarrow \mu_{A \cap B} = i(\mu_A(\mathbf{x}), \mu_B(\mathbf{x})) \quad \forall \mathbf{x} \in X^d$
- *product*: $AB \Leftrightarrow \mu_{AB}(\mathbf{x}) = \mu_A(\mathbf{x})\mu_B(\mathbf{x}) \quad \forall \mathbf{x} \in X^d$
- *sum*: $A + B \Leftrightarrow \mu_{A+B}(\mathbf{x}) = \mu_A(\mathbf{x}) + \mu_B(\mathbf{x}) - \mu_A(\mathbf{x})\mu_B(\mathbf{x}) \quad \forall \mathbf{x} \in X^d$

On most of these notions the community of fuzzy logic agrees. However the choice of the i-operator, denoting the intersection of two fuzzy sets and of the u-operator, denoting the union of fuzzy sets, depends much on the type of application. There are two frequently used conditions for choosing of operators: first, the operators should be generalizations of the crisp-set union and the intersection and second, they should satisfy the De Morgan's Laws:

$$\begin{aligned} \neg(A \cup B) &= \neg A \cap \neg B \\ \neg(A \cap B) &= \neg A \cup \neg B \end{aligned} \tag{4.12}$$

These conditions result in the well-known t-norms and t-conorms for the fuzzy union and intersection. However, there are applications in which even the De Morgan's Laws are not necessary, see [55].

Fuzzy logic

Quite often only the projections of a multidimensional membership function are known. Since in general a multidimensional function cannot be uniquely defined by its projections, we have to create the membership function by inference. This inference is based on extending the one-dimensional membership functions on the Cartesian product space $X \times Y$. Suppose we have a fuzzy set A defined on feature X by $\mu_A : X \rightarrow [0,1]$ and we have a set B defined on feature Y by $\mu_B : Y \rightarrow [0,1]$, then the following basic extensions can be done:

- *separable and-extension*:
  $R = A \wedge B \Leftrightarrow \mu_R(x,y) = \mu_A(x)\mu_B(y) \quad \forall (x,y) \in X \times Y$
- *separable or-extension*:
  $R = A \vee B \Leftrightarrow \mu_R(x,y) = \mu_A(x) + \mu_B(y) - \mu_A(x)\mu_B(y) \quad \forall (x,y) \in X \times Y$

The one-dimensional membership functions can always be derived from these extensions by projection:

- *and - projection*: $A_{\perp x} \Leftrightarrow \mu_A(x) = \sup_{y \in Y}\{\mu_{A \wedge B}(x,y)\}$
- *or - projection*: $A_{\perp x} \Leftrightarrow \mu_A(x) = \mu_{A \vee B}(x,0)$

It is said that R is a *fuzzy relation* on the Cartesian product space $R : X \times Y \rightarrow [0,1]$, as a generalization of the classical relation. Since a fuzzy relation is in itself a multidimensional fuzzy set, the fuzzy set operation can be applied to relations as well. These relations can be used to make inferences about B given A. A important relation is the implication, mathematically denoted as:

- *implication*: x **is** $A \Rightarrow y$ **is** B

The truth table for the fuzzy implication is given by $R = A \wedge B$. Suppose we have a an observation A', which can be a fuzzy set which resembles the set A, but it may also be a single measurement, say x_0. In the latter case, x_0 is referred to as a fuzzy singleton.

- *fuzzy singleton*: $\mu_{A'}(x) = 1$ **if** $x = x_0$ **but** 0 **elsewhere**

By using this observation and the above implication we can infer $\mu_{B'}$ from projection of $A' \wedge R$ on the y-axis.

- *inference*: $B' = A' \circ R$
- *set inference*: $\mu_{B'}(y) = \sup_{x \in X}\{\mu_{A'}(x)\mu_R(x,y)\}$
- *singleton inference*: $\mu_{B'}(y) = \mu_A(x_0)\mu_B(y)$

As an illustration consider the following. A can be the event "sounds like a duck" and B is the event "looks like a duck". The implication then is "if it sounds like a duck, then it looks like a duck". Suppose we have the observation A', meaning "the animal sounds approximately like a duck" (obtained by measuring the

number of decibels of a quacking sound of an unknown animal). The certainty that the animal also looks like a duck is then given by $\mu_{B'}$. Note that if x_0 is measured which is perfectly similar to the prototypical duck sound, it is inferred *with certainty* that the animal looks like a duck. Suppose the measured sound was actually from a frog who had a cold so that it sounded like the ideal duck. This type of uncertainty is not incorporated in the fuzzy model since it stems from randomness rather than similarity.

4.3 A Framework for Fuzzy Probability

We have referred to the two most popular views on "uncertainty", probability and fuzziness, without being exhaustive with respect to this subject. We discussed that probability captures the essence of occurrence, whereas fuzziness captures the essence of similarity. We showed that both types of uncertainty are firmly rooted in the (western) tradition of science. We have showed that in the probabilistic framework, a priori decisions have to be made with respect to "belonging", of which the certainty is no taken into account. On the other hand we have shown that the concept of randomness is not taken into account in a fuzzy framework. There may be more types of uncertainty and there are several subtleties in this matter which we may have ignored. The main point is, however that in many discussions on probability and fuzziness (see [48]) only one of the uncertainty models is chosen as *the* uncertainty model. Whereas we have tried to clarify that they both capture a specific type of uncertainty. The question now is whether they can indeed be combined.

We started our discussion on (un)certainty by noting that uncertainty is necessary to make sense of the world we observe. We assumed that this uncertainty arises from non-specificity. In view of the two types of uncertainty we can state the following on non-specificity. If we do not specify the exact conditions under which our observations occur we introduce randomness. However, if we do not specify exactly the concepts in which we want to express our observations, we introduce fuzziness. Non-specificity is usualy the result of (implicit) generalization. Hence, the two types of uncertainty essentially arise from a single principle: generalization. For this reason it is possible to combine them in a single framework, for they are two faces of the same man. Therefore, we regard fuzziness and probability as "complementary" to each other, and combining the two types in one framework for inference will lead to a more complete model of uncertainty.

To arrive at a mathematical framework in which both types of uncertainty exist, we will use the concept of "the probability" of a fuzzy event". The mathematical definition of the probability of a fuzzy event has already been suggested in [147]. The fuzzy probability space can be defined as the triplet $(\mathcal{R}, \mathcal{F}, P)$,

where $\mathcal{F}$ is a σ-field, for which holds that:

$$\begin{aligned} P(A) &\geq 0 \quad \forall \quad A \in \mathcal{F} \\ P(\mathcal{A}) &= 1 \\ P(A \cup B) = P(A) + P(B) \quad &iff \quad A \cap B = \emptyset \end{aligned} \tag{4.13}$$

In the same way as in the section on probability, the distribution function is defined by 4.6 and the probability density function by 4.8. The probability of the fuzzy event A can then be defined as:

$$P(A) = E[\mu_A] = \int \mu_A(x) dF(x) = \int \mu_A(x) p(x) dx \tag{4.14}$$

This definition was suggested by Zadeh, and has later been adopted and supported by several authors [8] [105] [66] [143] [127]. However the extension to conditional probabilities, independence etc. depends on the type of operators chosen. Further, also the authors interpret it differently. In [74][75] a rigorous analysis can be found of the probability of a fuzzy event and fuzzy random variables, where the fuzzy algebra is connected with multivalued logics. In [141] the probability of fuzzy events was extended to fuzzy probability. The difference is that for probabilities of fuzzy events a single number represents the probability, whereas in fuzzy probability the probability is expressed as a (fuzzy) set. More on fuzzy probabilities can be found in [145] [69]. Most publications on connecting fuzziness and probability stem from the seventies and eighties. Nowadays, papers on probabilities of fuzzy events or their use in applications are hardly encountered in literature. This may due to the fact that another extension of probability has emerged, which is essentially an extension of comparative probability (see [45]), and especially popular in the field of expert systems and reasoning. This extension is called possibility theory and directly related to Shafer's theory of evidence [121]. This extension has also been introduced by Zadeh [149] and has been further extended by Yager [144]. A comprehensive overview of possibility theory can be found in [30]. Finally we note that probability is one of many measures that can be extended in various ways to fuzzy-set theory. We refer to [139] for an overview of fuzzy-measure theory.

In as far as we know, despite the fair amount of literature on this subject, inference on the basis of the probability of a fuzzy event, has not yet been addressed explicitly, nor have we found applications of such an inference. Hence, in the remainder of this section we will give a framework for inference based on co-existence of probability and fuzziness. This framework will be build upon some algebra introduced in [147] and [127]. Before continuing with the mathematical framework for co-existence, we first give some considerations which may be used as guidelines. In this thesis we will often refer to the framework as *fuzzy probability*, where we always mean the probability of a fuzzy event.

4.3.1 Considerations for Co-existence

From the similarity interpretation of fuzzy sets, we reason as follows. Since a member is similar to the (ideal) concept (otherwise it would not be a member),

its features are also similar to the features of the concept. Therefore, the features of the member can be derived from the concept to the extent of the similarity of the member to the concept (membership). Vice versa, the features of the concept can be derived from the features of the members (a collection of the feature: average or measure) in as far as they are similar to the concept. In other words, the similarity is a modulator for generalizing features to the concept and for specifying features to the members. Note that this essentially bi-directional view of inheritance (not unusual in object-oriented programming) is only possible if the feature exists for both the concept and its members.

In order to deal with probability and similarity in a single inference framework, we extend the previous reasoning to probability. Hence, probability is viewed as a feature of concepts. Further, if the members are continuous, they possess the feature of probability density. If they are discrete, they possess the feature of discrete probability. So, a bi-directional view of inheritance is possible. In the continuous case the probability of the concept is the average (integral) of the densities of each member in as far as they are similar to the concept, and in the discrete case we obtain an average probability for the concept.

So if A is a fuzzy set with memberships $\mu_A(x)$ for $x \in \mathcal{R}$, then $P(A)$, the probability of A, can be expressed in the density $p(x)$ as:

$$P(A) = \int \mu_A(x)p(x)dx \tag{4.15}$$

which becomes a summation in the discrete case.

Since a member usually belongs to several concepts, we require that the features of a member can be expressed in the features of the concepts to the extent of its similarity to the concepts. In other words, the member is expressed in existing concepts. To this end it is often useful to let the sum of similarities with existing concepts be equal to one.

4.4 Fuzzy Probabilistic Algebra

In this section we will derive the basic notions for calculations with fuzzy probabilities. For all the notions it holds that if crisp sets are used then we obtain the usual probabilistic framework.

Intersection and union operator

For a complete fuzzy algebra, the intersection and union operator need to be defined. For that purpose we will define the notion of sets being *normalized disjunct*. Given m fuzzy sets A_s which are defined on the domain $\mathcal{R}$ for all $x \in \mathcal{R}$.

- *normalized disjunct:* $|A|^{\mathcal{R}} \Leftrightarrow \sum_{i=s}^{m} \mu_{A_s}(x) = 1 \ \ \forall \ \ x \in \mathcal{R}$

Given a probability space $(\mathcal{R}, \mathcal{F}, P)$ as defined previously for which $A_s \in \mathcal{F}$, and given further a density function $p(x)$ such that probabilities $P(A_s)$ can be

obtained from 4.14, then we have as an immediate consequence:

$$\sum_{s=1}^{m} P(A_s) = 1 \tag{4.16}$$

because:

$$\begin{aligned}
\sum_{s=1}^{m} P(A_s) &= \sum_{s=1}^{m} E[\mu_{A_s}(x)] \\
&= \sum_{s=1}^{m} \int \mu_{A_s}(x) p(x) dx \\
&= \int \sum_{s=1}^{m} \mu_{A_s}(x) p(x) dx \\
&= \int p(x) dx = 1
\end{aligned} \tag{4.17}$$

In the framework of fuzzy probability it therefore seems only logical to define the i-operator and the u-operator such that:

$$\begin{aligned}
&|A|^{\mathcal{R}} \Leftrightarrow \\
&P(\cup_s A_s) = P(\neg\emptyset) = 1 \\
&P(\cap_s A_s) = P(\emptyset) = 0
\end{aligned} \tag{4.18}$$

To imply this property the well-known bounded-sum and bounded-difference operators are used:

- *union*: $A \cup B \Leftrightarrow \mu_{A\cup B}(\mathbf{x}) = min\{1, \mu_A(\mathbf{x}) + \mu_B(\mathbf{x})\}\ \forall \mathbf{x} \in X^d$
- *intersection*: $A \cap B \Leftrightarrow \mu_{A\cap B} = max\{0, \mu_A(\mathbf{x}) + \mu_B(\mathbf{x}) - 1\}\ \forall \mathbf{x} \in X^d$

These operators were introduced by Yager [142] and satisfy the De Morgan laws. Further, they are equivalent to normal set union and intersection if A and B are crisp sets. Note that if we have $|A|^X \wedge |B|^Y$ that also $|A \wedge B|^{XY}$. That is, if we have sets on X that are normalized disjunct and some sets on feature Y that are normalized disjunct than also their extensions on the Cartesian product space are normalized disjunct. Therefore we have:

$$\cup_{ij}(A_i \wedge B_j) = 1 \tag{4.19}$$

$$\cup_{i}(A_i \wedge B_j) = B_j \tag{4.20}$$

Finally we note that for any pair of sets defined on the same domain X^d we have:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{4.21}$$

Independence

Suppose we have a fuzzy event A and a fuzzy event B which are defined by $\mu_A(x)$ and $\mu_B(y)$, where $x \in X$ and $y \in Y$ measure different features. By definition (4.14):

$$P(A \wedge B) = E[\mu_{A \wedge B})] = E[\mu_A(x)\mu_B(y)] \tag{4.22}$$

since $\mu_{A \cap B}$ is symmetric, we immediately have:

$$P(A \wedge B) = P(B \wedge A) \tag{4.23}$$

Further, in case of independence $p(x, y) = p(x)p(y)$:

$$P(A \wedge B) = E[\mu_A(x)\mu_B(y)] = E[\mu_A(x)]E[\mu_B(y)] = P(A)P(B) \tag{4.24}$$

Hence, the independence rule for probabilities of crisp events A *and* B holds also for fuzzy events. Further,

$$\begin{aligned} P(A \vee B) &= E[\mu_{A \vee B})] \\ &= E[\mu_A(x) + \mu_B(y) - \mu_A(x)\mu_B(y)] \\ &= E[\mu_A(x)] + E[\mu_B(y)] - E[\mu_A(x)\mu_B(y)] \\ &= P(A) + P(B) - P(A \wedge B) \end{aligned} \tag{4.25}$$

Hence, the common rule for the probability of crisp events A *or* B also holds for fuzzy events, and is also symmetric.

Conditionals

Implications are usually modeled by conditional probabilities. The conditional probability $P(A|B)$, A and B defined as above, is given by:

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} \tag{4.26}$$

In case of independence:

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{P(A)P(B)}{P(B)} = P(A) \tag{4.27}$$

Also, due to (4.23) we have:

$$P(A|B)P(B) = P(B|A)P(A) = P(A \wedge B) \tag{4.28}$$

Also notice that if $|B|^Y$:

$$P(A|\cup_s B) = \frac{P(A \wedge \neg\emptyset)}{P(\neg\emptyset)} = \frac{P(A)}{1} = P(A) \tag{4.29}$$

Further we can introduce conditional density functions by noting that:

$$P(A) = E(\mu_A(x)) = \int \mu_A(x)p(x)dx \tag{4.30}$$

$$\Rightarrow \int \frac{\mu_A(x)p(x)}{P(A)}dx = 1 \tag{4.31}$$

Hence, we define the conditional density function $p(x|A)$ for a given fuzzy event A as:

$$p(x|A) = \frac{\mu_A(x)p(x)}{P(A)} \tag{4.32}$$

In case we have $|A|^X$, then we immediately have:

$$p(x|\cup_s A_s) = \frac{\mu_{\cup_s A_s}(x)p(x)}{P(\cup_s A_s)} = p(x) \tag{4.33}$$

but also

$$\sum_{s=1}^{m} p(x|A_s)P(A_s) = \sum_{s=1}^{m} \frac{\mu_{A_s}(x)p(x)}{P(A_s)}P(A_s) = p(x) \tag{4.34}$$

therefore: $p(x|\cup_s A_s) = \sum_{s=1}^{m} p(x|A_s)P(A_s)$, which is only natural.

Estimation

The expectation can be estimated (in the continuous case) (shown by [76]) by:

$$P(A) = E[\mu_A(x)] \approx \frac{1}{N}\sum_{i}^{N} \mu_A(x_i) \tag{4.35}$$

which is exact in the discrete case. Since we saw in the previous section that counting can be generalized to counting membership values, it is easy to interpret the estimate of the probability of a fuzzy event as a relative count of the fuzzy event A with respect to the sure event. Note that when we use the expectation of the estimation in (4.35), and assume that x_i is an i.i.d. random variable:

$$\begin{aligned} E[\frac{1}{N}\sum_{i}^{N} \mu_A(x_i)] &= \int \frac{1}{N}\sum_{i}^{N} \mu_A(x_i)p(x_i)dx_i \\ &= \int \mu_A(x_i)p(x_i)dx_i \end{aligned} \tag{4.36}$$

which indeed results in the expectation of event A. As an extension of counting we note that $P(A|B)$ can be estimated by:

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} \approx \frac{\sum_i^N \mu_A(x_i)\mu_B(y_i)}{\sum_i^N \mu_B(y_i)} \tag{4.37}$$

4.5 Fuzzy Probabilistic Logic

Instead of the "algebraic" view used in Fuzzy Probabilistic Algebra, we can also employ a "logical" view. Essentially these views become equivalent when fuzzy probabilities are used; the difference lies in the representation.

Given an implication like:

$$R: \ x \text{ is } A \Rightarrow y \text{ is } C \tag{4.38}$$

then the truth table for the implication according to fuzzy logic is the fuzzy relation μ_R. Apart from this implication, we also have the probabilistic implication (strength) $P(C|A)$. Therefore, we let $P(C|A)$ represent the occurrence of set (concept, event) C given set A and we let μ_R represent the similarity implication *given* that the C occurs when A occurs. We can thus write:

$$R: \ x \text{ is } A \overset{P(C|A)}{\Longrightarrow} y \text{ is } C \tag{4.39}$$

Since in our view both similarity and probability are types of uncertainty that modulate the truth, we modify the truth table for the implication $\mu_R(x,y)$ to $\mu_{R'}(x,y) = \mu_R(x,y)P(C|A)$, such that we get for inference:

- *inference*: $C' = A' \circ R'$
- *set inference*: $\mu_{C'}(y) = \sup_{x \in X}\{\mu_{A'}(x)\mu_R(x,y)P(C|A)\}$
- *singleton inference*: $\mu_{C'}(y) = \mu_A(x_0)\mu_C(y)P(C|A)$

In case of multiple rules for C, the certainty is aggregated by using the fuzzy union:

- *aggregation*: $C = A' \circ \cup_i R'_i$

For a normalized disjunct rule base (that is $|A \wedge C|^X$) the singleton inference for set C becomes:

$$\mu_{C'}(y) = \sum_i \mu_{A_i}(x_0)\mu_C(y)P(C|A_i) \tag{4.40}$$

Note that if the conditional probabilities are one, we obtain the usual fuzzy logic framework.

Classification

Having a rule base consisting of rules of the form:

$$R: \ x \text{ is } A_s \overset{P(C_k|A_s)}{\Longrightarrow} y \text{ is } C_k \tag{4.41}$$

where y *is* C_k denotes the singleton "the decision is class C_k", we immediately obtain (because of the aggregation defined in 4.40):

$$\mu_{C'_k} = \sum_{s=1}^{m} \mu_{A_s}(\mathbf{x})\hat{P}(C_k|A_s) \tag{4.42}$$

4.6 Derivation of the Double-Kernel Estimator

Recalling that the fixed-bandwidth uniform double-kernel estimator for a one-dimensional probability density function is given by:

$$\hat{p}(x) = \frac{1}{nV_\mu H_\mu} \sum_{s=1}^{m} \mu(\frac{x - x_s}{H_\mu}) \sum_{i=1}^{n} \mu(\frac{x_s - x_i}{H_\mu}) \tag{4.43}$$

where we have used a kernel ν equal to kernel μ and chosen μ such that:

$$\sum_{i=1}^{n} \sum_{s=1}^{m} \mu(\frac{x_i - x_s}{H_\mu}) = n \tag{4.44}$$

We may rewrite this in terms of probabilities by using (4.35):

$$\hat{P}(A_s) = \frac{1}{n} \sum_{i=1}^{n} \mu(\frac{x_i - x_s}{H_\mu}) = \sum_{i=1}^{n} \mu_{A_s}(x_i) \tag{4.45}$$

Then the uniform double-kernel estimator becomes:

$$\hat{p}(x) = \frac{1}{V_\mu H_\mu} \sum_{s=1}^{m} \mu_{A_s}(x) \hat{P}(A_s) \tag{4.46}$$

Having rewritten the double-kernel estimator, we will now show that it can be derived by using Fuzzy Probabilistic Algebra. Given a normalized disjunct lattice A, $|A|^X$, then:

$$p(x) = \sum_{s=1}^{m} p(x|A_s) P(A_s) \tag{4.47}$$

noting that $p(x|A_s)$ is given by:

$$p(x|A_s) = \frac{\mu_{A_s}(x) p(x)}{P(A_s)} \tag{4.48}$$

If μ_{A_s} is a "small" set, such that $p(x)$ is constant wherever $\mu_{A_s}(x) \geq 0$ (thus approximating $p(x)$ as piece-wise constant function), we can assume locally uniform values $p(x) = c$. We thus obtain:

$$P(A_s) = \int \mu_{A_s}(x) p(x) dx = c \int \mu_{A_s}(x) dx = c V_{\mu_{A_s}} \tag{4.49}$$

Hence, the conditional density can be estimated by

$$\hat{p}(x|A_s) = \frac{\mu_{A_s}(x) p(x)}{P(A_s)} = c \mu_{A_s}(x) \frac{1}{c V_{\mu_{A_s}}} = \frac{\mu_{A_s}(x)}{V_{\mu_{A_s}}} \tag{4.50}$$

Substituting in (4.47), we get the uniform double-kernel estimator:

$$\hat{\hat{p}}(x) = \sum_{s=1}^{m} \hat{p}(x|A_s)\hat{P}(A_s) = \sum_{s=1}^{m} \frac{\mu_{A_s}(x)}{V_{\mu_{A_s}}} \hat{P}(A_s) \tag{4.51}$$

Note however, that it is not necessary to have an uniform distribution of kernels. Intuitively it is not difficult to imagine a partition which is not uniform, but which is normalized disjunct, based on a clustering.

Concluding: we may interpret the double-kernel estimator as an estimation based on fuzzy sets, using:

$$p(x) = \sum_{s=1}^{m} p(x|A_s)P(A_s) \tag{4.52}$$

where the kernels specify the fuzzy sets of our reference frame in which we partition the outcome space. As we have seen in the Chapter 3, the only way to be certain that we have obtained the density function is to have an nearly infinite number of sets, which needs a nearly infinite number of observations x_i. Further, the broader (smoother) our set A_s the better the estimated probability of $P(A_s)$, but the less valid the assumption that $p(x)$ is uniform over the entire set. Vice versa, the smaller our sets, the better our assumption of uniformity over the set, but the worse our estimation of the probability of the event (given a fixed number of observations).

4.7 Classification using Fuzzy Probability

In the previous chapter we explained in detail how equation 4.51 can be used for classification problems. For each class C from the set of classes $\mathcal{C}$, we can approximate the conditional density functions $p(x|C)$ by using 4.51 and the a posteriori probability is then found by applying the Bayes Rule:

$$P(C|x) = \frac{\hat{p}(x|C)P(C)}{\sum_C \hat{p}(x|C)P(C)} \tag{4.53}$$

From a conceptual point of view, this is a rather cumbersome approach. For each class a distribution of kernels is necessary; in other words, the sets are class-conditional. The advantage of such an approach is that the class-conditional covariances can be taken into account by using class-conditional kernels (membership functions) as shown previously. In this section we will simplify the approach by using a single distribution of kernels, irrespective of the classes. Such a single distribution will be referred to as a *fuzzification* of the outcome space, as opposed to a *quantization*.

Given a fuzzification of the d-dimensional outcome space $\mathcal{R}^d$ into m different sets $A_s \in \mathcal{A}$, for which holds:

$$\sum_{s=1}^{m} \mu_{A_s}(\mathbf{x}) = 1 \quad \forall\, \mathbf{x} \in \mathcal{R}^d \tag{4.54}$$

Given further a set $\mathcal{C}$ of c classes $C_k \in \mathcal{C}$, then we can estimate the class-conditional density functions by using 4.51:

$$\hat{p}(\mathbf{x}|C_k) = \sum_{s=1}^{m} \frac{\mu_{A_s}(\mathbf{x})}{V_{\mu_{A_s}}} \hat{P}(A_s|C_k) \tag{4.55}$$

where:

$$V_{\mu_{A_s}} = \int \mu_{A_s}(\mathbf{x}) d\mathbf{x} \tag{4.56}$$

Applying Bayes Rule to obtain the a posteriori probability:

$$\hat{P}(C_k|\mathbf{x}) = \frac{\sum_{s=1}^{m} \frac{\mu_{A_s}(\mathbf{x})}{V_{\mu_{A_s}}} \hat{P}(A_s|C_k) P(C_k)}{\sum_{k=1}^{c} \sum_{s=1}^{m} \frac{\mu_{A_s}(\mathbf{x})}{V_{\mu_{A_s}}} \hat{P}(A_s|C_k) P(C_k)} \tag{4.57}$$

If we use

$$P(A_s|C_k)P(C_k) = P(C_k|A_s)P(A_s) \tag{4.58}$$

the a posteriori can be written as:

$$\hat{P}(C_k|\mathbf{x}) = \frac{\sum_{s=1}^{m} \mu_{A_s}(\mathbf{x}) \hat{P}(C_k|A_s) \frac{P(A_s)}{V_{\mu_{A_s}}}}{\sum_{k=1}^{c} \sum_{s=1}^{m} \mu_{A_s}(\mathbf{x}) \hat{P}(C_k|A_s) \frac{P(A_s)}{V_{\mu_{A_s}}}} \tag{4.59}$$

Substituting:

$$W_s = \frac{P(A_s)}{V_{\mu_{A_s}}} \tag{4.60}$$

gives:

$$\hat{P}(C_k|\mathbf{x}) = \frac{\sum_{s=1}^{m} \mu_{A_s}(\mathbf{x}) \hat{P}(C_k|A_s) W_s}{\sum_{k=1}^{c} \sum_{s=1}^{m} \mu_{A_s}(\mathbf{x}) \hat{P}(C_k|A_s) W_s} \tag{4.61}$$

The quantity W_s reflects the density of sets A_s; as such it is a quantity determining the weight or importance of set A_s. Since the covariances of the separate class distributions are not taken into account, accurate estimates of the weights cannot be expected. Therefore, we assume that all weights are equal, which is equivalent to assuming an uniform density $p(\mathbf{x})$. The a posteriori probability then simplifies to (the denumerator amounts to one):

$$\hat{P}(C_k|\mathbf{x}) = \sum_{s=1}^{m} \mu_{A_s}(\mathbf{x}) \hat{P}(C_k|A_s) \tag{4.62}$$

Note that this result is equivalent to the result obtained in the section on Fuzzy Probabilistic Logic. Therefore, the algebraic view and the logical view are consistent, even though the algebraic view is somewhat richer in representation (e.g. the weigth W_s).

Estimation

Having n examples $x_i \in I \subset \mathcal{R}^d$, we calculate the estimate for $P(C|A)$ from 4.37:

$$\hat{P}(C_k|A_s) = \frac{\sum_{i=1}^{n} \mu_{A_s}(\mathbf{x}_i)\mu_{C_k}(\mathbf{x}_i)}{\sum_{i=1}^{n} \mu_{A_s}(\mathbf{x}_i)} \tag{4.63}$$

where $\mu_{C_k}(\mathbf{x}_i)$, the certainty of the class of the example, is provided by an expert (usually "0" or "1" in classification problems).

If the fuzzification does not exactly amounts to one, then the following estimates should be used:

$$\hat{P}(C_k|\mathbf{x}) = \frac{\sum_{s=1}^{m} \mu_{A_s}(\mathbf{x})\hat{P}(C_k|A_s)}{\sum_{s=1}^{m} \mu_{A_s}(\mathbf{x})} \tag{4.64}$$

which can be recognized as the normalized fuzzy mean. The conditional probabilities are in this case estimated by:

$$\hat{P}(C_k|A_s) = \frac{\sum_{i=1}^{n} \frac{\mu_{A_s}(\mathbf{x}_i)}{\sum_{s=1}^{m} \mu_{A_s}(\mathbf{x}_i)} \mu_{C_k}(\mathbf{x}_i)}{\sum_{i=1}^{n} \frac{\mu_{A_s}(\mathbf{x}_i)}{\sum_{s=1}^{m} \mu_{A_s}(\mathbf{x}_i)}} \tag{4.65}$$

It is easy to see that this reduces to equations 4.62 and 4.63 if we substitute in 4.64 and 4.65:

$$\mu_{A'_s}(\mathbf{x}) = \frac{\mu_{A_s}(\mathbf{x})}{\sum_{s=1}^{m} \mu_{A_s}(\mathbf{x})} \tag{4.66}$$

4.7.1 Example

The following example illustrates the use of 4.64 and 4.65. For a one-dimensional, two-class problem, 50 examples were drawn from a normal density function with mean -1 and variance 1, and 50 examples were drawn from a normal distribution with mean $+1$ and variance 1. Figure 4.3 shows the theoretical density functions and the theoretical a posteriori probabilities.

The a posteriori probabilities were estimated through a quantization and a fuzzification, both consisting of 5 sets (concepts, events), see Figure 4.4. The sets may reflect the qualifications "very low", "low", "normal", "high", "very high". For a quantization 4.62 reduces to an ordinary histogram approach, due to the bins. Within the interval [-5,5] the bins exactly amount to one; outside this interval the bins are not defined, and results are therefore only valid within the specified interval. For the fuzzification we used Gaussian functions separated by $\sqrt{2\pi}\,\sigma = 2$, such that the sum approximately equals one on the interval [-4,4] (hence $\sigma = 0.8$). Normalization, especially necessary outside the interval [-4,4], is obtained from 4.64 and 4.65; this is possible because even outside the interval the sets are defined due to the Gaussian functions.

The resulting a posteriori probabilities are given in Figure 4.5. Clearly, the sets were chosen rather clumsily, since one of the sets is exactly positioned on

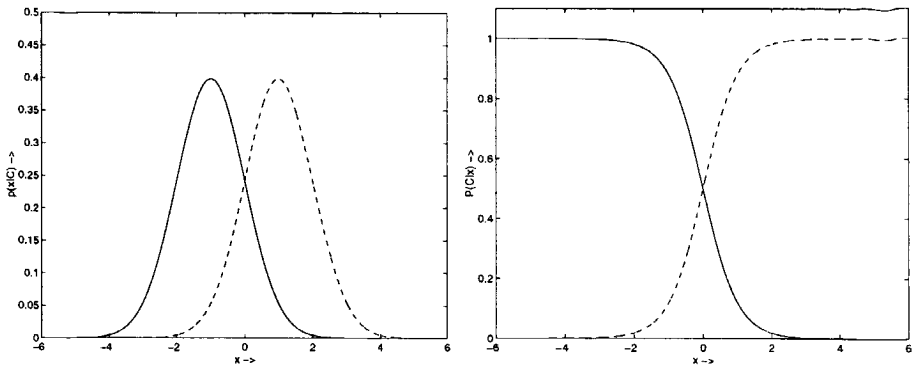


Figure 4.3: *Left figure shows the density functions from which the examples were drawn, right figure shows the resulting a posteriori probabilities (using equal a priori's). Solid lines denote the first class and dashed lines denote the second class.*

the decision region. However, it illustrates how a quantization is much more dependent on the position of the sets than a fuzzification.

The experiment has been repeated for two-dimensional normal distributions at positions (-1,-1) and (1,1); for each class 150 examples were generated. The decision boundary should in theory be the line $y = -x$. Results are shown in Figure 4.6 and 4.7. Note that in the fuzzification approach, the decision boundary is not necessarily perpendicular to the axes, whereas it is in the quantization approach.

4.8 Conclusion

We put forward the idea that there are two different types with which uncertainty can be modeled: probability and fuzziness. Both types arise from the same principle: generalization. Based on assumptions about knowledge, uncertainty and Zadeh's definition of the probability of a fuzzy event, both types of uncertainty are synthesized into a single framework: Fuzzy Probability. This framework takes both similarity and randomness into account for inference. The framework is such that if the probabilities are set to one, a fuzzy inference framework is obtained, whereas in the case of crisp sets the usual probability framework is obtained. Estimators for fuzzy probabilities as well as for fuzzy conditional-probabilities are provided. It is shown that the double-kernel estimator of Chapter 3 can be derived from the fuzzy probabilistic framework by assuming locally uniform densities.

The framework is suitable for classifier design on the basis of an a priori fuzzification. In essence, this fuzzy probabilistic classifier can be regarded as a special case of the double-kernel estimator, when we assume locally independent fea-

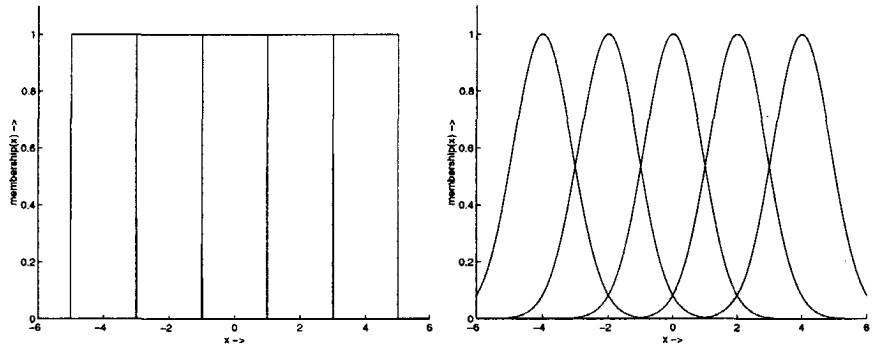


Figure 4.4: *Discretization by five sets, quantization on the left and fuzzification on the right. Sets are chosen such that the summation is (approximately) equal to one.*

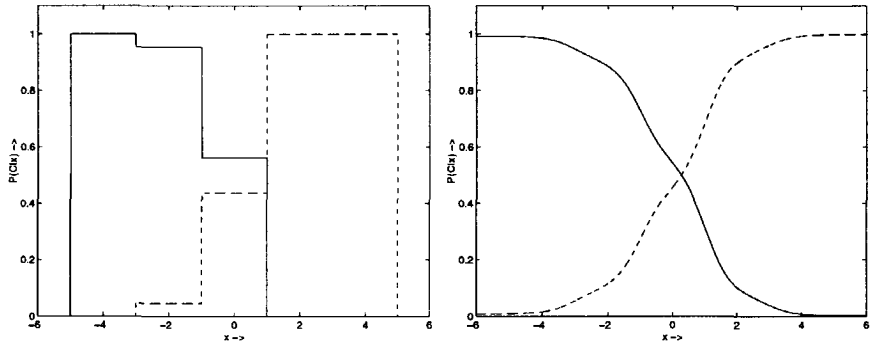


Figure 4.5: *Results for quantization on the left and fuzzification on the right. Results of quantization are only valid within the interval [-5,5]. Solid lines show the results for the first class, dashed lines show the results for the second class.*

tures and an uniform distribution of the data density. The main difference with the double-kernel estimator is that class-conditional covariances are not taken into account. Experiments on synthetic data show that a fuzzy discretization (fuzzification) results in a more accurate classifier than a quantization does, due to larger independence of the a priori discretization. By using a fuzzification, we can accurately estimate decision boundaries that are not perpendicular to the feature axes. These properties are useful in addressing the data fit versus mental fit problem in rule induction for continuous domains.

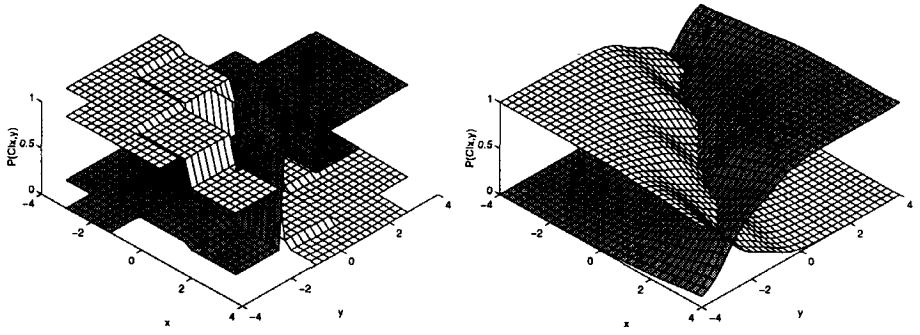


Figure 4.6: *Results for quantization on the left and fuzzification on the right in a two-dimensional problem. Empty spaces in the quantization approach denote areas where no examples have been observed.*

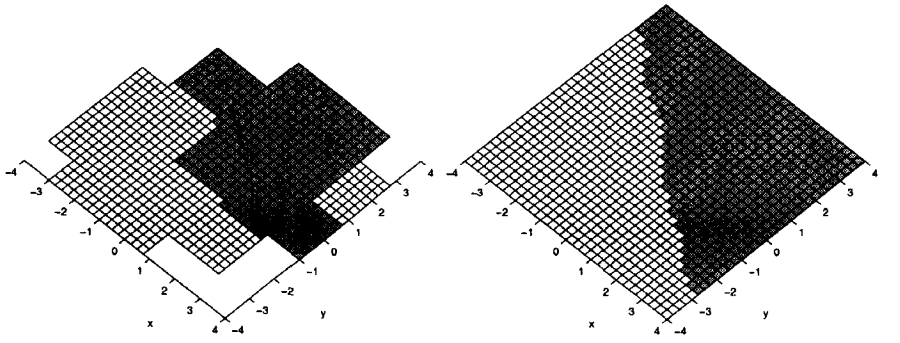


Figure 4.7: *Illustration of the decision boundaries for the quantization (left) and fuzzification (right). In the quantization approach, the decision boundaries are piecewise perpendicular to the x- and y-axis leading to relatively inaccurate classification results.*

Chapter 5

Fuzzy Probabilistic Rule Induction

The advantages of applying rule induction are the ability of explaining a classification and the ability of further reasoning. Both advantages are useful in decision-support systems and knowledge-based systems. If these advantages are not used and the goal is solely to obtain a decision, one may as well rely on other suitable approaches as found in Pattern Recognition. This chapter will present a rule induction technique that uses the fuzzy probabilistic framework of Chapter 4.

Approach

The explanation of a concept (of a class) seems only useful to us if the explanation itself is based on a relevant concept. For example, explaining that a patient is ill because he has a high blood pressure, is useful if this high blood pressure is indeed meaningful to an expert. It may be the case that a high blood pressure can, besides "illness", be associated with some medicine, with the age of the patient, or with the profession of the patient. In this respect it should be noted that is not uncommon to regard human knowledge as a very large associative map with many interconnections. So, the meaning of an explanation, and its use for further reasoning, essentially relies on the associations to other concepts existing in the mind or in other rule bases. Ultimately, any explanation in terms of a rule is based on a conjunction or disjunction of symbols, where the symbols are obtained from discretization. Therefore, if rule induction is used for a single classification problem, then the discrete values of the features should ideally be treated as given a priori; the total set of a priori discrete values will be denoted by the term *reference frame*. A reference frame can be obtained by any method, as long as it provides a meaning to the feature value irrespective of the decision. If the discrete values are too much "tuned" to the specific classification problem at hand, then the associations to problems that are not regarded may be lost, and thus their meaning and their use for further reasoning may disappear as

well. The discovery of the sub-concepts (sub-classes, within-concept clusters) can be lost for the same reason.

The previous paragraph motivates using the data-information-knowledge paradigm, where the formation of discrete values is somewhat *de-coupled* from the actual rule induction. The discretization is applied to the data to create the information, and the rule induction is applied to the information to create the knowledge. As pointed out in Chapter 2, a choice in favor of mental fit is often a choice against data fit. One of the key elements where this choice is of particular importance, is the discretization of the feature values. If a very refined discretization is used, the search space is large, the number of rules induced will most likely be large, the need for data is high, but the classification error may be small. On the other hand, if a very rough discretization is used, the search space is rather small, the number of rules induced will most likely be small, the need for data is low, but the classification error may be large. Hence, a de-coupling of the discretization and the rule induction is not trivial, because they are mutually dependent. By requiring a better mental fit in the discretization, some data fit in terms of accuracy may be sacrificed. However, it was pointed out in Chapter 4 that a fuzzification provides a much better means for reducing the influence of the discretization on the final decision boundaries than a quantization. Not only are the decision boundaries less dependent on the position of the fuzzy sets, but the decision boundaries obtained are not necessarily parallel to the axes of the feature space. The accuracy of the fuzzification in the fuzzy probabilistic approach does not arise from the fuzzification alone, but arises also from the fuzzy probability which handles noise. Therefore, the fuzzification combined with the probabilistic approach makes fairly accurate decision-making possible with a reference frame that is not specific for a particular decision problem.

In this chapter the Fuzzy Probabilistic Induction (FPI) framework will be discussed for generating classification rules of the form "If Premise Then Class". Although it mainly focuses on discussing the FPI approach in the light of classification problems, the FPI approach is certainly not restricted to it. In principle any implication from one fuzzy set to another can be learned, e.g. control rules like: "if pressure is low then temperature is set to high". This chapter concludes with some experiments using an implementation of the FPI framework called FILER: Fuzzy Inductive Learning of Expert Rules

Other Approaches

Recently, several rule induction algorithms using fuzzification have been proposed, see [22, 27, 62] for tree induction and [1, 2, 50, 59, 80, 103, 130, 138] for production rules. Other approaches, mainly used for fuzzy control systems, are based on fuzzy clustering, see [7]. Most of the rule induction algorithms use the example that has highest membership in a fuzzy concept and associate the class of this single example to the entire concept. As indicated by [130] such an approach is not very useful in "noisy" domains, and for accurate results this approach heavily relies on an optimization over the fuzzification. For these reasons [130] associate the class having the highest *average* membership. This

is one of the few approaches where statistics are combined with fuzziness. To some extent such a combined approach can also be recognized in the article by Janikow [62] on fuzzy tree-induction (although it is not mentioned as such).

5.1 Overview of FPI

The FPI approach for generating production rules in classification problems follows the Data-Information-Knowledge paradigm in the way depicted in Figure 5.1. First, the data are transformed from the data layer to the information layer by means of fuzzification using a reference frame. The reference frame can be obtained either from expert interviews or from clustering. The information layer is a rule base consisting of the most specific rules that can be formed. Each example (instance + decision) in the data layer should be covered, at least partially, by a specific rule. An information-theoretic approach is used to generalize the rules in the information layer to form the knowledge layer. The knowledge layer is a rule base containing general rules. Each specific rule in the information layer should be covered by a general rule from the knowledge layer. The knowledge layer is used for classifying and explaning new instances. The

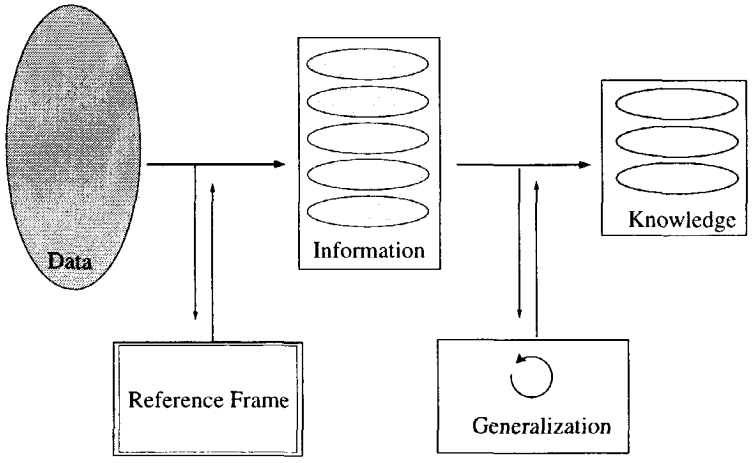


Figure 5.1: *DIK paradigm.*

reference frame, the information layer, the knowledge layer, and the classification will be discussed in the following sections. But first we will introduce some notations. The feature space will be denoted by Ω. A d-dimensional instance is denoted by $\mathbf{x}$, an element of the instance space $\mathcal{X}^d \subset \Omega$, and the i-th instance (observation) from the instance space will be denoted by the scalar $\mathbf{x_i}$. Each dimension is formed by a feature X. As an example we will frequently use the properties X_A, X_B, X_D. For each feature a lattice of sets is defined which will be denoted by A, B, D, containing a number of a, b, d fuzzy sets, respectively.

The classes are formed by C, having a total number of c classes. Rules will be of the *conjunctive* form (maximally 3-dimensional for notational convenience):

H^k_{uvw}: **If** x_1 is A_u **And** x_2 is B_v **And** x_3 is D_w **Then** y is C_k

which is more conveniently written as:

H^k_{uvw}: $R_{uvw} \Rightarrow C_k$

with $R_{uvw} = A_u \wedge B_v \wedge D_w \Leftrightarrow \mu_{R_uvw}(x_1, x_2, x_3) = \mu_{A_u}(x_1)\mu_{B_v}(x_2)\mu_{D_l}(x_3)$

Here the A_u, B_v, C_k, D_w's are one-dimensional fuzzy sets, the extent to which a set like $A_u \in A$ occurs due to $x \in X_A$ is modeled by a membership function $\mu_{A_u}(x)$ based on similarity to concept A_u. C_k is also a fuzzy set, and is modeled by $\mu_{C_k}(y)$, here y is the decision or the classification (hence, the fuzzy set C_k is regarded as a fuzzy singleton). Further, the certainty of the implication is represented by the conditional probability $P(C_k|R_{uvw})$. For obtaining the conditional probabilities and the final classification, the results from Chapter 4 will be used.

5.2 Reference Frame

For the fuzzification, a reference frame is needed that consists of fuzzy sets for each feature in the feature space. It is required that each feature is fuzzified individually so that the final classification can be explained in terms of the individual features. There are basically two methods for obtaining a reference frame for a feature:

- expert interviews,
- clustering.

The goal of the expert interviews is to obtain fuzzy sets that reflect the expert's opinion as closely as possible. These methods can be subdivided in direct methods such as "direct rating" [21, 70, 132, 133], and indirect methods such as pairwise comparisons [21, 117, 131]. The direct methods seem less laborious, which may be beneficial for dealing with experts, but the indirect methods may reflect the actual opinions more consistently. For clustering, several possibilities exist as well, for example the fuzzy c-means algorithm [11] or the K-means clustering algorithm [4], for an overview of clustering, see [9]. With the fuzzy c-means algorithm the fuzzy sets are directly obtained. With the K-means algorithm, only the prototypes are obtained and the sets are still to be completed.

Any of these methods can be used to obtain a reference frame, and any fuzzification can be used even if the fuzzification is not normalized (see also the previous chapter). However, for the FPI approach some preferences exist which are nearly all based on the analysis in Chapter 3 of the double-kernel estimator,

of which FPI can be thought as a special case. First, the number of fuzzy sets should be sufficiently high. This has been motivated from a sampling point of view in Chapter 3. Second, the fuzzy sets defined for a feature should be normalized disjunct. This preference requires that the summation:

$$\sum_{u=1}^{a} \mu_{A_u}(x) = 1 \quad \forall x \in \mathcal{X} \tag{5.1}$$

holds for all features X that form the feature space. The reason for this is partially motivated by the fact that it helps to normalize the probabilities, as pointed out in Chapter 4. But in Chapter 3 it has also been required so that maximum use is made (in terms of resolution) of the number of kernels. The third preference is that the fuzziness of a set must be restricted to its nearest neighbors. This preference is motivated by observing that the double-kernel estimator only converges to the "real" pdf if a sufficiently small kernel is used, as has been demonstrated in Chapter 3.

The approach followed in this chapter is requiring a number of clusters of the expert, using K-means clustering to find the prototypes and then forming the sets on the basis of relative (weighted) distances, e.g. Gaussian functions. We can also find the number of clusters by using an appropriate cluster-validation scheme, a default setting (e.g. the number of classes) or can even be chosen de facto by repeating the rule induction process for several numbers of clusters and choosing the one that leads to the "best" rule base. However, the last option should be carried out carefully in order not too tune the reference frame too much, which is motivated at the beginning of this chapter. The Gaussians consists of a left-hand and a right-hand side with different standard deviations σ, for obtaining a normalization. The standard deviation for one side is obtained by dividing the one-dimensional distance between two prototypes by $\sqrt{2\pi}$. An example of a K-means approach using three clusters and Gaussian functions is shown in Figure 5.2. Instead of Gaussians, trapezoids or a triangularization can be used. However, the disadvantage of such kernels is that they are truly zero outside some interval, which may lead to problems when new instances are classified which lie outside this interval. A Gaussian or any other exponential function may become very small outside some interval, but never zero. This functionality aids the generalization process since for each x there is a membership larger than zero.

5.3 Information and Knowledge

Both the information and the knowledge layer consists of rules H, for which some notions are important. The information layer consists of *specific rules* for which all features are specified:

H^k_{uvw}: $R_{uvw} \Rightarrow C_k$

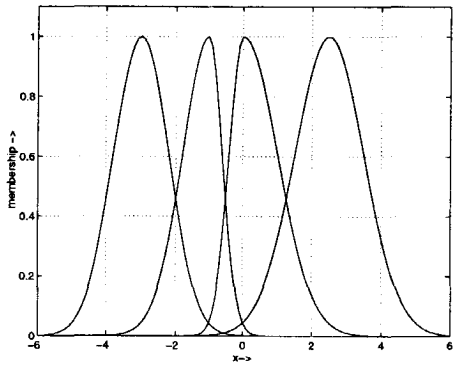


Figure 5.2: *Reference frame for a feature formed by fuzzy sets based on Gaussians. Summation of the fuzzy sets is approximately 1 in the interval [-3,2.5].*

The number of specific rules is equal to $s = abd$, where a, b, d are the number of fuzzy sets for each of the three features, respectively. This can lead to a very large information layer in high-dimensional feature spaces, but we return to this issue in section 5.6. The knowledge layer contains *general rules*, for which not all features need to be specified. An example of a general rule is:

H^k_{uw}: **If** x_1 is A_u **And** x_3 is D_w **Then** y is C_k

which is more conveniently written as:

H^k_{uw}: $R_{uw} \Rightarrow C_k$

In principle, general rules like:
$H^k_{uu'|w}$: **If** x_1 is $A_u \cup A_{u'}$ **Or** x_3 is D_w **Then** y is C_k

with:
$R_{uu'|w} = (A_u \cup A_{u'}) \vee D_w$

are also possible but will not be used for reasons to be discussed in the next section.

If we are not interested in the actual premise of a rule, specific or general, it will be denoted by H^k_g or H^k_h with $g, h \in 1, ..., r$, where r is the number of rules in the rule base.

Coverage

Given an instance $\mathbf{x}_i$, the extent to which it is *covered* by a specific rule is given by the membership:

$$\mu_{R_{uvw}}(\mathbf{x}_i) = \mu_{R_{uvw}}(x_{1i}, x_{2i}, x_{3i}) \tag{5.2}$$

The coverage of an example by a general rule H^k_{uw} is given by the product of the memberships of the specified features, for example:

$$\mu_{R_{uw}}(\mathbf{x}_i) = \mu_{R_{uw}}(x_{1i}, x_{2i}, x_{3i}) = \mu_{A_u}(x_{1i}) 1 \mu_{D_w}(x_{3i}) \tag{5.3}$$

Hence, $\mu_{R_{uw}}(\mathbf{x})$ is thought of as being an extension in the three-dimensional feature space. If the reference frame for each feature is normalized disjunct then we can also calculate $\mu_{R_{uw}}(\mathbf{x})$ by (see also section 4.4 in the previous chapter):

$$\begin{aligned} \mu_{R_{uw}}(\mathbf{x}) &= \sum_{v=1}^{b} \mu_{R_{uvw}}(\mathbf{x}) \\ &= \mu_{A_u}(x_1)\mu_{D_w}(x_3) \sum_{v=1}^{b} \mu_{B_v}(x_2) \qquad (5.4) \\ &= \mu_{A_u}(x_1)\mu_{D_w}(x_3) 1 \qquad (5.5) \end{aligned}$$

Apart from coverage of an example, a rule H^k_g is said to *cover* another rule H_h if the premise of the latter rule is a subset of the premise of the first: $R_g \subset R_h$. For example, the previously used general rule H^k_{uw} covers specific rule H^k_{uvw} because:

$$\mu_{R_{uw}}(\mathbf{x}) \leq \mu_{R_{uvw}}(\mathbf{x}) \quad \forall\, \mathbf{x} \in X^d \tag{5.6}$$

A rule is said to be *representative* for an example if it both covers the example with an extent larger than zero *and* if the decision on the basis of the rule is equal to the decision (the class) of the example. Sometimes we also say in this case that the example is *represented* by the rule.

Estimation of conditionals

Given n examples consisting of instances $\mathbf{x}_i$ with an associated classification (decision) $y_i \in \{C_1, ..., C_c\}$, the conditional probabilities can be estimated by (see 4.7):

$$P(C_j|R_h) = \frac{\sum_{i=1}^{n} \mu_{R_h}(\mathbf{x}_i)\mu_{C_j}(y_i)}{\sum_{i=1}^{n} \mu_{R_h}(\mathbf{x}_i)} \tag{5.7}$$

Here $\mu_{C_j}(c_i)$ is the membership of the classification being class C_j, as given by experts. Usually examples have a single label, say $y_i = C_k$, so the membership

$\mu_{C_j}(y_i)$ is one and for all other classes it is zero. The rule decision is obtained from the maximum over all conditionals:

$$P(C_k|R_h) = \max_j\{P(C_j|R_h)\} \tag{5.8}$$

Therefore a rule H_h^k is said to be the most likely of all the c-possible rules given the same premise R_h, i.e. $\{H_h^j\}\ \ j \in 1,...,c$, which also exist and play an important role for inference.

The conditionals for the specific rules should be derived from 5.7, but the conditionals of the general rules can be derived from the specific rules. Notice that a general rule is essentially a union of all the specific rules covered by a general rule:

$$R_{uv} = \cup_{R_{uvw} \subset R_{uv}} R_{uvw} = \cup_w R_{uvw} \tag{5.9}$$

$$\Rightarrow \tag{5.10}$$

$$P(C_j|R_{uv}) = \frac{\sum_{v=1}^{d} P(R_{uvw})P(C_j|R_{uvw})}{\sum_{v=1}^{d} P(R_{uvw})} = \frac{\sum_{v=1}^{d} P(R_{uvw})P(C_j|R_{uvw})}{P(R_{uv})} \tag{5.11}$$

because the specific rules are normalized disjunct, the implication holds (see previous chapter) and we also have for the observed a priori probability $P(C_j)$:

$$\neg\emptyset = \cup_{R_{uvw} \subset \neg\emptyset} R_{uvw} = \cup_{uvw} R_{uvw} \Rightarrow \tag{5.12}$$

$$P(C_j) = \sum_{u=1}^{a}\sum_{w=1}^{b}\sum_{v=1}^{d} P(R_{uvw})P(C_j|R_{uvw}) \tag{5.13}$$

Properties like these (others can be easily derived) are useful since it is not necessary to store all (training) examples. It is sufficient to store the information: the set of specific rules. Since the information layer is finitely large and the instance space is usually not finite, this may be useful for incremental learning, see [93, 116] for some early work, and [34] for an overview. Past experiences can be stored in the specific rules and generalizations can be derived completely from the specific rules as long as the same reference frame is used.

5.4 Generalization

In this section we will discuss two basic schemes for arriving at the general rules in the knowledge layer (rule base) from the information layer, although many more schemes may exist. In essence the information layer already is a set of rules that can classify (as demonstrated in the previous section) and explain (as will be discussed in the next section) new instances. However, by further generalization the set of specific rules is reduced to a much smaller set of general rules whilst the same training examples are used for estimation of

the conditionals. This has at least two important implications. First, the rules present in the knowledge layer are usually easier to comprehend and thus give a better mental fit because they are much simpler (where the simplicity of a rule is related to the number of tests in the rule premise). Both from the user's point of view as well as from Occam's Razor point of view this is desirable, as argued before. Second, the rules may be able to classify new instances better than the specific rules because of two reasons. On one hand the specific rules may be too finely tuned to the training examples (over-fitting), on the other hand the general rules have a better estimate of the conditional probability due to the larger number of examples covered. However, the general rules may be too general; important details with respect to the decision boundaries may get lost due to generalization. We reason that, in high-dimensional feature spaces, where not all dimensions are relevant for the classification, general rules may have a larger predictive ability than the specific rules. Resuming: generalization has at least two advantages:

- fewer rules,
- simpler rules.

The consequences are that we obtain a rule base with which the system can reason faster and which the user can comprehend more easily. In some cases the general rules may even be better in classification performance as well.

In order to evaluate general rules we will use the J-information measure as discussed in Chapter 2 (see section 2.3.3). For a rule H_h^j, the J-measure is given by:

$$J(R_h) = P(R_h)\left[P(C_k|R_h)log(\frac{P(C_k|R_h)}{P(C_k)}) + (1 - P(C_k|R_h))log(\frac{1 - P(C_k|R_h)}{1 - P(C_k)})\right] \tag{5.14}$$

The higher its value, the better the rule. Since the J-measure compares the a priori probability with the a posteriori probability, it is also referred to as information gain. It may be questioned whether the information measure can be applied to the fuzzy probabilities, since they are not the regular probabilities for which information theory has been developed. We remark here that the discrete information measure deals with symbolic concepts for which an (un)certainty measure is available, so it does not really matter what these symbolic concepts actually represent. What is important is that the uncertainty measure satisfies some properties. We recall here that the fuzzy probabilities satisfy the Kolmogorov axioms, and that the notion of independence is valid. Further, we recall that the fuzzy probabilities of a set of normalized disjunct events amount to one. Therefore, we see no objection against using the usual definition of information.

5.4.1 Hypothesis Generation

There are many ways in which hypotheses may be generated to arrive at general rules, many of which rely on more or less heuristic methods. However, the way in which a hypothesis is formed determines the type of rules obtained. Therefore we first briefly motivate our choice for the type of general rules that we like to obtain.

Our main goal is to obtain simple rules that can be interpreted by the expert. The fewer the number of tests in a condition of the rule, the simpler it becomes. The easiest way to obtain such simple rules is by dropping as many conditions as possible. This means that we should use the *dropping condition* type of generalization, which leads to general rules of the conjunctive-premise form as discussed in Chapter 2, see 2.2.3. Another motivation comes from a well-known phenomenon in the area of statistical classifier design: the "peaking-effect" [33, 60], which is a result of the dimensionality problem encountered in 2.1. Briefly stated it says that for many classification problems it holds that the higher the dimensionality is, the better classes may be separated but the more difficult it becomes for a classifier to estimate the actual decision boundaries. The curse is that the latter effect may be so dominant that the total performance may even decrease when compared to a lower number of dimensions. To avoid this effect we should reduce the dimensionality of a rule by the *dropping condition* type of generalization. An additional advantage of looking for general rules of this kind is that the search space of possible general rules is rather restricted and hence the search is rather simple. The additional disadvantage is that the decision boundaries become somewhat biased, and parallel to the axis, which may reduce the performance if the features are not independent. Hence, for classification problems where (1) the number of dimensions is near the optimal number, and where (2) the features are highly dependent, the results of the general rules may not be better than the results of the specific rules. For such problems "region growing"[1] may be a better type of generalization. Although it may be appropriate in specific cases, region growing may also lead to more complex, relatively high-dimensional, rules.

We will use the following heuristic for the *dropping condition* type of generalization. Given a specific rule H^k_{uvw} (hence the most likely decision for this rule is C_k) and a second specific rule $H^j_{uv'w}$, we hypothesize as follows:

$$k = j \Rightarrow H^k_{uw} \tag{5.15}$$

$$k \neq j \Rightarrow H^k_v$$

$$j \neq k \Rightarrow H^j_{v'} \tag{5.16}$$

That is, if two rules share the same (most likely) conclusion, then the premise of the hypothesis is obtained by specifying the features that are shared. If the conclusion of the rules are different, then the premise of the hypothesis is obtained by specifying the features that are not shared. Note that the first way of

[1] i.e. forming disjunctive sets for a feature such as in the *adding alternative* principle or by climbing some qualifier hierarchy as in the *extending reference* principle

hypothesizing leads to *characteristic rules*, whereas the second way of hypothesizing usually leads *discriminative* rules. To us these seem logical heuristics. By comparing each specific rule with the other specific rules we can obtain many possible hypotheses. However these heuristics also restrict the search space, since usually not all possible hypotheses are generated. However, a disadvantage of these heuristics is that if the number of examples increases, then the computational time for hypotheses generation increases usually non-linearly.

5.4.2 Non-Disjoint Rule Induction

The goal in this approach is to find the set of most informative rules that completely explain or cover the set of specific rules. In this approach we are not concerned with the organization of the rules, so they need not be normalized disjunct. A simple way of obtaining this set of general rules is by searching the general rules having the highest J-information measure, the most informative rules, according to:

- step 0. Start with an empty rule base,
- step 1. for a specific rule H_s^k form the set $\mathcal{H}$ of possible hypotheses that *represent* H_s^k,
- step 2. add the rule having the highest value for the J-measure to the rule base,
- step 3. Repeat step 1 and step 2 for each specific rule H_s^k until all specific rules are *represented.*

This approach will be referred to as non-disjoint rule induction.

The advantage of this approach is that each specific rule is generalized to the best rule available to represent it. Therefore all the best rules will be present, the best rules being of interest in applications like knowledge discovery or data mining. Also when clusters (sub-classes) are present in the data that intersect (overlap) one another, this approach is useful since it does not require that the rules are normalized disjunct. A problem occurs, however, when the data set is extremely noisy (with noisy we mean that the probability density functions are widely spread, leading to a high degree of class overlap). In that case it may not be possible to find a (general) representative rule other than the specific rule itself. This disadvantage can be somewhat solved by the following scheme for very noisy data sets:

- step 0. Start with an empty rule base,
- step 1. for a specific rule H_s^k form the set $\mathcal{H}$ of possible hypotheses that *cover* H_s^k,
- step 2. add the rule having the highest value for the J-measure to the rule base,

- step 3. repeat step 1. and step 2. for each specific rule H_s^k until all specific rules are *covered.*

Note that "represented" implies "covered", but the inverse is not true. Hence, the second method is less restrictive and usually leads to a smaller rule base.

The main disadvantage of any of the above approaches is that the set of rules obtained is not normalized disjunct. This implies for example that many specific rules are covered by multiple general rules. So, there may be a more economical way to cover the specific rules. It also implies that the classification will be more complicated in order to deal with multiple coverage. This problem will be the subject of the next section.

5.4.3 Disjoint Rule Induction

The goal of this learning process is to obtain a rule base covering the complete instance space with a minimum number of disjoint (non-overlapping) rules and a maximum value for the overall J-measure: $J(\mathcal{R}) = \sum J(R_i)$. The rules should be normalized disjunct, or disjunct for short.

To find the optimal rule base, all possible combinations of rules that cover the instance space are to be evaluated. Since this is hardly ever possible, a recursive approximation can be used.

- step 0. Start with an empty rule base,
- step 1. generate all possible hypotheses that are disjoint with the rule base and that cover some specific rules,
- step 2. add to the rule base the hypothesis having maximum J-measure value,
- step 3. repeat step1 and step2 until all specific rules are covered.

The major difference between this approach and the non-disjoint approach is that this approach tries to optimize the total rule base whereas the non-disjoint approach only optimizes each specific rule by generalizing it to the most informative rule.

Relevance of a rule base

The problem with the recursive approach of disjoint rule induction is that often at the end of the recursion process rather poor rules are generated. These rules hardly contribute to the overall J-measure but are necessary to obtain a complete coverage of the instance space. Especially when the data set is noisy these rules are not interesting and represent the noise rather than an useful concept description. In such cases an alternative approach might be to replace some of the poor rules by a default rule, which can be interpreted as an "else"-rule with respect to the rules in the rule base. The question then becomes: which rules should be replaced, or in other words: what are relevant rules?

The problem of relevancy can be approached by defining a threshold for the J-measure, but this threshold will most likely depend on the data at hand and has to be set by a user or expert. Therefore, we suggest a different approach. The overall J-measure is also known in information theory as the "mutual information"; $I(\mathcal{R};\mathcal{C})$. The mutual information for a rule base $\mathcal{H}$, associating regions $\mathcal{R}$ to the set of classes $\mathcal{C}$, can also be expressed as:

$$I(\mathcal{R};\mathcal{C}) = H(\mathcal{C}) - H(\mathcal{C}|\mathcal{R}) \tag{5.17}$$

There is a theoretical bound to the maximum of the overall J-measure, namely $H(\mathcal{C})$. This maximum value can be obtained in many ways, of which the extreme cases are:

- each class is perfectly described with a single rule,
- each class is perfectly described by as many rules as there are mutually different examples of this class (most specific rules).

The first case is what is ultimately aimed at in the learning process, the second case resembles a one-nearest-neighbor classifier using some Voronoi partition of the instance space; for an overview of Voronoi diagrams we refer to [104]. Due to limitations to the representational flexibility of rules, arising from both the fuzzy sets used and the hypotheses generated, neither of the above partitions will be found exactly. From statistics it is well known that estimators of the probability of some event (region or symbol) can certainly be estimated accurately if many examples are at hand. If a rule covers only a few examples, the confidence in the conditional probability is low. We assume that relevance depends on this confidence. Since confidence depends on the information used for estimation, relevance depends on the amount of information used in a rule, or more precisely, the information covered in the conditional part of the rule.

To proceed, we note that the coverage of a rule is defined by the conditional part of the rule. The combination of all the conditional parts forms a partition of the instance space. The relevance of a rule base will be based solely on the partitioning it forms of the instance space, irrespective of the associated classes. The average information necessary to represent a partition $\mathcal{R}$, on which the rules are based, is:

$$H(\mathcal{R}) = \sum_i -P(R_i)log(P(R_i)) \tag{5.18}$$

The amount of information of the (observed) instance space that $\mathcal{R}$ maximally represents, can be measured through the most specific partition formed on the basis of the instances. This virtual partition will be denoted by $\mathcal{S}$. Since the partition $\mathcal{S}$ consist of regions, a region for each mutually exclusive instance observed, the maximum number of rules that can be formed in this way is exactly equal to the number of data. Each region in the virtual partition $\mathcal{S}$ is nearly completely irrelevant since its probability estimate is based on a single instance. Because a region of $\mathcal{R}$ covers some regions of $\mathcal{S}$, its probability estimate

is based on multiple instances, thus the relevancy of $\mathcal{R}$ is larger. The amount of information that $\mathcal{R}$ on average represents through $\mathcal{S}$ can be measured by the conditional information $H(\mathcal{S}|\mathcal{R})$. If the partitions $\mathcal{R}$ and $\mathcal{S}$ are normalized disjunct then:

$$H(\mathcal{S}|\mathcal{R}) = H(\mathcal{S}) - H(\mathcal{R}) \tag{5.19}$$

Now the larger this quantity, the more relevant the partition $\mathcal{R}$ is on average. Normalized by $\mathcal{S}$, this quantity is defined as the relevancy of $\mathcal{R}$ with respect to $\mathcal{S}$:

$$rel(\mathcal{R}) = 1 - \frac{H(\mathcal{R})}{H(\mathcal{S})} \tag{5.20}$$

Hence, relevancy is a mapping to $[0, 1]$ and can be considered as a certainty measure. We now simply decide that a rule base is *relevant* if the relevance of its partition is larger than or equal to 0.5. Put in words this means that "A rule base $\mathcal{H}$ is relevant if the average information necessary to represent its partition $\mathcal{R}$ is at least smaller than or equal to the average information that the partition $\mathcal{R}$ represents". Note that one of the partitions is the "else"-rule, which is necessary to complete the coverage.

For a set of n instances and an induced partition of equiprobable regions it is easily derived that the maximum number of relevant partitions is $\sqrt{n}$. This is strikingly similar to the rule of thumb used in forming histograms for estimating densities.

We can now modify step 3 in the recursive process such that we repeat step 1 and step 2 as long as the rule base is *relevant*. We will refer to this approach as *restricted disjoint rule induction*. Note that the concept of relevancy can also be used as a criterion for comparing different rule bases. Normally rule bases are compared on the basis of their classification accuracy and the number of rules, reflecting the data fit and mental fit, respectively. However, we think that the relevancy as defined here is a much better reflection of the mental fit than the absolute number of rules.

5.5 Classification and Explanation

In principle both the specific rules and the general rules can be used for classification and explanation, but the general rules are more suitable for this task for reasons specified. In case of a normalized-disjunct rule base the classification takes place as follows. Given a new instance $\mathbf{x}_o$:

$$P(C_j|\mathbf{x}_o) = \sum_h \mu_{R_h}(\mathbf{x}_o) P(C_j|R_h) \tag{5.21}$$

as derived in section 4.7 of the previous chapter. Note that all conditional probabilities of a rule are used for deriving the a posteriori probability. The final classification is then the class C_k having the maximum probability. The

explanation for this class given by the system is the condition of the rule which contributes most to the final decision:

$$\{R_h | \max_h \mu_{R_h}(\mathbf{x}_o) P(C_k|R_h)\} \tag{5.22}$$

Hence, the explanation is based on three elements which are important for a good explanation: the class derived, the membership and the conditional probability.

Non-disjoint rules

If the rule base is not normalized disjunct, the classification is somewhat more complicated due to

- intersecting rules,
- mutual coverage.

Suppose that a rule H_v^k intersects the rule H_u^k of the same rule base, then these rules are not normalized disjunct, since the region R_{uv} is covered twice. If a rule H_{uv}^k would be present in the rule base, then things would even be worse, because the same region would be covered three times. If with such a rule base the usual classification method is performed then there are two complications. First, the sum of the memberships will not be equal to one and the a posteriori will not be normalized. Second, the conditional probabilities $P(C_j|R_{uv})$ are present both in H_u^j, H_v^j and obviously in H_{uv}^j. These consequences can be corrected by making the rules disjoint and by recalculating the probabilities. That is, the region R_{uv} has to be excluded from the rules H_u^k and H_v^k, both in terms of memberships as well in terms of conditionals. For the memberships we recall that:

$$\mu_{R_u}(\mathbf{x}_o) = \sum_{v'} \mu_{R_{uv'}}(\mathbf{x}_o) \tag{5.23}$$

and for the conditional probabilities we recall that:

$$P(C_j|R_u)P(R_u) = \sum_{v'} P(C_j|R_{uv'})P(R_{uv'}) \tag{5.24}$$

and

$$P(R_u) = \sum_{v'} P(R_{uv'}) \tag{5.25}$$

So in our example, the membership can be corrected by simply subtracting the membership of the rule covered:

$$\mu_{R'_u}(\mathbf{x}_o) = \mu_{R_u}(\mathbf{x}_o) - \mu_{R_{uv}}(\mathbf{x}) \tag{5.26}$$

and the conditional probability is corrected by:

$$P(C_j|R'_u)P(R'_u) = \frac{\sum_{v'} P(C_j|R_{uv'})P(R_{uv'}) - P(C_j|R_{uv})P(R_{uv})}{P(R_u) - P(R_{uv})} \tag{5.27}$$

The same can be done for the rule H_v^k so that the rule $H_u^k v$ need not to be corrected. The final classification can then be obtained in the usual way if the corrected memberships and corrected conditionals are used. Note that if we only had two intersection rules H_v^k and H_u^k the procedure would be the same, but the rule H_{uv}^k should have been constructed first from the information layer.

Although the procedure as outlined above is correct from a theoretical point of view, it is also very cumbersome and complicated. Quite often a lot of multiple intersections occur and it requires some sophisticated accountancy to keep track of all regions. Yet strategies like these are not uncommon, and are often referred to as "backtracking" as was mentioned in section 2.2.4 of Chapter 2. The complicated accountancy is not the only disadvantage, a classification with different rules than actually present in the rule base is also problematic from an explanation point of view. Further, we notice that, in the case of many intersecting rules and mutual coverages, backtracking essentially results in a classification that could have been obtained immediately from the specific rules (information layer). For these reasons we use a different approach.

First of all we notice that it is often not necessary to correct the conditionals. This can be defended by noticing that the intersection is always a relatively small subset. In case of our previous example, if $P(R_{uv})$ is much smaller than $P(R_u)$ then the corrected conditional is nearly equal to the original. Further we reason that the intersections do not occur by accident, each is the most informative rule for the underlying specific rules covered. Therefore instead of deploying a regressive reasoning scheme (backtracking), we prefer a progressive reasoning scheme where all rules are considered as being equally important implications. What is then needed is a normalization of the memberships. All being equally important then an obvious way to normalize the a posteriori probabilities is by:

$$P(C_j|\mathbf{x}_o) = \frac{\sum_h \mu_{R_h}(\mathbf{x}_o) P(C_j|R_h)}{\sum_h \mu_{R_h}(\mathbf{x}_o)} \tag{5.28}$$

However, it should be noticed that in regions of the feature space where several rules having equal conclusions intersect, the class decision will be biased by the number of intersecting rules having the same most likely decision. The explanation is again found by maximizing over the contributions to the final class decision.

5.6 Coping with High Dimensionality

The number of possible specific rules is generally rather large in high-dimensional feature spaces. In order to cope with this large number of rules only the nearest-neighbor rule is used to cover an example. This strategy was also successfully applied in Chapter 3. The nearest-neighbor rule for an example is the rule for which the example has the highest membership. This specific rule can be easily found by combining the fuzzy sets for which the example has highest membership per feature. In this way the total number of specific rules, being r, is always smaller than or equal to the number of examples. Having established which r

specific rules are to be used, we can estimate the conditional probabilities for the specific rules by using a forced normalization over all the present regions R_h:

$$P(C_j|R_h) = \frac{\sum_{i=1}^{n} \frac{\mu_{R_h}(\mathbf{x}_i)}{\sum_h \mu_{R_h}(\mathbf{x}_i)} \mu_{C_j}(y_i)}{\sum_{i=1}^{n} \frac{\mu_{R_h}(\mathbf{x}_i)}{\sum_h \mu_{R_h}(\mathbf{x}_i)}} \tag{5.29}$$

The induction of the general rules can follow each of the schemes proposed, if the available specific rules are used. However, to insure that the complete instance space is covered by the induced knowledge layer, a default rule can be added. Such a default rule is only used when all other rules have zero membership for a new instance. The default rule usually has conditional probabilities equal to the a priori class probabilities.

The conditionals for the general rules can be obtained by taking the union over all specific rules as pointed out in section 5.3. However, this is usually not equivalent to using 5.7 for estimating the conditionals for the general rules, due to the forced normalization in 5.29. Note that for very large data sets 5.7 and 5.29 are equivalent. The reason for this forced normalization is that the specific rules do not form a complete cover of the instance space, the general rules will in general also not completely cover the instance space. The classification is obtained by using 5.28, see also the previous chapter, section 4.7, where we discussed classification using reference frames which do not exactly sum out to one (i.e. are not normalized disjunct).

5.7 Experimental Results

For experimental results we used an implementation called FILER (version 5.0). It has been developed as a tool to experimentally verify and evaluate the framework of fuzzy probabilistic induction. Many data sets have been used for rule induction and many experiments have been set up. One of the most interesting of these experiments was a comparison on the basis of 5 data sets using some of the results of the Statlog project [96]. The Statlog project, which ended in 1994, compared 24 algorithms on several types of data sets on basis of their classification performance, number of rules (storage), easiness of use and computational time. The algorithms used represented state-of-the-art techniques in statistics, neural networks and machine learning for pattern recognition under supervised circumstances.

5.7.1 Experimental Setup

We used three methods for generalization:

- disjoint rule induction,
- restricted disjoint rule induction,

- non-disjoint rule induction with representative criterion.

For each method the classification error and the number of rules were calculated for comparison. To estimate the classification error, we used the same method as in the Statlog project; 9- or 10-fold cross-validation. Since an expert was not available, we used fuzzy sets obtained from K-means clustering as described in section 5.2 to obtain the reference frame. For each feature we used the same number of clusters. The experiment was repeated for different numbers of clusters. Here, we restricted the number of clusters since it is generally acknowledged that much more than seven qualifications severely complicates an explanation.

5.7.2 Data Sets

Here we give a brief description of five data sets from the Statlog project, which are also available to the public.

Heart

This data set originated from the Cleveland Clinic Foundation. It has been collected in order to predict a heart disease. It contains 270 examples, divided in two classes: yes/no disease. The dimensionality is 13, hence a total of 13 features. 7 features are continuous, 3 are binary and 3 are categorical. For this data set the classification error was estimated using 9-fold cross-validation. Best result of the 9 rule induction algorithms evaluated in the Statlog project was an error-rate of 0.44 (44%) obtained by a decision tree consisting of 51 nodes. The smallest rule base of all examined rule induction algorithms had 21 rules and an error rate of 0.844.

Australian Credit

This data set has been collected to learn what potentially good or bad credit-card holders are. The classification rules can be used to assign a credit card or not. It consists of 690 examples divided in two classes (307/383). It has 14 features, 7 continuous and 7 categorical. For this data set the classification error was estimated using 10-fold cross-validation. Best result of the 9 rule induction algorithms evaluated in the Statlog project is an error-rate of 0.131 obtained by of a decision tree consisting of 128 nodes. The smallest rule base of all examined rule induction algorithms had 28 rules and an error rate of 0.181.

Diabetes

This data set has been donated by Vincent Sigillito from the John-Hopkins University. It has been collected among the Pima Indians Tribe for the diagnosis of diabetes among the Pima Indians. It contains 768 examples (500/268), in two classes. The majority class is the diabetes-negative decision. It has 8 features, all

being continuous. For this data set the classification error was estimated using 9-fold cross-validation. Best result of the 9 rule induction algorithms evaluated in the Statlog project is an error-rate of 0.245 by a rule base consisting of 60 rules. This was also the smallest rule base of all examined rule induction algorithms.

Segment

This data set has been donated by the Vision Group of the University of Massachusetts. It has been collected in order to learn image segmentation. By hand, seven pictures, taken outside, have been segmented to label 3x3 pixels as brick, air, window, grass, bush, pavement and cement. It contains 2310 examples in 19 dimensions characterizing color and shape by using image analysis techniques. For this data set the classification error was estimated using 10-fold cross-validation. Best result of the 9 rule induction algorithms evaluated in the Statlog project is an error-rate of 0.031 obtained by a decision tree consisting of 7830 nodes. The smallest tree obtained of all examined rule induction algorithms had 57 nodes and an error rate of 0.040.

Vehicle

This data set comes from the Turing Institute and concerns the recognition of cars by 2D images. It contains 846 examples divided in the classes: Chevrolet van, Saab 9000, Opel Manta 400 and the double decker bus (Londoner). It contains 18 features. Each feature is obtained from standard image analysis techniques. For this data set the classification error was estimated using 9-fold cross-validation. Best result of the 9 rule induction algorithms evaluated in the Statlog project is an error-rate of 0.235 obtained by a decision tree consisting of 158 nodes. The smallest tree obtained of all examined rule induction algorithms had 71 nodes and an error rate of 0.271.

5.7.3 Results

In the Statlog project the cross-validation was performed only once for each algorithm with exactly the same subsets. To make a fair comparison, we have repeated the cross-validation 5 times and took the average of all trials. For the number of rules, we calculated the average as well. Further, we calculated the standard deviation for both averages.

The results are summarized in tables (standard deviations in brackets) . For comparisons with the Statlog experiments, the relative rank of FILER among the 10 rule-based algorithms (including itself) is indicated in the column "rank". Here we have indicated the rank in terms of accuracy as well as the rank in terms of the number of rules. For the latter rank, the number of nodes were treated as the number of rules (although a decision tree having n nodes usually represents much more than n rules!).

Specific rules

In Table 5.1 the results obtained from the specific rules are summarized. Surprisingly, the specific rules can obtain very accurate results. Only two times it is not capable of obtaining the first position in accuracy, for the credit data set and the vehicle data set. This indicates that the specific rules are very well capable of classifying new (unobserved) instances. This is probably due to the use of the Gaussian fuzzy sets. Clearly, the main drawback of the specific rules is the number of rules and their complexity (due to the fact that they tie all features). This drawback makes the classification somewhat slow and more complex to explain.

When compared to other techniques evaluated in the Statlog project, in terms of the error-rate, the specific rules obtain rank 1 for the Heart and Segmentation data set. Remarkably, it also outperformed the back-propagation and radial-basis-function neural network techniques on all data sets except for the Vehicle data set. On the Diabetes and Australian Credit data set, discriminant-function approaches outperformed the specific rules with maximally 0.02 lower error-rate (overall 25 algorithms: best result on Diabetes: 0.223, best result on Australian Credit 0.131, best result on vehicle 0.151).

Table 5.1: *Results FILER: specific rule base.*

Data	# sets	# rules	X-fold	rank	
				X-fold	# rules
Heart	2	194 (0)	0.383 (0.01)	1	9
	3	222 (0)	0.44 (0.02)	2	10
	4	237 (0)	0.40 (0.02)	1	10
Credit	2	346 (0)	0.151 (0.004)	4	7
	3	488 (0)	0.185 (0.008)	9	8
	4	548 (0)	0.176 (0.009)	7	9
Diabetes	2	110 (0.08)	0.240 (0.006)	1	5
	3	397 (0)	0.249 (0.006)	2	9
	4	489 (0)	0.267 (0.006)	4	9
Segment	5	1041 (0)	0.050 (0.001)	8	8
	6	1250 (0)	0.036 (0.001)	4	8
	7	1429 (0)	0.033 (0.001)	3	8
	8	1481 (0)	0.026 (0.001)	1	8
Vehicle	3	496 (0)	0.3466 (0.006)	10	6
	4	655 (0)	0.320 (0.003)	9	6
	5	711 (0)	0.326 (0.007)	10	6
	6	739 (0)	0.298 (0.003)	7	6

Disjoint generalization

The results are outlined in Table 5.2. FILER performs on the first three data sets excellent in this mode, both in terms of the number of general rules as well in terms of the error obtained. Note that on the first three data sets, the disjoint generalization also leads to somewhat more accurate results. Although it only scores third on the Credit data set, it should be noted that the second position on this data set is the ITrule algorithm [129] with an error of 0.137 with 124 rules. There are also two data sets where it really does not perform well in terms of accuracy, the Segment data set and the Vehicle data set (both being image classification problems). Since the specific rules were capable of obtaining high accuracy on the Segment data set, it seems that the disjoint generalization is not suitable for this data set.

Table 5.2: *Results FILER: disjoint general rule base.*

Data	# sets	# rules	X-fold	rank	
				X-fold	# rules
Heart	2	45 (2)	0.40 (0.04)	1	2
	3	37 (1)	0.43 (0.02)	1	2
	4	41 (1)	0.37 (0.02)	1	2
Credit	2	41 (2)	0.142 (0.007)	3	2
	3	61 (2)	0.150 (0.007)	4	2
	4	57 (2)	0.149 (0.004)	4	2
Diabetes	2	36 (1)	0.237 (0.007)	1	1
	3	94 (2)	0.248 (0.004)	2	5
	4	107 (6)	0.252 (0.009)	3	5
Segment	5	121 (2)	0.124 (0.008)	9	2
	6	108 (1)	0.099 (0.003)	9	2
	7	122 (8)	0.088 (0.008)	9	2
	8	167 (1)	0.106 (0.004)	9	3
Vehicle	3	86 (3)	0.389 (0.007)	10	2
	4	125 (4)	0.337 (0.01)	10	2
	5	131 (5)	0.342 (0.009)	10	2
	6	146 (4)	0.330 (0.007)	10	2

Using rule base relevancy

The results are outlined in Table 5.3. The results with restricted disjoint rule induction are hardly different than obtained from the standard disjoint generalization. Only on the vehicle data set it is significantly better both in terms of accuracy and in terms of the number of rules. This may indicate that the vehicle data set is rather noisy.

Table 5.3: *Results FILER: restricted disjoint rule base.*

Data	# sets	# rules	X-fold	rank	
				X-fold	# rules
Heart	2	41 (2)	0.403 (0.03)	1	2
	3	38 (2)	0.44 (0.03)	1	2
	4	40 (1)	0.38 (0.02)	1	2
Credit	2	40 (1)	0.141 (0.003)	3	2
	3	61 (2)	0.158 (0.005)	6	2
	4	57 (1)	0.139 (0.004)	3	2
Diabetes	2	36 (1)	0.241 (0.004)	1	1
	3	94 (4)	0.247 (0.008)	2	5
	4	106 (3)	0.256 (0.007)	4	5
Segment	5	117 (1)	0.119 (0.002)	9	2
	6	108 (1)	0.097 (0.001)	9	2
	7	121 (2)	0.089 (0.003)	9	2
	8	168 (2)	0.104 (0.003)	9	3
Vehicle	3	84 (2)	0.388 (0.009)	10	2
	4	80 (5)	0.321 (0.009)	9	2
	5	88 (2)	0.337 (0.009)	10	2
	6	106 (3)	0.329 (0.007)	10	2

Non-disjoint generalization

The results are outlined in Table 5.4. As indicated by the rank in terms of the number of rules, this type of generalization certainly leads to very few rules. For the first two data sets, Heart and Credit, this seems a suitable way of generalization since the error rate remains equal or even better.

Like disjoint generalization, also this type of generalization is not suitable for the segment data set. Since the specific rules are capable of obtaining high accuracy on this data set, this suggest that a third generalization method should be used. We think that for this data set, region-growing may be very suitable.

Analysis of the Vehicle data set

For all data sets a suitable rule base could be found that obtained excellent classification results, except for the Vehicle data set. To investigate the vehicle data set in somewhat more detail, we used the non-uniform double-kernel estimator. The obtained result, see Table 5.5, shows that much smaller error rates are possible on the vehicle data set with a very small number of kernels. The optimum in the table lies at an error-rate of 0.147 with a number of kernels equal to 51% of the number of data in the training (which comes down to approximately 388 kernels). However, with only 15 kernels still an error-rate of 0.156 can be obtained. Note that the best result on this data set reported in

Table 5.4: *Results FILER: non-disjoint using general rule base.*

Data	# sets	# rules	X-fold	rank	
				X-fold	# rules
Heart	2	36 (2)	0.40 (0.03)	1	1
	3	27 (2)	0.44 (0.03)	1	1
	4	24 (1)	0.38 (0.02)	1	1
Credit	2	28 (1)	0.139 (0.005)	3	1
	3	31 (1)	0.138 (0.003)	3	2
	4	29 (1)	0.137 (0.004)	2	2
Diabetes	2	22 (1)	0.299 (0.007)	10	1
	3	51 (1)	0.317 (0.005)	10	1
	4	38 (1)	0.304 (0.005)	10	1
Segment	5	72 (1)	0.223 (0.004)	9	1
	6	92 (1)	0.124 (0.003)	9	1
	7	87 (1)	0.121 (0.004)	9	1
	8	76 (1)	0.094 (0.001)	9	1
Vehicle	3	83 (2)	0.500 (0.02)	10	2
	4	92 (1)	0.419 (0.008)	10	2
	5	103 (1)	0.372 (0.008)	10	2
	6	123 (1)	0.381 (0.008)	10	2

Table 5.5: *Results of the non-uniform double-kernel estimator for the Vehicle data set.*

data set	h	maximum loss%	Reduction Ratio	X-fold error
Vehicle	1.22	0	0.98 (0.01)	0.155 (0.006)
		5	0.91 (0.01)	0.155 (0.007)
		10	0.87 (0.01)	0.153 (0.009)
		20	0.77 (0.01)	0.158 (0.006)
		30	0.65 (0.01)	0.151 (0.005)
		40	0.60 (0.02)	0.153 (0.004)
		50	0.51 (0.01)	0.147 (0.008)
		75	0.27 (0.01)	0.154 (0.006)
		99	0.02 (0.01)	0.156 (0.002)

the Statlog project is an error-rate of 0.151. Since the fuzzy probabilistic rule induction can be seen as a special case of the double-kernel estimator, it may seem strange that such a large difference can occur. The fundamental difference, however, is that the class-conditional covariances are not taken into account in the rule induction method.

5.8 Conclusion

We have discussed a synthesis of the information-theoretic approach to rule induction and the fuzzy probabilistic framework on the basis of the DIK paradigm. We have shown how the information layer can be derived from the data by using a reference frame. The reference frame is either an a priori discretization or a discretization obtained by clustering. Highly accurate decisions can be made and explained in full detail with the specific rules, if the reference frame is a discretization on the basis of fuzzy sets having a Gaussian shape. We have used two methods for generalization: disjoint and non-disjoint rule generalization to obtain knowledge represented as general rules. Both methods are based on a very simple scheme for hypothesis generation: dropping the condition. The criterion for rule selection is based on the J-information measure, in which the fuzzy probabilities were used. Comparative experiments on the basis of publicly available data sets, earlier used in the Statlog project, show that fuzzy probabilistic rule induction can lead to excellent classification results with only a few extremely simple rules. The experiments also show that there is not a single type of generalization suitable for all problems, like clustering techniques, it depends on the type of problem at hand which type of generalization is most suitable. Unfortunately, it cannot be said beforehand what type of generalization should be used. The experiments indicate that there may be a need for a third type of generalization: region-growing, for which a more sophisticated hypothesis generation is necessary than is used here. For this type of generalization a hierarchy in fuzzy qualifications of a feature can be very useful.

The problem of data fit versus mental fit[2] for a large class of problems in continuous domains has been successfully addressed with the fuzzy probabilistic rule induction technique. However, the demonstrated algorithm is mainly suitable for problems where the class-conditional covariances provide little information or are difficult to estimate. Such problems frequently occur in image and speech analysis, where features often have little meaning. Yet, results with the double-kernel estimator demonstrate that with only a few kernels highly accurate classifications can be obtained for such problems. Hence, in principle, it is possible to incorporate the class-conditional covariance in the fuzzy probabilistic rule induction, but the translation of the kernels used in the double-kernel estimator to expressive rule conditions is an open problem. Of course, whitening the data or principal components analysis removes the covariances and makes the problem suitable for rule induction, but then the new features are mixtures of the original features. These mixtures of features are often not useful for decision support, but we think that unequal weighting of the features on the basis of the class-conditional covariances can help to solve this problem. However, it may also turn out that for problems such as aural and visual pattern recognition, rule-based methods are not appropriate. Fortunately, for such problems we often do not expect an explanation.

[2]Here, mental fit is somewhat reduced to: "a low number of simple rules". Although this is a good quantitative measure, it is also necessary to involve the expert in the evaluation of the mental fit.

Chapter 6

The Intelligent Anesthesia Monitor

In 1994 the Intelligent anesthesia monitor project started at the Delft University of Technology. The goal of this project was to increase the safety of a patient undergoing an operation by improving the decision support for the anesthetist. During an operation, it is the primary task of the anesthetist to suppress the patient's awareness of pain and of the operation itself, and to ensure sufficient functioning of the patient's vital physiological processes. Hence, it is not only his task to put the patient asleep, but also to monitor the patient's health and to intervene as necessary to keep the patient as healthy as possible while the operation proceeds.

Improving the anesthesia process has been a subject of research since the first reported anesthesia in 1846. These improvements have involved three aspects of the anesthesia process. First, knowledge of human physiology has increased considerably over the last century. Second, improved drugs and new anesthesia equipment enable the anesthetist to intervene quite effectively in the state of the patient. Third, new monitoring devices have become available which allow a better assessment of the state of the patient. As a result of these improvements, anesthesia safety has increased considerably, but further improvements can and should be made. Recent studies on anesthesia incident analysis show that as much as 70% of the incidents reported nowadays involve human error [24].

One of the areas where anesthesia can be improved is patient monitoring. In the modern operating theater many physiological signals and parameters (sometimes as many as 30 parameters) are measured and displayed on the anesthesia monitor. In itself this is an improvement, but there is a great risk that the anesthetist becomes overloaded in the amount of data that he has to process in order to asses the patient's health. Hence, assuming that the anesthetist's data processing capacity is limited and not infallible, support for this monitoring task may be an important step towards reducing human errors in anesthesia.

Data processing by commercial monitors has proven to be inadequate for

reliably alarming the anesthetist [68], especially because so many false alarms are given [10]. It is not uncommon that the alarms go off constantly, where at best only 10% of all generated alarms is useful. Therefore the goal of the IAM project is to improve the decision support to the anesthetist. As such, a monitor had to be designed that could process the available data and present the relevant information to the anesthetist. Instead of modeling the patient and explicating medical knowledge, like in [51, 56], the decision-making process of the anesthetist was modeled in the IAM project.

In this chapter we will give first an outline of the project and we will then focus on using the fuzzy probabilistic rule induction algorithm for the Intelligent Anesthesia Monitor project. For a detailed description of other aspects of the Intelligent Anesthesia Monitor we refer to the thesis of de Graaf [29] (information management) and to the thesis of Vullings [137] (waveform validation).

6.1 Monitor Design

In order to arrive at the specifications for the IAM, the anesthesia process has been analyzed. In this analysis we focused on the decision-making process of the anesthetist. This analysis resulted in requirements for the monitor.

Anesthesia process

The decision-making process of the anesthetist can be split in four separate levels.

- signal validation,
- trigger identification,
- diagnosis,
- treatment.

The first level is the perceptual level, which involves signal waveform interpretation for diagnostic and validation purposes. Decisions on this level are concerned with whether the waveform observed is indeed a valid measurement or that it is disturbed due to patient movement and alike. The second level is the pattern recognition level, in which parameter patterns are detected that trigger further analysis. For example, it may be decided that the current heart rate is increased and the blood pressure is decreased, and that this needs further diagnosis. The third level is the diagnosis level, in which many sources of information to diagnose the current situation are integrated. The sources of information are many; the trigger caused by the parameter patterns, the surgeon providing information on the surgery, anesthesia knowledge concerning incidents etc.. Decisions on this level are concerned with identifying the cause for the observed patterns. For example, it may be diagnosed that the decreased blood pressure and the increased heart rate are caused by blood loss. The fourth level is concerned

with deciding what the proper treatment is. For example, the bleeding may be stopped and extra blood should be given to the patient to restore his condition.

Monitor requirements

From the data-processing point of view, the monitor should be especially supportive in the first two stages. If the anesthetist can be supported here, then he can concentrate on the diagnosis and the treatment. To support the anesthetist on the first two levels, monitoring is required. Signals should be measured and displayed, preferably accompanied by some trigger. In commercial monitors this function is in principle available. However, some improvements are necessary in current monitoring.

- reduction of false alarms,
- increased reliability of default alarms,
- improved presentation of information.

The reduction of false alarms is the first task that is set for the IAM monitor. In commercial monitors the alarms are often silenced by the anesthetist because the alarms go off so often that it becomes annoying to most anesthetists. The second task involves the reliability of the default alarms. The default alarms in current monitors are often threshold alarms on single parameters. Hence, warning on the basis of multiple parameters is not possible, nor do the alarms go off on the basis of trends. Hence, some improvement in reliability of the default alarms can be expected if multiple parameters and trends are incorporated. Finally, there is a need to have the information presented in some orderly way. One can easily imagine a situation in which a monitor simply produces beeps and flashes, and where the anesthetist is wondering why while the patient is not getting any attention. In that sense, a decision-support system can be counterproductive since it simply is another source of data. What is very important in decision support systems is, therefore, that the system should present causes for its support which can be understood by the anesthetist.

IAM

To achieve the improvements we designed a monitor consisting of three stages, see Figure 6.1. Physiological signals and parameters measured from available sources enter the monitor as data. The first stage in the IAM is to validate the signals. Signals may be disturbed due to electro-magnetic interference by other equipment, detached sensors, movement of the patient, etc.. Since nearly all parameters are derived from (nearly) continuous waveform signals such as: ECG, blood pressure, capnogram (CO2 production), pulseoxy (SPO2) and such alike, it is reasoned that validating these waveforms is sufficient. If the waveform is valid, then the derived parameters are assumed to be valid as well. If the waveforms are not valid, then all other data processing is suppressed. Suppressing all other processing prevents false alarms. The valid parameters then enter the

analysis stage. In this stage, patterns are detected that require the attention of the anesthetist. The patterns, as detected in the valid parameters, are then sent to the strategy stage where it is decided what information is presented and how it is presented to the anesthetist. The minimum functionality of the strategy stage is that of a user interface. Note, however, that the functionality of the strategy stage can be extended to diagnosis and treatment if knowledge concerning these decision levels becomes available through scientific progress. For example, some patterns may be related to specific incidents, which can be presented to the anesthetist as a suggestion. An on-line library of relevant anesthesia knowledge related to the detected patterns can also be activated. In principle the IAM can support the anesthetist on each level of decision making, however, its feasibility depends on advances in the area of medicine, risk analysis, signal analysis, knowledge-based systems and pattern recognition. The validation stage is described in [137] and the strategy stage is described in [29]. The analysis stage will be described in this chapter

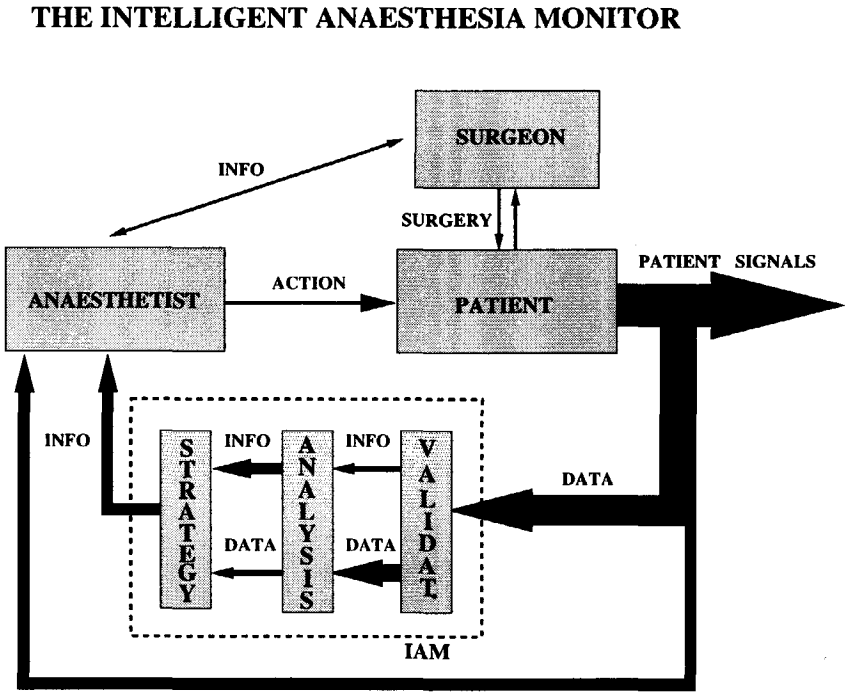


Figure 6.1: *The three stage design of the Intelligent Anesthesia Monitor.*

6.2 Analysis Stage

In the analysis stage patterns must be identified in the physiological parameters which require attention of the anesthetist. The patterns identified should be described such that further reasoning and diagnosis is possible. The reasoning may take place in the strategy stage or in the mind of the anesthetist. Simply put, this means that the analysis stage should identify alarms and be able to explain them. An alarm is a pattern in the physiological parameters which requires attention of the anesthetist. Essentially there is only one basic problem: What are the patterns that require the attention of an anesthetist?. When asked to anesthetists in the field, the general consensus is that, no matter what the patterns are, they should be warned as little as possible, only if necessary. Hence, the warnings should at least be reliable. However, this does not solve the alarm problem: when should an alarm be given? There are at least two viewpoints from which this problem can be tackled.

6.2.1 Alarms vs. Triggers

One way to solve the alarm problem is to regard those patterns as alarms that lead to seriously dangerous situations for the patient. Such situations can be observed in incidents, where something went very wrong and the anesthetist was not alarmed. Several scenarios of such incidents are known and are generally quite complex in the sense that often a combination of physiological causes, anesthetist interventions and other factors are involved [49]. Although such incidents are rare and often more is involved than the physiological causes only, it is possible to collect and analyze incidents to find the circumstances under which they occurred and use this information to alarm the anesthetist. An alarm system like this would be very reliable since it sounds an alarm only when something is really wrong, and otherwise it would not. The problem with such a system is, however, that as soon as it alarms, it may be already too late. Therefore, it would be necessary to investigate many of such incidents and to find some factors on the basis of which such incidents can be predicted. Since the incidents are not available on a large scale this is not a practical approach. Even if they were available, it might turn out that predicting incidents is as difficult as predicting the stock exchange.

A second way to solve the alarm problem is by regarding patterns that trigger the anesthetist. A trigger which is ignored may lead to an alarm, but not necessarily. Further, a trigger is often necessary when a *change* in the physiological parameters is detected. However, the physiological parameters are almost never completely stable, there is always some noise or fluctuations present depending on the patient and the type of operation. Therefore, the problem becomes to determine which changes are *relevant* and which are not. This view is shared by Ballast, who he called this type of alarm "an early warning system" [10].

6.2.2 Learning Triggers

there are generally two approaches to determine relevant changes in the physiological parameters. One is to obtain the knowledge for determining such relevant changes from the experts: expert interviews. From the field of Expert Systems it is generally known that such elicitations of expert knowledge are very problematic because the expert himself can usually not be so specific that a workable system is produced. This is often referred to as the knowledge-acquisition bottleneck in expert systems, as we mentioned in the introduction of this thesis. A second approach is to use some sort of statistics in order to characterize the concept of relevant change and of stability on the basis of observed parameters. This method in general is known as supervised learning, and we use it for the analysis stage. For such an approach it is necessary that examples are available that reflect the concept of "stability" and the concept of "relevant change". In general these examples are not available but can be derived from the interventions of the anesthetist. The anesthetist is regarded as a simple stimulus-response system, see Figure 6.2. The response is the intervention and the stimulus is the pattern observed in the parameters. What remains to be done is to learn the relations between the responses and the proper stimulus. We reason that every intervention is in principle initiated by a trigger observed in the physiological parameters. Hence, we can deduce that roughly every intervention defines a "relevant change", whereas from non-interventions the "stability" concept is defined. Since in general it is not known *why* the intervention took place, any of the parameter changes or a combination may have been the trigger responsible for the intervention.

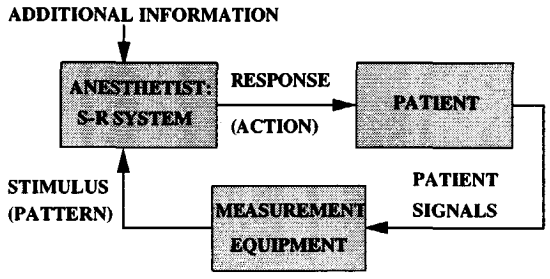


Figure 6.2: *Anesthetist modeled as a simple Stimulus-Response system. The stimulus is the pattern in the physiological parameters that triggers the anesthetist.*

Characterization of the learning problem

On the basis of the above stimulus-response model we arrive at a two-class learning problem, where the feature space is formed by the physiological parameters. These parameters are for example systolic, mean and diastolic blood pressure, end-tidal CO2, oxygen saturation, heart rate etc.. In principle the

feature space consists of all physiological parameters that were measured while the intervention took place. Since we want to detect *changes* in the parameters, each parameter should be characterized in time. This can be done by several features such as current value, short-term trend, long-term trend, moving average etc.. Other characterizations of a parameter in time (a signal), such as by Fourier transforms or wavelets, may also be possible but we must bear in mind that the goal is to explain a trigger such that further reasoning is possible.

If we have say six parameters and each parameter is characterized by five features (e.g. trends), then the total space consists of 30 dimensions. Further, we know that each intervention is most likely due to only one or to a few of the features and not because of all of the features. This means that not all features may be relevant, or not relevant for every intervention. From the learning point of view this means that the class of interventions consists of many sub-classes such as: "if mean blood pressure trend is high then intervention", but also, "if heart rate decreases a lot then intervention". On the other hand, the concept (class) of stability will be related to all parameters.

Selection of a learning method

For learning under these circumstances regular pattern recognition techniques like k-nearest-neighbor classifiers or Parzen Windows are not suitable, since they mainly focus on data fit (see also Chapter 2 and Chapter 5). As a result, they cannot answer the question "Because of which parameter pattern(s) is there an intervention?", other than by returning the observed parameter values. The same argument holds for learning techniques like neural nets. The only technique suitable for such a task is found in the area of rule induction, where through rule generalization a likely explanation for an intervention can be found. The problem with most rule induction techniques is the problem of mental fit versus data fit, as described in the beginning of this thesis. We recall that the mental fit of the rules is very important in the final decision-support system, since further reasoning by the anesthetist is necessary to obtain a diagnosis. This problem has motivated the development of a new rule induction algorithm: the FILER software, of which the principles have been described in this thesis.

6.3 Feasibility Study

In cooperation with the Academic Medical Centre (AMC) of Amsterdam, a first study on the feasibility of the proposed approach was carried out and described in detail in [126]. Here we will briefly summarize this study.

6.3.1 Goal

The main goal of this study was twofold. The first goal was to obtain fuzzy sets for the most frequently used physiological parameters from experts. The second goal was to obtain a rule base that was could recognizing patterns in the parameters that could trigger an anesthetist.

6.3.2 Fuzzy Sets

When anesthetists are asked to explain why certain drugs are administered, they often use qualifications like "the heart rate increased a lot" or "the systolic blood pressure is low". However, when asked to quantify these qualifications there was often some confusion due to the vagueness of the qualifications. Hence, the concept of fuzzy sets only seemed natural to anesthetists. By case-based interviewing we obtained the fuzzy sets for the systolic, diastolic and mean blood pressure, as well as for the end-tidal CO2, oxygen saturation and the heart rate. In total more than 30 cases were used and 28 anesthetists were involved. A case was based on the patient "on the table" in the operating theater and the anesthetist present. It appeared that different sets were used by experts for different types of patients. For example, the normal blood pressure for an older person having some heart disease appeared to be somewhat higher than that for a young and healthy person. On the basis of the interviews three patient classes were formed using the well-known standard of the American Society of Anesthesiologists, denoted as ASA [5]. The first class was formed by the standard ASA1 and ASA2 types of patients, the second class consisted of the ASA3 and ASA4 types, while the third class consisted of children. For each of these three classes the fuzzy sets were obtained. First it was asked in how many different sets the anesthetist would divide a parameter. Second, the anesthetist was asked to indicate for the patient on the table to which sets the parameter value in question belonged most by using a pair-wise comparison method. As a result several regions were obtained which could be denoted as "transition" areas between two sets. It was decided to model the fuzzy sets by triangular or trapezoidal shapes, such that the total membership equaled one (a requirement for the learning program) see also Figure 6.3. In retrospect it was concluded that a direct question like: "where is the transition between normal and high?" may be equally effective and less time consuming. Important seemed to be the patient class, the number of sets and the transitory (fuzzy) parts. The detailed shape did not seem to bother the anesthetist very much.

6.3.3 Rule Induction Method

For the induction of the rules the FILER algorithm was used. The type of induction used was non-disjoint rule induction, which usually leads to the most simple rule base. It was reasoned that the simpler the rule, the more expressive the rule would be. As an example consider the following two rules:

If systolic is very high **And** diastolic is high **And** mean is very high **And** heart rate is low **And** saturation is normal **Then** intervention

If mean is very high **Then** intervention

If sufficient support for the second rule can be found, then such a rule is to be preferred as an explanation since it requires less processing by the anes-

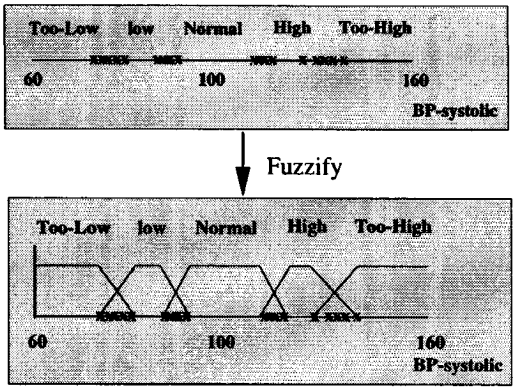


Figure 6.3: *Example of how fuzzy sets can be obtained from interviews. In this example several anesthetist were asked to force a crisp decision-boundary between several qualifications.*

thetist. The trigger is much more clear in case of such simple rules. By the non-disjoint rule induction method such simple rules are much more likely to be obtained than by the disjoint rule induction method.

6.3.4 Data Acquisition

An important aspect of the experiment was the acquisition of the examples. For this purpose a data-acquisition unit was developed that could communicate with the commercial monitor present in the operating theater. Over a time period of several weeks, data were collected during the operation. Further, the data was annotated with interventions from the anesthetist. Each annotation was associated with a time label indicating when the intervention occurred. The whole data acquisition appeared to be very time-consuming and when about 11 fully monitored and annotated operations were obtained, the acquisition was stopped.

In order to obtain a set of training data, the following parameters were used:

- systolic blood pressure,
- diastolic blood pressure,
- mean blood pressure,
- end-tidal CO2,
- oxygen saturation,
- heart rate.

By hand nearly 100 interventions and an equal number of non-interventions were selected. Although during the operation the number of interventions is usually much smaller than the number of non-interventions, they were treated as equally important for learning.

Three different data sets were formed for further study. In one data set the parameters of the 200 examples were all represented by the absolute values actually measured. In the second data set the parameters of the 200 examples were represented by relative values, relative with respect to the pre-operative values of the parameters. In the third data set the parameters of the 200 examples were also represented by relative values, but now relative with respect to the values of the parameters at the beginning of the operation. It was expected that these relative values would provide more accurate classification since they represent a "change" of the parameter value (and we already argued that interventions may be guided by relevant changes in the parameters).

6.3.5 Results and Conclusions

The data sets were split in a training and test set for both the ASA1/ASA2 class and the ASA3/ASA4 class. The overall classification performance on the test set was at best about 80%. When applied to a complete operation the results of the classification looked like the one depicted in Figure 6.4. It can be seen that the classification score nicely correlates with the decreasing blood pressures. Other classification algorithms such as k-nearest-neighbors and Parzen Windows performed not quite as well (although not much worse). The number of general rules generated was between 6 and 10 rules, depending on the type of data set. It was found that these rules were indeed very simple and surprisingly accurate. However, nearly all the rules only used the blood pressure parameters, indicating that nearly all the interventions were triggered by the blood pressure. It was also found that the rules learned on the basis of the relative data sets performed better than the rules learned on the basis of the absolute data set. Especially for the ASA3/ASA4 type of patients, the results on the basis of the relative values with respect to the pre-operative parameter value was about 15% better than the absolute value. This seems to confirm the assumption that changes form the trigger.

With respect to the fuzzy sets it was concluded that these can be rather simply elicited from the anesthetists. Most important in this matter is that up to seven qualifications can be easily understood by the anesthetist (in ascending order: very low, low, normal, high, very high) and are useful for the blood pressure parameters and the heart rate paramter. For the other parameters three fuzzy sets seemed to be sufficient. The actual shape did not matter much, what mattered was the formation of sets for different patient classes. With respect to the performance it was concluded that the FILER package worked well. The derived rules were simple and made sense and their performance was excellent, at least when compared to the other classification techniques. However it was also concluded that the number of data was much too small to make a relevant extrapolation beyond this feasibility study. Further, the type of

interventions were more or less all the same (all related to the blood pressure). This also indicated that much more data are needed to obtain a system that can be used in a general anesthesia monitor. Finally it was concludes that trends should be used to represent the parameters in time.

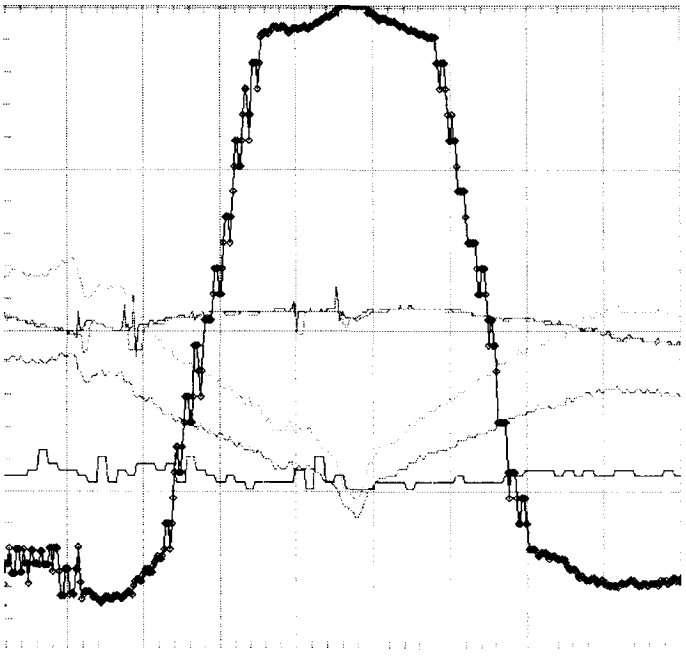


Figure 6.4: *Example of a trigger corresponding to decreasing blood pressures. The three decreasing blood pressures (Systolic, diastolic and mean) induce the trigger (bold line going up). Normalization of the trigger certainty between 0 (at the bottom of the scale) and 1 (at the top of the scale).*

6.4 Case Study

In order to learn the triggers from examples it is necessary to have a large quality database. Thanks to the University of Groningen we were able to use the Carola database [67]. Although the Carola database has not been designed to learn the patterns for our purpose, we think that from a case study much can be learned for future design of the analysis stage of the monitor and the requirements for databases used. Unlike the feasibility study, we were not so much interested in the formation of the fuzzy sets, but more whether such a large database could be used for learning triggers related to changes in the parameter values. Further, we wanted to investigate what trend information would be useful for representing the parameters in time.

6.4.1 Database

The part of the Carola database that we used, consists of approximately a thousand individual operations, each recorded from begin to end. For the part of the database that we used the type of operation was the same for all: cardiac bypass surgeries. A typical cardiac surgery can be divided in several stages: "initiation" (all before opening the sternum), "intermediate", "bypass" and "recovery". Several physiological parameters are measured and recorded: the arterial blood pressure, SpO2, pulmonary blood pressure, central venous pressure, heart rate and body temperature. However, only during the intermediate stage all these parameters were measured. The only parameter recorded always is the mean arterial blood pressure. Apart from the parameters all the medicine-based interventions of the anesthetist have been recorded. The medicine-based interventions can be categorized into two classes: bolus (one-time shot) and continuous (by infusion).

Data quality

Although the parameters are recorded every minute, almost all parameters suffer from noise. This is partly caused by the fact that the values in the database are not averaged over the minute they represent, which results in a poor minute-based signal. Further, as mentioned earlier, during many stages parameter values are missing or invalid (mean blood pressures of 180). A final problem with the quality of the data are the time labels of the interventions. They are stored per minute, and hence, have an accuracy of +/- one minute. Even more serious, they are likely to have a time lag which depends on the anesthetist; usually the anesthetist immediately enters the intervention after it is done, however, sometimes the anesthetist has to wait for several minutes before he can enter the intervention (especially if more interventions at the same time were performed). Hence, the final accuracy of the interventions lies somewhere in the plus one to minus five minute range.

These problems form restrictions on the automatic selection of examples of interventions. It is necessary to pre-process the data by estimation of missing values (introducing bias), filtering (introducing an even larger time lag than one minute) and by "optimizing" the time labels of the interventions (bias).

6.4.2 Selection of Interventions and Features

A rough estimate of the relative number of interventions versus the relative number of non-interventions is 1 to 10, i.e. 1 intervention in every 10 minutes. Many of these interventions are default:

- all stages: Glucose-solution, KCl and some anesthetic (infusion),
- initiation stage: Midazolam, Pancuronium, Dexamethason, Cefamandol (bolus),
- intermediate stage: heparine, sufentanil (infusion)

- recovery stage: Nitroglycerine (infusion), protamine (bolus),

Since we are looking for interventions which are representative for "triggers", these default interventions are not of much help. We looked therefore at the "non-default" bolus interventions, since infusions in general are given too much by default or based on expectations on the long term.

These non-default bolus interventions are even less frequent and occur on average once in every 200 minutes (on average: almost one in every operation, a typical operation last about two to three hours). The majority (90%) of these bolus interventions consist of two classes: sufentanyl and phenylefrine shots. The first is given to increase the anesthesia depth (usually if an increase of the blood pressure or heart rate occurs), whereas the second is a vasopressor given when the the blood pressure is too low. These two interventions have been used throughout our experiments as they seem to be good representatives of a too high and a too low blood pressure or heart rate.

Complicating factors with these interventions are twofold. First, sufentanyl is often given on the basis of what is expected by the anesthetist, for example almost always before the opening of the sternum (the end of the initiation stage) sufentanyl is given as a predictive response (or anticipatory action) to the pain-stimulus which comes with the opening. Such a predictive response cannot be observed from the parameters but comes from communication between the surgeon and the anesthetist. This implies that sufentanyl may not be a very good representative intervention for a trigger. Second, sufentanyl is almost only given in the "initiation" stage and phenylefrine is almost only given during the "by-pass" stage (in general, the blood pressure is too high in the intermediate stage and too low in the "by-pass" stage). Therefore, interventions have to be selected from both stages. Unfortunately the only parameter available in both stages is the mean arterial blood pressure. This means that the only patterns that can be learned are patterns for the mean arterial blood pressure. Multisignal patterns can therefore not be learned with this data base, reducing the accuracy of the (pattern) analysis and performance.

Feature extraction

Since the only available parameter for the interventions is the arterial blood pressure, we derived our initial features from the arterial blood pressure only. The features used are:

- 30 minute moving average,
- momentary difference with moving average,
- 3 minute trend (obtained by linear regression),
- 7 minute trend (obtained by linear regression),
- 15 minute trend (obtained by linear regression),
- 30 minute trend (obtained by linear regression).

The motivation for using these features is the following. The 30 minute moving average can be regarded as a base-line or target for the blood pressure. The momentary difference reflects the actual blood pressure, but, in order to limit the patient variability it is calculated as a difference with the moving average (also in his thesis Ballast proposes moving averages as useful features for detecting differences [10]). Finally, it has been often stated by experts that trends (up to 30 minutes) are very important. Since it is unknown which trends are of interest, several trends have been extracted to cover a range between 1 and 30 minutes.

Before feature extraction we took several preprocessing steps. First, we estimated missing values by taking the previous "known" value. Second, we filtered the data by using a spot-noise filter (median filter). Finally, we used only those values for trend extraction that were indeed recorded (sometimes 30 minutes were not available and a 30 minute trend was then calculated by as much minutes as available).

6.4.3 Selection of Examples

Due to the small number of interventions, it is difficult to learn the difference between an intervention and non-intervention. Therefore, we selected all the interventions and an approximately equivalent number of non-interventions. The interventions were drawn out of approximately 800 operations in the intermediate and by-pass stage, and the non-interventions were randomly chosen from the same operations in the same stages. Due to the inaccuracy of the time labels, the time labels were optimized over the previous four minutes according to the largest momentary difference (hence biasing an intervention). For non-interventions a random time label was selected of at least 15 minutes before or after an intervention (preferably 30 minutes but then many interventions may be missed), further, the non-interventions were biased by optimizing over +/- 2 minutes according to the smallest momentary difference. The purpose of these biases is to obtain better representative data of both classes. Using these selections we obtained a data set of approximately 900 interventions and non-interventions, representing three classes: sufentanyl, phenylefrine and non-interventions. The other 200 surgeries were left for validation afterwards.

6.4.4 Results

FILER was especially designed for obtaining a rule-based classifier such that both classification and explanation are possible with these rules. In the FILER algorithm the type of generalization used was non-disjoint rule induction as also used in the feasibility study. Apart from FILER, the double-kernel estimator was used for benchmarking. The classification error for both classifiers was estimated by using 10 fold cross-validation which was repeated 5 times to obtain the variance. It turned out that the mean-square error (standard deviation) was always smaller than 0.5%. o

Table 6.1: *Results FILER 345 fuzzy rules (overall result: 76.3% +/- 0.5) trigger vs. no-trigger: 79.3%.*

CLASS	ZERO	ONE	TWO
zero	73	11	24
one	7	86	6
two	20	3	70

Table 6.2: *Results FILER 42 fuzzy rules (overall result: 70.6% +/- 0.5), trigger vs. no-trigger: 74.3%.*

CLASS	ZERO	ONE	TWO
zero	65	13	29
one	9	84	8
two	26	3	63

Final results are shown in the tables by using the so-called confusion matrix, all results are expressed in percentages. Horizontally the true class-label is given, vertically the classification by the algorithm is given, therefore the sum of each column is 100.

- Class ZERO is the non-intervention class,
- Class ONE is the Phenylefrine intervention Class (BOLUS),
- Class TWO is the Sufentanyl intervention Class (BOLUS).

As an example, the first entry in the second row of the matrix in the first table should be read as: "7% of all non-interventions is classified as a phenylefrine intervention".

The best result was obtained by FILER for 7 fuzzy sets per feature, the total performance was 76% obtained with the specific rules, see Table 6.1. The data set could also be explained in about 42 general rules with an total accuracy of

Table 6.3: *Results FILER 54 crisp rules (overall result: 69.3% +/- 0.5), trigger vs. no-trigger: 73%.*

CLASS	ZERO	ONE	TWO
zero	69	14	36
one	10	83	8
two	21	3	56

Table 6.4: *Results kernel-distribution estimator (overall: 72.3% +/- 0.5), trigger vs. no-trigger: 74.3%.*

CLASS	ZERO	ONE	TWO
zero	78	16	39
one	8	82	4
two	14	2	57

70.5%+- 0.5%, see Table 6.1. It shows that these results are very good when compared with the kernel distribution estimator (see Table 6.4), the specific rules being more accurate and the general rules being somewhat less accurate. However, when considering both the phenylefrine and the sufentanyl interventions as triggers, then the total trigger specificity is 82% (using equal trigger and non-trigger a priori probabilities). This is an excellent result when compared to the 10% specificity of existing monitors.

As an additional study, the experiment was repeated using crisp sets. These results are summarized in Figure 6.5 for the general rules and for the specific rules. The confusion matrix for the optimal crisp rule base is given in Table 6.3. Clearly, using a fuzzification leads to more accurate results with less rules than using a quantization. The gain in accuracy for the general rules is rather small, about 1%, whereas the gain in the classification accuracy for the specific rules is 7%. This discrepancy between the general and the specific fuzzy rules suggests that a more refined generalization is needed for this problem to profit from the high accuracy of the specific rules. However, the reduction in the number of rules is large: more than 20% less rules are used in case of a fuzzification. This implies that the fuzzy rules are more general, which makes it easier to interpret these than the crisp rules. Further, the simpler the rule base, the faster the classification. This important advantage of the fuzzy rule base makes real-time decision support more feasible when much larger databases with much more features become available.

6.4.5 Feature Selection

Using feature selection, the original six features could be reduced to three without losing performance. In essence this was done by repeating the previous exercise on a selection of features. The features were selected by investigating the most frequently occurring features in the rule base (in the appendix an example is given of such a rule base containing 42 rules). For six features FILER needed about 42 rules in its optimal rule base whereas for three features FILER needed about 30 rules. The three most important features found were:

- 30 minute moving average,
- momentary blood pressure (measured relative to moving average),

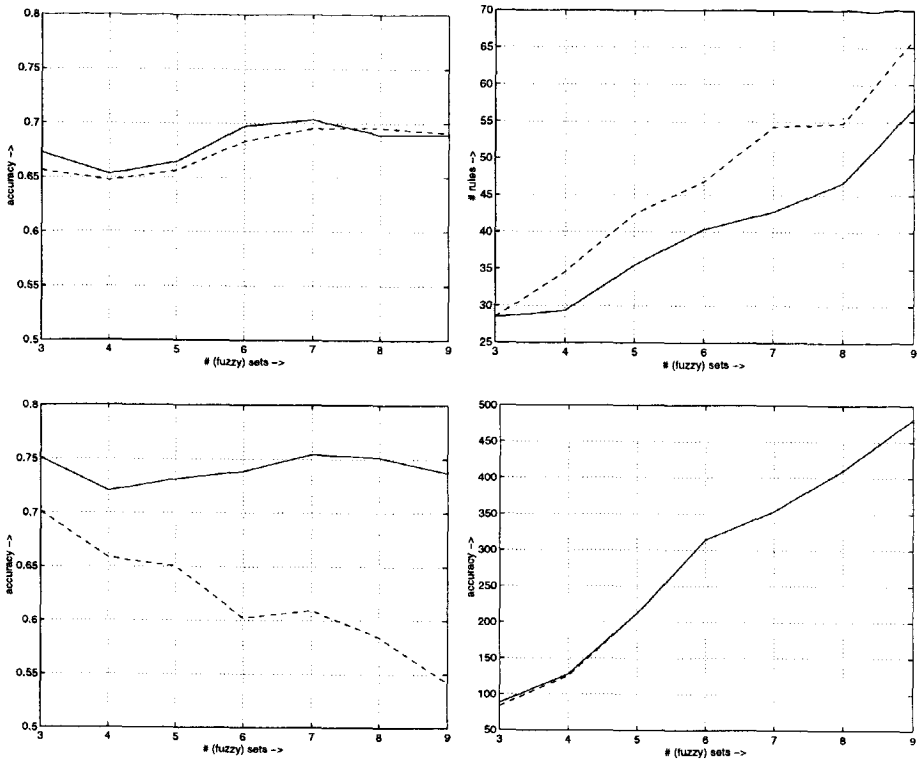


Figure 6.5: *Results obtained from the general rules (Top) and from the specific rules (Bottom). Results of the crisp sets are depicted by the dashed lines and of the fuzzy sets by the solid lines. Left figures show accuracy, right figures show the number of rules.*

- 30 minute trend.

The 30 minute moving average can be regarded as the target. This implies that anesthetists behave not only as a proportional controller based on the momentary blood pressure, as suggested in the case study, but also use parameter changes reflected in the 30 minute trend for prediction.

6.4.6 Performance on the Remaining Operations

When from the remaining operations a validation data base was constructed in the same way as for the training/test data base during, no significant differences were observed for the general rules. All results lied within 1% of the results obtained previously. The most interesting part of the study was the analysis of the full operations (intermediate and by-pass stage) with the general rules. We selected several operations with sufficient action (i.e. sufficient

sufentanyl/phenylefrine interventions).

On the full operations the general rule base was used for analysis and the results were compared with the expert decisions. The general rule base was so small and simple that a complete operation (of two hours and more) could be analyzed within seconds, even with the specific rules it didn't take much more than a minute. Hence, the real-time requirement should pose no problem. The overall performance (predictive accuracy) was approximately 73% with the a priori probabilities[1] set to 0.65 for a normal situation and 0.35 for a trigger. It should be noted that an overall performance of 95% can be easily obtained on these operations by simply never triggering the anesthetist. However, in that case all interventions are missed. The remaining 27%, the error rate, contained both "false" triggers and missed triggers. About 25% "false" triggers where given by the system, however, these triggers still seemed reasonable to a layman, although the anesthetist did not perform an (sufentanyl/phenylefrine) intervention. Quite often, though, the anesthetist intervened (within a few seconds difference form the system trigger) which was not phenylefrine or sufentanyl, but a more "regular" infusion. This indicates that the triggers learned with phenylefrine and sufentanyl possess some generality. The final 2% contained the missed triggers. However, since the number of interventions is much smaller than the number of "normal" situations, this 2% represented about 30% of all the (sufentanyl and phenylefrine) interventions. Although 30% of all of these interventions were missed (nearly always sufentanyl), quite often a minute later or earlier a "false" intervention was given. This means that the system does not give a trigger at the exact time that the anesthetist would intervene, nevertheless this was counted as a false trigger. Hence, a good evaluation of the performance on the full operations can only be obtained by using expert evaluation. Further, such an evaluation may indicate that the system is actually more accurate than the figures presented.

6.4.7 Preliminary Conclusion

FILER performs well when compared to the other classifiers. Best results for the data set are around 76% of total performance. A performance of about 76% indicates that the three classes are not perfectly separable. The class leading to this rather low separability is clearly the class of sufentanyl interventions. This is not surprising since we already mentioned that often sufentanyl is given as a "default" intervention to keep a sufficient anesthesia depth. Actually, what is surprising is that still about 70% of the sufentanyl cases can be classified correctly.

This study shows that even with only one physiological signal, measured only once a minute, significant triggers can be learned. Further it is striking that the triggers learned are clearly related to *changes* in the physiological parameters.

[1] Almost 2% (sufentanyl/phenylefrine) interventions were present in the remaining operations studied, so an a priori setting of 0.95 and 0.05 would be reflecting this. However, we argued that the cost for a missed intervention is much higher than the cost for an extra trigger. We estimated the cost ratio as 10 to 1.

Despite these fair results, we expect that the accuracy can increase significantly if:

- the minute-based values are minute-based averages (preferably with some means of trend within the minute),
- more physiological signals are measured continuously and reliably (some means of validation should be available before the measurements are stored),
- time labels of the interventions are annotated more accurately,
- an anesthetist indicates the "trigger" class directly rather than indirectly by medicine,
- expert analysis of results is available (leading to a better estimate of the performance).

6.4.8 Discussion with an Expert Panel

The result were presented to an expert panel consisting of two experienced anesthetists and a medical computer-science expert working in anesthesia. The experts were surprised (in positive sense) by the amount of knowledge and understanding which has been extracted by using the database in the short time available (the preprocessing, analysis and rule generation took about a month). However, they also pointed out that the knowledge found is not new. The results merely confirmed what they knew already, both interventions are highly correlated with the blood pressure and indeed phenylefrine is a clear case and sufentanyl is not. One expert remarked that in only about 50% of the cases where he *considered* to apply sufentanyl he actually *did* apply it. This seems to be reflected in the accuracy by which the sufentanyl interventions are recognized with the rules. Further, they remarked that the time labels are more likely to be inaccurate due the *decision process* rather than the annotation process. The decision process is simply not that clear and some time is needed for this process to take place between the moment of "self-triggering" ("hey, the blood pressure seems a little high") and the actual intervention. The experts also looked at the generated rules and they evaluated them in a natural way by looking at the most frequent and most certain rules for a class. These kind of rules can be thought of as being the most representative rules for a class. They clearly recognized the rules without problems and confirmed their use. However, this also implies that the rules should be presented by their order of *representativeness* (in FILER these rules are known as the most informative rules and this concept is even used by FILER to generate rules). Finally, the experts showed great enthusiasm for using such a system for other problems such as learning the pre-intubation dose of an anesthetic on the basis of patient data and pre-operative blood pressure.

6.5 Conclusion

We have demonstrated the use of fuzzy probabilistic rule induction for a decision-support system in an exacting environment: anesthesia monitoring. By regarding the expert as a stimulus-response system, knowledge with respect to triggers has been acquired by learning from examples. The obtained rule base is can accurately recognize and explain triggers in real-time with simple and expressive rules, that can draw the anesthetist to the cause of the trigger.

On the basis of the results we conclude that a robust warning system can be designed using rule induction. The total number of warnings can be significantly reduced when compared to current monitors. Further, these warnings can be much more meaningful than those of current monitors. However, the specificity of the warning system is still a challenge. If the current rule base would be used as it is (on the basis of sufentanyl/phenylefrine interventions) then the anesthetist will be triggered on average once in every five minutes, with a specificity of about 10% and with 30% missed triggers. Although 90% of the triggers does not result in an intervention, this does not mean that all these triggers are wrong. On the basis of our experience, probably 80% of these triggers is simply correct. It is not unusual that the anesthetist himself performs an evaluation of a trigger (due to a change in the patient signals) every five minutes which in only 10% of all cases results in an action. The problem lies not in the specificity of the triggers, but in the evaluation of these triggers. Often other observations and circumstances, not always visible in the physiological signals, are used by the anesthetist to evaluate these triggers and to decide whether he should intervene or not. It may very well turn out that these "additional" observations finally determine the specificity of a warning system, rather than the triggers recognized in the patterns. Hence, the focus of further research should be the evaluation of the triggers in the strategy stage of the system. On the basis of such an evaluation, the strategy stage can decide to warn the anesthetist. However, care should be taken not to cross the delicate line between decision-support systems and decision-making systems.

In triggering for decision-support some improvements can be made as well. First, the examples on the basis of which the triggers are learned can be selected with much more care. Clearly, the sufentanyl intervention used in our case study is not sufficiently specific for learning triggers. Second, more interventions should be taken into account, as well as more patient signals. However, this will require a sophisticated data base. In the end, the reliability of the obtained knowledge greatly depends on the available database. To develop the next generation of anesthesia monitors, collection of annotated and validated data is a prerequisite.

Chapter 7

Discussion

In this thesis we have been concerned with knowledge acquisition for decision-support systems in exacting environments by learning from examples. We have pointed out in the introduction that the acquired knowledge in such environments should provide both a good data fit and a good mental fit to the decision problem. Although the data fit can be measured by the error rate over "unseen" examples (predictive accuracy), the mental fit is much more difficult to measure. One of the properties of a good mental fit is the expressiveness of the knowledge, which for a rule base - generally acknowledged as the most expressive representation of knowledge - can be obtained (1) by using the expert's frame of reference, and (2) by reducing the complexity[1] of the rule base as much as possible. Most existing algorithms in Artificial Intelligence (i.e. Pattern Recognition and Machine Learning) provide either a good data fit or a good mental fit. Therefore, we have focused on developing an algorithm that provides a good data fit as well as a good mental fit.

We have synthesized density estimation from the pattern recognition domain with fuzzy rule induction from the Machine Learning domain. Here, density estimation has an emphasis on data fit, whereas rule-induction has an emphasis on mental fit. The synthesis has resulted in the fuzzy probabilistic rule induction algorithm. We have demonstrated in a comparative study with nine other rule induction algorithms (incl. decision trees) that fuzzy probabilistic rule induction provides both a good data fit and a good mental fit. Further, we have shown how the system can be applied in anesthesia monitoring for real-time recognition and explanation of patterns that should trigger, warn, the anesthetist. On the basis of the results obtained from a case study and a discussion with an expert panel we are confident that the system is very suitable for decision support in anesthesia monitoring.

[1]The complexity can for example be measured by the total number of rules in the rule base.

7.1 Reflections

We introduced a fuzzy probabilistic framework. At the heart of this framework lies the notion that both fuzzy sets and probability theory capture a specific type of uncertainty. Fuzzy sets essentially measure uncertainty on the basis of similarity, whereas probability theory measures uncertainty on the basis of occurrence. The fuzzy probabilistic framework takes both types into account by defining the probability of a fuzzy event. The framework accounts for reasoning with fuzzy probabilities, and generalizes many notions from both fuzzy sets and probability theory. We have shown that the framework reduces to probability theory when crisp events (sets) are used, whereas it reduces to fuzzy logic if the probabilities are assumed to equal one. Throughout the thesis we have hold on to a more "probabilistic view" of the fuzzy probabilistic framework. We have not encountered any inconsistencies in the framework, and are therefore confident that fuzziness and probability can coexist in a single framework. As a matter of fact, we think that it provides a more complete, general, probability theory than the relative frequency view on probability does.

From the fuzzy probabilistic framework we have derived two algorithms: the double-kernel estimator and the fuzzy probabilistic rule induction algorithm. Both are more efficient than the methods to which they are related: to Parzen Windows and (information-theoretic) rule induction, respectively. Because of their efficiency the two algorithms can make predictions that are at least as accurate with less concepts (kernels, rules) than the methods to which they are related. Apart from the generalization in the fuzzy probabilistic rule induction algorithm, the two algorithms are not really different. The main difference between them is the way they deal with the covariance of the data. In double-kernel estimation the covariance is estimated from the data whereas in fuzzy probabilistic rule induction the covariance is assumed to be zero, i.e. it is assumed that the features are statistically independent. For the specific rules only the local covariance is assumed to be zero, but during the generalization this assumption is extended to the whole data set. The advantage of the independence assumption, and the actual motivation behind it, is that a more general and meaningful description of the decision-making process can be obtained by which the decision can be explained. The disadvantage of this assumption is that it sometimes results in somewhat more errors in decision making than methods that take the covariance into account. However, this is not generally the case. The estimation of covariances is sometimes erroneous, especially for decision making in high dimensions for which there are not a lot of data available. In these cases, fuzzy probabilistic rule induction can still be more accurate.

Fuzzy probabilistic rule induction follows the Data-Information-Knowledge paradigm. The information layer consists of the most specific rules that can be obtained according to the reference frame. These specific rules may be a little too complex for providing a good mental fit to the decision problem - too complex for explaining the decision - but they provide a good data fit. The knowledge layer is more appropriate for the mental fit since it consists of simple general rules obtained by generalization of the specific rules. Through general-

ization the number of rules and their complexity is reduced. The general rules arrive at accurate decision-making, but it is the expressiveness of these rules that is much better than the expressiveness of the specific rules. The reason for this is that they try to capture the "essence" of the knowledge latent in the examples. However, these general rules hardly ever obtain a significantly higher accuracy than the specific rules, if at all. This may be due to the biased generalization procedure used to obtain the knowledge. This bias is caused by the fact that the general rules are projections on the feature axes, so, the estimated decision boundaries become somewhat parallel to these axes. Therefore, we suggest a default strategy: to use the specific rules for accurate decision-making, and to use (non-disjoint) general rules for providing the most informative reason underlying the decision made. In this way the data fit and the mental fit are somewhat de-coupled. Note that this strategy is only useful when a fuzzy reference-frame is used. In rule induction that uses a quantization, generalization is both necessary for data fit and for mental fit. This necessity has been indicated by both our experiments with crisp reference-frames, and a vast amount of literature on rule induction in the field of Machine Learning. By using the above strategy, fuzzy probabilistic rule induction can keep a very good balance between data fit and mental it.

So what have we learned about knowledge acquisition for decision-support systems in exacting environments? Compared to knowledge acquisition by expert interviews, we have given the expert a more tactical role. First of all, he (or better: a panel) should provide examples of the decision-making problem. Second, he should be involved in formulating the reference frame. Third, he should evaluate the rules after they have been formed by the system. Finally, he should play a central role in the validation of the system in operational settings. For some of these role aspects one would expect to be able to use existing interviewing techniques for knowledge acquisition. Unfortunately, existing interviewing techniques do not accommodate any of these aspects. First of all they do not focus on examples but, instead, try to formalize the knowledge directly from interviews. Second, they do not provide means to separate the reference frame from the possible associations, but try to solve this all-in-one. Third, they do not provide a structured approach for tuning the system if it does not pass the evaluation or validation. Hence, what we have learned is that interaction with the expert remains necessary if rule induction is used, but that there is a need for interactive techniques which support the tactical role of the expert. Further, we have learned that fuzzy probabilistic rule induction can reduce this interaction to an effective minimum, once an annotated data base is available.

7.2 Further Extensions

From a cognitive point of view, we have mainly dealt with learning on the stimulus-response level. The current system learns very basic knowledge about stimulus and response but is still a poor reflection of our "rich" and "meaningful" mental world. Further extensions of the current system should be directed

towards obtaining more meaningful knowledge. We see two possible directions: (1) optimizing the reference frame over multiple decision-making problems, (2) multistage decision-making.

One direction involves the formation of the reference frame. Essentially we have regarded the formation of a reference frame as a separate (unsupervised) learning problem; obtained from either clustering or from experts. However, we have only dealt with forming a reference frame for a single decision problem. Since a decision problem can be expressed in several reasonable reference frames from a data-fit point of view, the most suitable reference frame can be "picked" by the expert. However, if a reference frame has to be picked out of several possible ones, it is also an option to study the problem of expressing several decision-making problems in a single reference frame. In this way the reference frame can become more meaningful, since more meanings can be associated to it. Further, such a single reference frame makes it easier to find relations between separate decision-making problems. For example, in anesthesia monitoring, to decide upon the patient status is one thing but to decide upon the proper treatment is another. Although the latter decision-making process is not only based on the physiological signals - but also on knowledge concerning the effects of certain treatment on the patient (e.g. drugs) - it can make use of the same reference frame for the physiological signals. The reference frame can then be obtained from an optimization for both decision-making processes.

Another direction for obtaining meaningful knowledge is by designing multistage systems. After all, our knowledge seems to be somewhat hierarchically ordered, and a multistage system could reflect such a hierarchy. Further, in a multistage system it could be possible to decompose the covariance of the data in a set of overlapping reference frames, one for each stage. By reasoning with uncertainty across the different stages of the system, for which the fuzzy probabilistic framework provides ample foothold, the problem of taking the covariance into account while still being able to express the knowledge in meaningful terms, can perhaps be solved.

The suggested directions will probably require a lot of data, but can provide a rather clear and compact organization of the data for the user. Therefore, we think that the directions suggested above should be topics addressed in the area of data mining and data warehousing.

7.3 Future Research

Reflecting on the past and future of knowledge acquisition, we can indicate the following approaches. First, knowledge was acquired through interviewing experts, and more or less directly "implanted" in the system. This can be characterized as "learning by being told". Second, knowledge is acquired through algorithms capable of inductive learning or "learning from examples". Third, knowledge will be acquired through a process characterized as "learning by self-organization". The latter statement requires some explanation.

At the beginning of this thesis we stated that learning is a process of reduc-

ing the uncertainty in our knowledge. We have gone in much detail to make this uncertainty explicit and measurable. We have used the Data-Information-Knowledge paradigm to design a learning algorithm that is able to reduce the uncertainty. We even deployed an information-theoretic measure to quantify the amount of uncertainty that is reduced in order to guide the learning process. As a result we have obtained a useful rule induction algorithm. However, there is still much to learn about systems that reduce uncertainty. For a start, we could re-examine nature, and look for systems that reduce uncertainty by an innate principle. The understanding of such an innate principle could be helpful for creating systems that acquire "rich" knowledge. We are confident that these systems already exist, we ourselves are a complex one for sure. We are also confident that these systems exist because the concept of uncertainty is closely related to the concept of entropy, and nature is full of systems that reduce entropy. In these systems the reduction of entropy, i.e. organization, is not specified in advance but arises as a result of an innate principle. Hence, such systems can be called self-organizing. We think that the seeds for studying these types of systems have already been planted in fields like Chaos Theory, Artificial Life and Autonomous Agents. We suggest that some long-term research should be directed to understanding these systems and making their underlying principle suitable for knowledge acquisition.

Bibliography

[1] S. Abe and M.S. Lan, A Method for Fuzzy Rules Extraction Directly from Numerical Data and Its Application to Pattern Recognition, *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 1, pp. 18 - 28, 1995.

[2] S. Abe and R. Thawonmas, A Fuzzy Classifier with Ellipsoidal Regions, *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 3, pp. 358- 368, 1997.

[3] J.H.J. Almering, *Analyse*, DUM, 1988.

[4] M.R. Anderberg, *Cluster analysis for applications*, Academic Press, New York, 1973.

[5] American Society for Anesthesiologists, New Classification of Physical Status, *Journal of Anaesthesiology* vol. 24, no. 11, 1963.

[6] G.A. Babich and O.I. Campus, Weighted Parzen Windows for Pattern Classification, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, pp. 567-570, 1996.

[7] R. Babuska, *Fuzzy Modeling and Identification*, Ph.D. Thesis, Delft University of Technology, the Netherlands, 1997.

[8] E. Backer, A non-statistical type of uncertainty in fuzzy events, in: *Colloquia Mathematica Societatis:topics in information theory*, J. Bolyai (Ed.),Keszthely (Hungary), pp. 53-73, 1975.

[9] E. Backer, *Computer-Assisted Reasoning in Cluster Analysis*, Prentice Hall International, 1995.

[10] A. Ballast, *Warning Systems in Anesthesia: Human vigilance supported by Clinically Relevant Warnings*, Ph.D. Thesis, Groningen: Rijksuniversiteit, 1992.

[11] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[12] J.C. Bezdek, A Review of Probabilistic, Fuzzy and Neural Models for Pattern Recognition, *Journal of Intelligent and Fuzzy Systems*, vol. 1, no. 1, pp. 1 - 25, 1993.

[13] N.M. Blachman, The amount information that y gives about X, *IEEE Transactions on Information Theory*, vol. 14, pp. 27 - 31, 1968.

[14] L. Breiman, W. Meisel and E. Purcell, Variable Kernel Estimates of Multivariate Densities, *Technometrics*, vol. 19, pp. 135-144, 1977.

[15] L. Breiman, J. Friedman, J. Olsen, and C. Stone, *Classification And Regression Trees*, Wadsworth International Group, 1984.

[16] J. Breuker and B. Wielinga, Models of expertise in knowledge acquisition, in: *Topics in expert system design: methodologies and tools*, G. Guida and C. Tasso (eds.), North Holland Publishing company, Amsterdam, The Netherlands, 1988.

[17] I. Bruha, Quality of Decision Rules; Definitions and Classification Schemes for Multiple Rules, in: *Machine Learning and Statistics; the interface* G. Nakhaeizadeh and C.C. Taylor (eds.), John Wiley & Sons, Inc., 1997.

[18] J.S. Bruner, J.J. Goodnow, and G.A. Austen, *A study of thinking* New York, Wiley, 1956.

[19] B.G. Buchanan and T. Mitchell, Dendral and Meta-Dendral: their Applications Dimension, *Artificial Intelligence*, vol. 11, 1978.

[20] B. Cestnik, I. Kononenko, and I. Bratko, Assistant 86: A Knowledge Elicitation Tool for Sophisticated Users, *Proc. of the 2nd European Working Session on Learning* pp. 31 - 45, Bled, Yugoslavia: Sigma Press, 1987.

[21] J.L. Chameau and J.C. Santamarina, Membershipfunctions: Comparing Methods of Measurement *and* Trends in Fuzziness and Implication, *Int. J. of Approximate Reasoning*, vol. 1, pp. 287 - 317, 1987.

[22] Z. Chi and H. Yan, ID3-Derived Fuzzy Rules and Optimized Defuzzification for Handwritten Numeral Recognition, *IEEE Transactions on Fuzzy Systems* vol. 4, no. 1, pp. 24 - 31, 1996.

[23] A.T.W. Chu, R.E. Kalaba, and K. Springarn, A Comparison of Two Methods for Determining the Weights of Belonging to Fuzzy Sets, *J. of Optimization Theory and Applications*, vol. 27, no. 4, pp. 531 - 538, 1979.

[24] V. Chopra, J.G. Bovill, J. Spierdijk, and F. Koornneef, Reported significant observations during anesthesia: a prospective analysis over an 18 month period, *British Journal of Anesthesia*, vol. 68, pp. 13 - 17, 1992.

[25] V. Chopra, *Aspects of Quality Assurance in Anaesthesia* Ph.D. Thesis, Leiden University, 1994.

[26] P. Clark and T. Niblett, The CN2 Induction Algorithm, *Machine Learning Journal*, vol. 3, pp. 261 - 283 1989.

[27] K. Crocket and Z. Bandar, Fuzzy Rule Induction From Data Sets *Proc. 10th Florida Artificial Intelligence Research Symposium* Florida AI Research Society, 1997.

[28] P.M.A. de Graaf, G.C. van den Eijkel, H.J.L.M. Vullings, and B.A.J.M. de Mol, A decision-driven design of a decision support system in anesthesia, *Artificial Intelligence in Medicine*, vol. 11, pp. 141 - 153, Elsevier, 1997.

[29] P.M.A. de Graaf, *How to design an intelligent anesthesia monitor*, Ph.D. Thesis, Delft University of Technology, the Netherlands, 1998.

[30] D. Dubois and H. Prade, *Possibility Theory*, Plenum Press, New York, 1988.

[31] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, New York: John Wiley & Sons Inc., 1973.

[32] R.P.W. Duin, On the Choice of Smoothing Parameters for Parzen Estimators of Probability Density Functions *IEEE Trans. on Computers*, vol. 25, pp. 1175-1179, 1976.

[33] R.P.W. Duin, *On the accuracy of statistical pattern recognizers*, Thesis, Delft University of Technology, 1978.

[34] D.M. Dutton and G.V. Conroy, A Review of Machine Learning, *The Knowledge Engineering Review*, vol. 12:4, pp. 341 - 367, 1996.

[35] G.C. van den Eijkel, E. Backer, and J.J. Gerbrands, A Proposal for Fuzzy Rule Generation Using Temporal Reasoning, *Proc. of the first annual conf. of the Advanced School for Computing and Imaging (ASCI '95)*, J. van Katwijk, J.J. Gerbrands, M.R. van Steen, J.F.M. Tonino (eds.),Heijen, The Netherlands, May 16 - 18, pp. 47 - 55, 1995.

[36] G.C. van den Eijkel, E. Backer, and J.J. Gerbrands, On Intelligent Data Analysis for Hierarchical Rule-Base Generation, *Proc. IDA-'95: Advances in Intelligent Data Analysis Vol. I* X. Lui, G.E. Lasker (eds.), Baden-Baden, Germany, August 16-20, The Int. Institute for Adv. Studies in Systems Research and Cybernetics, Windsor, Canada, pp. 49 - 53, 1995.

[37] G.C. van den Eijkel, J.C.A. van der Lubbe, E. Backer, and J.J. Gerbrands, Fuzzy-Rule Generation Using Incremental Learning for a Knowledge-Based Anaesthesia Monitor, *Proc. of the second annual conf. of the Advanced School for Computing and Imaging (ASCI '96)*, E.J.H. Kerckhoffs, P.M.A. Sloot, J.F.M. Tonino, A.M. Vossepoel (eds.), Lommel, Belgium, June 5 - 7, pp. 271 - 276, 1996.

[38] G.C. van den Eijkel, J.C.A. van der Lubbe, and E. Backer, Fuzzy Incremental Learning of Expert Rules for a Knowledge-Based Anesthesia Monitor, *Proc. of the fourth European Congress on Intelligent Techniques and Soft Computing (Eufit '96)*, Aachen, Germany, sept. 2-5, vol. 3, Verlag Mainz, Aachen, pp. 2056 - 2060, 1996.

[39] G.C. van den Eijkel and E. Backer, Knowledge Acquisition using a Fuzzy Machine-Learning Algorithm for a Knowledge-Based Anesthesia Monitor in CD-ROM: *Proc. of the 18th Annual International Conf. of the IEEE Eng. in Med. and Biol. Society (EMBS '96): Bridging Disciplines for Biomedicine* Oct. 31 - Nov. 3, Amsterdam, The Netherlands, Soekekouw Videoproductions, Nijmegen, The Netherlands, 1996.

[40] G.C. van den Eijkel, J.C.A. van der Lubbe, and E. Backer, A Modulated Parzen-windows Approach, *Proc. of the third annual conf. of the Advanced School for Computing and Imaging (ASCI)*, H.E. Bal, H. Corporaal, P.P. Jonkers, J.F.M. Tonino (eds.), Heijen, The Netherlands, June 2-4, pp. 157 - 161, 1997.

[41] G.C. van den Eijkel, J.C.A. van der Lubbe, and E. Backer, A Modulated Parzen-windows Approach for Probability Density Function Estimation, *Lecture Notes in Computer Science: Advances in Intelligent Data Analysis - Reasoning About Data* X. Lui, P. Cohen and M. Berthold (eds.), Springer Verlag, Berlin, pp. 479 - 489, 1997.

[42] G.C. van den Eijkel, Rule Induction to appear in: *Introduction to Intelligent Data Analysis*, D. Hand and M. Berthold (eds.), 1998.

[43] J. Fan and J.S.Marron, Fast Implementations of Nonparametric Curve Estimators, *J. Computational and Graphical Statistics*, vol.3, pp. 35-56, 1994.

[44] C. Feng and D. Michie, Machine Learning of Rules and Trees, in: *Machine Learning, Neural and Statistical Classification*, D. Michie, D.J. Spiegelhalter, C.C. Taylor (eds.), Ellis Horwood, 1994.

[45] T.L. Fine, *Theories of Probability; an examination of foundations*, Academic Press, New York and London, 1973.

[46] R.A. Fisher, The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, vol. 7, pp. 179-188, 1936.

[47] K. Fukunaga, *Statistical Pattern Recognition*, San Diego, Calif: Academic Press Inc., 1990.

[48] Fuzziness vs. Probability, the N-th round, special issue of *IEEE Trans. on Fuzzy Systems*, vol. 2, no. 1 1994.

[49] D.M. Gaba, K.J. Fish, and S.K. Howard, *Crisis Management in Anaesthesiology*, Churchill Livingstone, New York, 1994.

[50] B.R. Gaines and M.L.G. Shaw, Induction of Inference Rules For Expert Systems, *Fuzzy Sets and Systems*, vol. 18, pp. 315-328, 1986.

[51] A.F. de Geus and E. Rotterdam, *Decision Support in Anaesthesia*, Ph.D. Thesis, Groningen University, 1992.

[52] Y. Hamamoto, S. Uchimura, and S. Tomita, A Bootstrap Technique for Nearest Neighbor Classifier Design, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 73 - 79, 1997.

[53] D.J. Hand, *Construction and Assessment of Classification Rules*, Wiley, 1997.

[54] W.K. Hardle and D.W. Scott, Smoothing by Weighted Averaging of Shifted Points, *Computational Statistics*, vol. 7, pp. 97 - 128, 1992.

[55] J. Harmse, Continuous Fuzzy Conjunctions and Disjunctions, *IEEE Trans. on Fuzzy Systems*, vol. 4, pp. 295-314, 1996.

[56] B. Hayes-Roth, R. Wahsington, D. Ash, R. Hewett, A. Collinot, A. Vina, and A. Seiver, Gaurdian: A Prototype Intelligent Agent for Intensive-Care monitoring, *Artificial Intelligence in Medicine*, vol. 4, pp. 165-185, 1992.

[57] T.B. Ho, E. Diday, and M. Gettler-Summa, Generating rules for expert systems from observations, *Pattern Recognition Letters*, vol.7, pp. 265-271, 1988.

[58] M. Holsheimer and A.P.J.M. Siebes, Data mining: the Search for Knowledge in Databases, *Technical Report CS-R9406* CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands, 1994.

[59] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms, *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 260 - 270, 1995.

[60] A.K. Jain and W.G. Waller, On the optimal number of features in the classification of multivariate Gaussian data, *Pattern Recognition*, vol. 10, pp. 365 - 374, 1978.

[61] A.K. Jain and M.D. Ramaswami, Classifier Design with Parzen Windows, in: *Pattern Recognition and Artificial Intelligence*, pp. 211 - 228, Elsevier Science Publishers B.V., E.S. Gelsema and L.N. Kanal (eds.), 1988.

[62] C.Z. Janikow, Fuzzy Decision Trees: Issues and Methods, *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 28, no. 1, pp. 1 - 14, 1998.

[63] M.C. Jones, Discretized and Interpolated Kernel Density Estimates, *Journal of the American Statistical Association*, vol. 84, pp. 733 - 741, 1989.

[64] O.A. Johnson, *The mind of David Hume: a companion to book I of A treatise of human nature*, University of Illinois Press, 1995.

[65] Personal communication with A. Kandel, 1995.

[66] A. Kandel On fuzzy statistics, in: *Fuzzy Set Theory and Applications*, R. Ragade, M. Gupta and R. Yager (eds.), New York: North Holland, pp. 181-199, 1979.

[67] G.F. Karliczek, A.F. de Geus, G. Wiersma, S. Oosterhaven, and I. Jenkins, Carola, a computer system for automatic documentation in anesthesia, *International Journal for Clinical Monitoring and Computing*, vol. 4, pp. 211-221, 1987.

[68] I.G. Kestin, B.R. Miller, and C.H. Lockart, Auditory Alarms during Anesthesia Monitoring, *Journal of Anaesthesiology*, vol. 69, pp. 106-109, 1988.

[69] E.P. Klement, Fuzzy σ-algebra's and Fuzzy Measurable Functions, *Fuzzy Sets and Systems*, vol. 4, pp. 83-93, 1980.

[70] G.J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic, Theory and Applications*, New Jersy, Prentice Hall Inc., 1995.

[71] I. Kononenko, Combining Decisions of Multiple Rules, in: *Artificial Intelligence V; Methodology, Systems, Applications*, B. du Boulay and V. Sgurev (eds.), Elsevier, Amsterdam, 1992.

[72] B. Kosko, Counting with fuzzy sets, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 556-557, 1986.

[73] B. Kosko, Fuzzy Entropy and Conditioning, *Information Sciences*, vol. 40, pp. 165-174, 1986.

[74] H. Kwakernaak, Fuzzy Random Variables - I, Defininitons and Theorems, *Information Sciences*, vol. 15, pp. 1-29, 1978.

[75] H. Kwakernaak, Fuzzy Random Variables - II, Algorithms and Examples for the Discrete Case, *Information Sciences*, vol. 17, pp. 253-278, 1979.

[76] R. Kruse The Strong Law of Large Numbers for Fuzzy Random Variables *Information Sciences*, vol. 28, pp. 233-241, 1982.

[77] P.A. Lachenbruch and M.R. Mickey, Estimation of error rates in discriminant analysis, in: *Technometrics*, vol. 10, pp. 1-11, 1968.

[78] P. Langley, Data-Driven Discovery of Physical Laws, *Cognitive Science*, vol. 5, 1981.

[79] P. Langley *Elements of Machine Learning*, Morgan Kaufman Publishers, inc., 1996.

[80] L. Lesmo, L. Saitta, and P. Torasso, Learning of Fuzzy Production Rules for Medical Diagnosis, in: *Approximate Reasoning in Decision Analysis*, M.M. Gupta and E. Sanchez (eds.), pp. 249 - 259, North Holland Publishing Company, 1982.

[81] P. Liang and F. Song, What does a Probabilistic Interpretation of Fuzzy Sets Mean?, *IEEE Trans. on Fuzzy Systems*, vol. 4, pp. 200-205, 1996.

[82] J. Locke, *An Essay Concerning Human Understanding*, originally appeared in 1690, Abridged and Edited by J.W. Yolton, Everyman, 1995 .

[83] M. Loeve, *Probability Theory*, 3rd ed., Van Nostrand Reinhold Comp., 1963.

[84] D.O. Loftsgaarden and C.P. Quesenberry, A Nonparametric estimate of a Multivariate Density Function, in: *The Annals of Mathematical Statistics*, vol. 36, pp. 1049-1051, 1965.

[85] B.A. Macdonald and I.H. Witten, A Framework for Knowledge Acquisition through Techniques of Concept Learning, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 3, pp. 499 - 512, 1989.

[86] R.J. Marks II, *Introduction to Shannon Sampling and Interpolation Theory*, New York: Springer-Verlag Inc., 1991.

[87] J.S. Marron and D. Nolan, Canonical Kernels for Density Estimation, *Statistics And Probability Letters*, vol. 7, pp. 195-199, 1988.

[88] R.S. Michalski, On the Quasi-minimal Solution of the General Covering Problem, *Proc. of the 5th International Symposium on Information Processing*, pp. 125- 128, Bled, Yugoslavia, 1969.

[89] R.S. Michalski and J.B. Larson, Selection of the most representative training examples and incremental generation of VL1 hypotheses: the underlying method and description of programs ESEL and AQ11, Report 867, Computer Science Department, University of Illinois, 1978.

[90] R.S. Michalski, Pattern Recognition as rule-guided inductive inference, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, no. 4, pp. 341 -361, 1980.

[91] R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds.), *Machine Learning; an Artificial Intelligence approach*, vol. 1, Morgen Kaufmann, 1983.

[92] R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds.), *Machine Learning; an Artificial Intelligence approach*, vol. 2, Morgen Kaufmann, 1986.

[93] R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds.), Understanding the Nature of Learning, in: *Machine Learning; an Artificial Intelligence approach*, R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds.), vol. 2, Morgen Kaufmann, 1986.

[94] R.S. Michalski, I. Mozetic, J. Hong, and N. Lavrac, The multi-purpose incremental learning system AQ15 and its testing application to three medical domains, *Proceedings 5th National Conference on AI*, pp. 1041 - 1045, Morgan Kaufman, 1986.

[95] R.S. Michalski, I. Bratko, and M. Kubat, *Machine Learning and Data Mining: Methods and Applications*, Wiley, 1998.

[96] D. Michie, D.J. Spiegelhalter, and C.C. Taylor (eds.) *Machine Learning, Neural and Statistical Classification*, Ellis Horwood Limited, 1996.

[97] T.M. Mitchell, Version spaces: A candidate elimination approach to rule learning. *Proc. of the fifth Joint Conference on Artificial Intelligence*, pp. 305-310, Cambridge, MA, Morgan Kaufmann, 1977.

[98] T.M. Mitchell, Generalization as search, *Artificial Intelligence*, vol. 18, pp. 203-226, 1982.

[99] K. Morik, S. Wrobel, J.-U. Kietz, and W. Emde, *Knowledge Acquisition and Machine Learning: theory methods and applications*, Academic Press, London, 1993.

[100] V.K. Murthy, Nonparametric estimation of multivariate densities with applications, in: *Multivariate Analysis*, P.R. Krishnaiah (Ed.), New York: Academic, pp. 43-48, 1966.

[101] W. Muller and F. Wysotzki, The Decision Tree Algorithm CAL5 Based on a Statistical Approach to its Splitting Algorithm, in: *Machine Learning and Statistics; the interface* G. Nakhaeizadeh and C.C. Taylor (eds.), John Wiley & Sons, Inc., 1997.

[102] G. Nakhaeizadeh and C.C. Taylor (eds.) *Machine Learning and Statistics; the interface* John Wiley & Sons, Inc., 1997.

[103] K. Nozaki, H. Ishibuchi, and H. Tanaka, Adaptive Fuzzy Rule-Based Classification Systems, *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 238 - 250, 1996.

[104] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tesselations: Concept and Applications of the Voronoi Diagram*, John Wiley and Sons, New York, 1992.

[105] T. Okuda, H. Tanaka, and K. Asai, A Formulation of Fuzzy Decision Problems with Fuzzy Information using Probability Measures of Fuzzy Events, *Information and Control*, vol. 38, pp. 135-147, 1978.

[106] A. Papoulis, *The Fourier Integral And its Applications*, McGraw-Hill Book Company Inc., 1962.

[107] E. Parzen, On estimation of a probability density-function and mode, *Ann. Math. Statistics*, vol. 33, pp. 1065-1076, 1962.

[108] E. Parzen, *Modern Probability Theory and its Applications*, New York, John Wiley, 1960.

[109] C.S. Peirce, *Philosophical Writings Of Peirce*, Several papers on probability appeared since 1878 (ch. 12-14), Selected and Edited by Justus Buchler, new Dover edition, Dover Publications Inc., 1955.

[110] I. Pitas, E. Milos, and A.N. Venetsanopoulos, An Minimum Entropy Approach to Rule Learning from Examples, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 4, pp. 621 - 635, 1992.

[111] K. Popper, *Objective Knowledge*, Oxford, 1973.

[112] J.R. Quinlan, Discovering rules by induction from large collections of examples, in: *Expert Systems in the micro-electronic age*, D. Michie (Ed.), Edinburgh University Press, 1979.

[113] J.R. Quinlan, Induction of Decision Trees, *Machine Learning*, vol. I, pp. 81-106, 1986.

[114] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan-Kaufmann, 1993.

[115] S.J. Raudys and A.K. Jain, Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 252-264, 1991.

[116] R. Reinke and R. Michalski, Incremental Learning of concept descriptions; a method and experimental results in: *Machine Intelligence 11, Logic and the Acquisition of Knowledge*, J. Hayes, D. Michie, and J. Richards (eds.), pp. 263-288, Clarendon Press, 1988.

[117] T.L. Saaty, Measuring the Fuzziness of Sets, *J. of Cybernetics*, vol. 4, no. 4, pp. 53 - 61, 1974.

[118] S.R. Sain and D.W. Scott, On Locally Adaptive Density Estimation, *Journal of the American Statistical Association*, vol. 91, no. 436, pp. 1525 - 1534, 1996.

[119] J.C. Schlimmer and R.H. Granger JR., Incremental Learning from Noisy Data, *Machine Learning*, vol. I, pp. 317 - 354, 1986.

[120] D.W. Scott, Average Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions, *The Annals of Statistics*, vol. 13, pp. 1024 - 1040, 1985.

[121] G. Shafer, *A Mathematical Theory of Evidence*, Princeton U.P., 1976.

[122] J. Shavlik and T. Dieterrich, General Aspects of Machine Learning, in: *Readings in Machine learning*, J. Shavlik and T. Dietterich (eds), pp. 1 -10, 1990.

[123] Z. Shi, *Principles of Machine Learning*, International Academic Publishers, 1992.

[124] D. C. Sills, William of Ockham, in: *International Enceclopedia of the Social Sciences*, pp. 269-270, Macmillian Company & The Free Press, New York, 1968.

[125] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, London, Chapman and Hall, 1986.

[126] H. van der Sluijs, *Design and Evaluation of an Alarm System for the Intelligent Anesthesia Monitor*, M.Sc. thesis, Delft University of Technology, Information Theory group, the Netherlands, 1996.

[127] P. Smets, Probability of a fuzzy event, an axiomatic approach, *Fuzzy sets and Systems*, vol. 7, pp. 153-164, 1982.

[128] P. Smyth and R.M. Goodman, Rule Induction using Information Theory, in: *Knowledge Discovery in Databases*, G. Piatetsky and W. Frawley (eds.), MIT Press, Cambridge MA, 1990.

[129] P. Smyth and R.M. Goodman, An Information-Theoretic Approach to Rule Induction from Databases, *IEEE Transactions on Knowledge and Data Engineering*, vol. 4, no. 4, pp. 301 - 316, 1992.

[130] T. Sudkamp and R.J. Hammell II, Interpolation, Completion and Learning Fuzzy Rules, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 4, pp. 621 - 635, 1992.

[131] E. Trianphyllou and S.H. Mann, An Evaluation of the Eigenvalue Approach for Determining the Membership Values in Fuzzy Sets, *Fuzzy Sets and Systems*, vol. 35, pp. 295 - 301, 1990.

[132] I.B. Turksen and A.M. Norwich, A Model for the Measurement of Membership and the consequences of its emperical Implementation, *Fuzzy Sets and Systems*, vol. 25, pp. 1 - 25, 1984.

[133] I.B. Turksen, Measurement of Memberships and their Acquisition, *Fuzzy Sets and Systems*, vol. 40, pp. 5 - 38, 1991.

[134] P. Utgoff, Incremental induction of decision trees, *Machine Learning Journal*, vol. 4, pp. 161 - 186, 1989.

[135] J.C.A. van der Lubbe, *Information Theory*, Cambridge University Press, 1997.

[136] V. Vapnik, *The Nature of Statistical Learning*, Springer-Verlag, New York, 1995.

[137] H.J.L.M. Vullings, *Biomedical Waveform Validation*, Ph.D. Thesis, Delft University of Technology, the Netherlands, 1999.

[138] L. Wang and J.M. Mendel, Generating Fuzzy Rules by Learning from Examples, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 4, pp. 1414 - 1427, 1992.

[139] Z. Wang and G.J. Klir, *Fuzzy Measure Theory*, Plenum Press., New York, 1992.

[140] M. West, Approximating posterior distributions by mixtures, *J. Royal Statistics Soc. B*, vol. 55, no. 2, pp. 409 - 422, 1993.

[141] R.R Yager, A Note on Probabilities of Fuzzy Events, *Information Sciences*, vol. 18, pp. 113-129, 1979.

[142] R.R Yager, On a general class of fuzzy connectives, *Fuzzy Sets and Systems*, vol. 4, no. 3, pp. 235-242, 1980.

[143] R.R Yager, Fuzzy Sets, Probabilities and Decison, *Journal of Cybernetics*, vol. 10, pp. 1-18, 1980.

[144] R.R Yager, Generalizied Probabilities of Fuzzy Events from Fuzzy Belief Structures, *Information Sciences*, vol. 28, pp. 45-62, 1982.

[145] R.R Yager, Probabilities from Fuzzy Observations, *Information Sciences*, vol. 32, pp. 1-31, 1984.

[146] L.A. Zadeh, Fuzzy Sets, *Information and Control*, vol. 8, pp. 338-353, 1965.

[147] L.A. Zadeh, Probability Measures of Fuzzy Events, *Journal of Mathemetical Ann. and Appl.*, vol. 23, pp. 421-427, 1968.

[148] L.A. Zadeh, Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, *IEEE Trans. on Systems, Man and Cybernetics*, vol. 3, no. 1, pp. 28 - 44, 1973.

[149] L.A. Zadeh, PRUF-A meaning representation for natural languages, *Internat. J. of Man-Machine Studies*, vol. 10, no. 1, pp. 395 - 460, 1978.

Appendix A

Proof of Convergence

In this section it is proven for the one-dimensional case that the uniform DK estimator converges to a convolution with the unknown pdf given an unlimited number of samples and data. Before turning to the proof itself, definitions and theorems are given which will be used for the proof.

Parzen Windows definition

The Parzen Windows (PW) estimator is defined for a set I of examples x_i :

$$\hat{p}(x) = \frac{1}{V_p n} \sum_{i=1}^{n} \phi_p(x - x_i) \tag{A.1}$$

$$V_p = \int \phi_p(x) \tag{A.2}$$

such that:

$$\begin{aligned} \int \hat{p}(x)dx &= \int (\frac{1}{V_p n} \sum_{i=1}^{n} \phi_p(x - x_i))dx \\ &= \frac{1}{V_p n} \sum_{i=1}^{n} (\int \phi_p(x - x_i)dx) \\ &= \frac{V_p n}{V_p n} = 1 \end{aligned} \tag{A.3}$$

Convolution theorem

$$\lim_{n \to \infty} \hat{p}(x) = \frac{1}{V_p} \int \phi_p(x - x_i) p(x_i) dx_i \tag{A.4}$$

The proof that PW in the limit of $N \to \infty$ approaches to a convolution with the real pdf, is completely given in [31] using *convergence in mean square.*

Uniform DK estimator definition

The uniform DK estimator can be defined as:

$$\hat{\hat{p}}(x) = \frac{1}{V_m m'_{n,m}} \sum_{s=1}^{m} \phi_m(x - x_s)\hat{c}_n(x_s) \tag{A.5}$$

where V_m is the volume of ϕ_m, and with $m'_{n,m}$ defined as:

$$m'_{n,m} = \sum_{s=1}^{m} \hat{c}_n(x_s) = \sum_{s=1}^{m} V_p n \hat{\hat{p}}(x_s) \tag{A.6}$$

Furthermore $S = \{x_1, x_2, ..., x_m\}$ is an equidistant scatter-field in an interval $[a, b]$ (sampling) such that:

$$a < x_1 < x_2 < ... < x_m < b \tag{A.7}$$

and where the granularity (sampling distance) Δ_S is defined as:

$$\Delta_i = \max_i\{(x_{i+1} - x_i)\} = \Delta_S = \frac{b-a}{m+1} > 0 \quad \forall x_i \in \Sigma \tag{A.8}$$

The Riemann-integral definition

Let $\Pi = \{x'_0, x'_1, ..., x'_m\}$ be a partition of an interval $[a, b]$ such that:

$$a = x'_0 < x'_1 < ... < x'_m = b \tag{A.9}$$

Further, the granularity of Π is defined as: $\mu(\Pi) = \max_i\{(x'_i - x'_{i-1})\}$
Let $S = \{x_1, x_2, ..., x_m\}$ be a scatter-field such that:

$$x'_{i-1} < x_i < x'_i \quad \forall x_i \in S \tag{A.10}$$

The (Riemann-)integral for a function f over interval $[a, b]$ is now defined as:

$$\int_a^b f(x)dx = \lim_{\mu(\Pi)\to 0} \sum_{i=1}^{m} f(x_i)(x'_i - x'_{i-1}) \tag{A.11}$$

This definition can be found in [3].

The limit-product theorem

if L and M are real values, and

$$\begin{aligned} \lim_{x\to a} f(x) &= L \\ \lim_{x\to a} g(x) &= M \end{aligned} \tag{A.12}$$

then

$$\lim_{x\to a}(f(x)g(x)) = LM \tag{A.13}$$

for any a. Proof is given by [3].

The limit-division theorem

if L and M are real values and $M > 0$,

$$\begin{aligned} \lim_{x \to a} f(x) &= L \\ \lim_{x \to a} g(x) &= M \end{aligned} \tag{A.14}$$

then

$$\lim_{x \to a} \frac{f(x)}{g(x)} = \frac{L}{M} \tag{A.15}$$

for any a. Proof is given by [3].

TO BE PROVEN:

$$\lim_{n,m \to \infty} \hat{\hat{p}}(x) = \frac{1}{V_p V_m} \int \left(\phi_m(x - x_s) \left[\int \phi_p(x_s - x_i) p(x_i) dx_i \right] \right) dx_s \tag{A.16}$$

IF $m \to \infty$ such that $\Delta_S = (\infty - (-\infty))/(m+1) \to 0$.

PROOF:

To proof that the uniform DK estimator converges to a double convolution with the real pdf, we first proof that for $m \to \infty$ equation (A.5) converges to a convolution with the estimator $\hat{p}(x)$.

Since S is an equidistant scatter-field on interval $[a, b]$, there exists an equidistant partition $\Pi = \{x'_0, x'_1, ..., x'_m\}$ on interval $[a, b]$ such that:

$$a = x'_0 < x'_1 < ... < x'_m = b \tag{A.17}$$

where the granularity Π is:

$$\mu(\Pi) = \max_i \{(x'_i - x'_{i-1})\} = (x'_i - x'_{i-1}) = \Delta_S \tag{A.18}$$

Multiplying and dividing with the granularity Δ_S of partition Π (from (A.8) $\Delta_S > 0$) gives for (A.5):

$$\hat{\hat{p}}(x) = \frac{\Delta_S}{V_m m'_{n,m} \Delta_S} \sum_{s=1}^{m} \phi_m(x - x_s) \hat{c}_n(x_s) \tag{A.19}$$

Observing that Δ_S depends on m but not on s we can bring Δ_S under the summation:

$$\hat{\hat{p}}(x) = \frac{1}{V_m m'_{n,m} \Delta_S} \sum_{s=1}^{m} \phi_m(x - x_s) \hat{c}_n(x_s) \Delta_S \tag{A.20}$$

which can be written as a product of two functions $f(\Delta_S)$, and $g(\Delta_S)$:

$$\hat{p}(x) = f(\Delta_S)g(\Delta_S) \tag{A.21}$$

$$f(\Delta_S) = \frac{1}{V_m m'_{n,m} \Delta_S} \tag{A.22}$$

$$g(\Delta_S) = \sum_{s=1}^{m} \phi_m(x - x_s)\hat{c}_n(x_s)\Delta_S \tag{A.23}$$

If we now take the limit of $\Delta_S \to 0$ for both functions f and g, and show that these limits exist, then the *limit-product* theorem provides us the answer for $\hat{\hat{p}}(x)$.

We will now first proceed with $f(\Delta_S)$. Using (A.6):

$$\lim_{\Delta_S \to 0} f(\Delta_S) = \lim_{\Delta_S \to 0} \frac{1}{V_m m'_{n,m} \Delta_S} \tag{A.24}$$

$$= \lim_{\Delta_S \to 0} \frac{1}{V_m \Delta_S \sum_{s=1}^{m} V_p n \hat{p}(x_s)} \tag{A.25}$$

again taking Δ_S under the summation:

$$\lim_{\Delta_S \to 0} f(\Delta_S) = \lim_{\Delta_S \to 0} \frac{1}{V_m \sum_{s=1}^{m} V_p n \hat{p}(x_s) \Delta_S} \tag{A.26}$$

using the *limit-division* theorem

$$\lim_{\Delta_S \to 0} f(\Delta_S) = \frac{1}{\lim_{\Delta_S \to 0} \{V_m \sum_{s=1}^{S} V_p n \hat{p}(x_s) \Delta_S\}} \tag{A.27}$$

Using the *Riemann-integral* definition for the denumerator:

$$\lim_{\Delta_S \to 0} f(\Delta_S) = \frac{1}{V_m \int_a^b V_p n \hat{p}(x_s) dx_s} \tag{A.28}$$

observing that the integral of the estimator $\hat{p}(x_s)$ is equal to 1 (see (A.3)), the limit converges to:

$$\lim_{\Delta_S \to 0} f(\Delta_S) = \frac{1}{V_m V_p n} \tag{A.29}$$

Hence, the limit for $f(\Delta_S)$ exists, and is real if (1) the kernel volumes are not zero, and (2) if the number of data n is more then zero.

Turning to $g(\Delta_S)$:

$$\lim_{\Delta_S \to 0} g(\Delta_S) = \lim_{\Delta_S \to 0} \sum_{s=1}^{m} \phi_m(x - x_s)\hat{c}_n(x_s)\Delta_S \tag{A.30}$$

Since (A.6) $\hat{c}_n(x) = V_p n \hat{p}(x)$, we can write for $g(\Delta_S)$:

$$\lim_{\Delta_S \to 0} g(\Delta_S) = \lim_{\Delta_S \to 0} V_p n \sum_{s=1}^{m} \phi_m(x - x_s)\hat{p}(x_s)\Delta_S \tag{A.31}$$

Using the *Riemann-integral* definition:

$$\lim_{\Delta_S \to 0} g(\Delta_S) = V_p n \int_a^b \phi_m(x - x_s)\hat{p}(x_s)dx_s \tag{A.32}$$

Hence, the limit for $g(\Delta_S)$ exists and is real.

Using (A.23), and substituting (A.29) and (A.32) (according to the *limit-product* theorem), we may write:

$$\begin{aligned} \lim_{\Delta_S \to 0} \hat{\hat{p}}(x) &= \lim_{\Delta_S \to 0} f(\Delta_S)g(\Delta_S) \\ &= \frac{1}{V_m V_p n} V_p n \int_a^b \phi_m(x - x_s)\hat{p}(x_s)dx_s \\ &= \frac{1}{V_m} \int_a^b \phi_m(x - x_s)\hat{p}(x_s)dx_s \end{aligned} \tag{A.33}$$

Hence, we obtain a convolution if $\Delta_S \to 0$. However, since the partition granularity depends on the sampling distance, requiring that $\Delta_S \to 0$ is equivalent to requiring $S \to \infty$ on a finite interval $[a, b]$. Therefore:

$$\lim_{S \to \infty} \hat{\hat{p}}(x) = \frac{1}{V_m} \int_a^b \phi_m(x - x_s)\hat{p}(x_s)dx_s \tag{A.34}$$

Taking the limit for $N \to \infty$ and substituting (A.4), it follows that:

$$\lim_{N,S \to \infty} \hat{\hat{p}}(x) = \frac{1}{V_m V_p} \int_a^b \left(\phi_m(x - x_s) \left[\int \phi_p(x_s - x_i)p(x_i)dx_i \right] \right) dx_s \tag{A.35}$$

Finally, extending $[a, b]$ such that it becomes infinitely large:

$$\lim_{n,m \to \infty} \hat{\hat{p}}(x) = \frac{1}{V_m V_p} \int \left(\phi_m(x - x_s) \left[\int \phi_p(x_s - x_i)p(x_i)dx_i \right] \right) dx_s \tag{A.36}$$

however, for our sampling we must now require that $m \to \infty$ such that $\Delta_S = (\infty - (-\infty))/(m + 1) \to 0$ as a condition for the first convolution.
Q.E.D.

Appendix B

A Generalized Class of DK estimators

In order to construct an even more general class of DK estimators, we need to define a lattice of kernels on some bins positioned in the feature space. Using these lattices it is possible to describe kernels that are locally different, such that the fixed-bandwidth problem can be overcome. In this section we only show that these estimators *can* be designed within the DK framework, we do not show *how* they are designed.

Lattice definition

A lattice $M_{C,B}$ consists of b types of kernel functions μ_j which are defined on a set B of b bins β_j, and positioned according to a set C of centers $\mathbf{x}_c$. The only condition that we place on the kernels is that they be squared-integrable. The lattice is defined as:

$$M_{C,B}(\mathbf{x}-\mathbf{x}_c) = \sum_{j=1}^{b} \mu_j(\frac{\mathbf{x}-\mathbf{x}_c}{H_{\mu_j}})\beta_j(\frac{\mathbf{y}(\mathbf{x},\mathbf{x}_c,\mathbf{x}_j)}{H_{\beta_j}}) \tag{B.1}$$

Here, $\beta_j(\mathbf{y})$ is a multidimensional (sample-)bin (not necessarily rectangular) for which holds:

$$\begin{aligned} \beta_j(\frac{\mathbf{y}}{H_{\beta_j}}) &= \begin{cases} 1 & for \quad |\frac{(\mathbf{y})}{H_{\beta_j}}| <= 1 \\ 0 & otherwise \end{cases} \\ \mathbf{y} &= \mathbf{y}(\mathbf{x},\mathbf{x}_c,\mathbf{x}_j) \end{aligned} \tag{B.2}$$

Where $\beta_j(\mathbf{y})$ has the following properties:

$$\begin{aligned} \textstyle\sum_{j=1}^{b} \beta_j(\tfrac{\mathbf{y}}{H_{\beta_j}}) &= 1 \\ \beta_{j,k} = \beta_j(\tfrac{\mathbf{y}}{H_{\beta_j}})\beta_k(\tfrac{\mathbf{y}}{H_{\beta_k}}) &= \begin{cases} 1 & for \quad j = k \\ 0 & otherwise \end{cases} \end{aligned} \tag{B.3}$$

where H_{β_j} is the bin-width for β_j. Such a lattice is given in figure B.1.

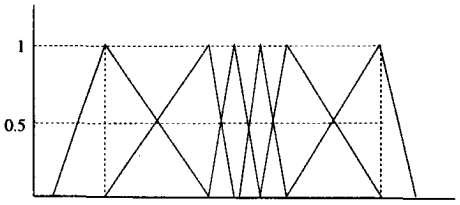


Figure B.1: *A lattice of triangular kernels where* $\mathbf{y} = \mathbf{y}(\mathbf{x}_s)$.

Generalized DK definition

Using the lattice definition, a generalized DK estimator may be written as:

$$\begin{aligned} \hat{\hat{p}}(\mathbf{x}) &= \frac{1}{m'} \sum_{s=1}^{m} \frac{N_{S,\Delta}(\mathbf{x} - \mathbf{x}_s)}{V_N(\mathbf{x} - \mathbf{x}_s)} \sum_{i=1}^{n} M_{I,B}(\mathbf{x}_i - \mathbf{x}_s) \\ &= \frac{1}{m'} \sum_{s=1}^{m} \sum_{i=1}^{n} \frac{N_{S,\Delta}(\mathbf{x} - \mathbf{x}_s)}{V_N(\mathbf{x} - \mathbf{x}_s)} M_{S,B}(\mathbf{x}_i - \mathbf{x}_s) \end{aligned} \tag{B.4}$$

where m' is a normalizing constant, and where $V_N(\mathbf{x} - \mathbf{x}_s)$ is a generalized volume:

$$\begin{aligned} m' &= \sum_{s=1}^{m} \sum_{i=1}^{n} M_{S,B}(\mathbf{x}_i - \mathbf{x}_s) \\ V_N(\mathbf{x} - \mathbf{x}_s) &= \int N_{S,\Delta}(\mathbf{x} - \mathbf{x}_s) d\mathbf{x} \end{aligned} \tag{B.5}$$

such that the estimator is a density function:

$$\begin{aligned} \int \hat{\hat{p}}(\mathbf{x}) d\mathbf{x} &= \int \frac{1}{m'} \sum_{s=1}^{m} \sum_{i=1}^{n} \frac{N_{S,\Delta}(\mathbf{x} - \mathbf{x}_s)}{V_N(\mathbf{x} - \mathbf{x}_s)} M_{S,B}(\mathbf{x}_i - \mathbf{x}_s) d\mathbf{x} \\ &= \frac{1}{m'} \sum_{s=1}^{m} \sum_{i=1}^{n} \int \frac{N_{S,\Delta}(\mathbf{x} - \mathbf{x}_s)}{V_N(\mathbf{x} - \mathbf{x}_s)} d\mathbf{x} M_{S,B}(\mathbf{x}_i - \mathbf{x}_s) \\ &= \frac{m'}{m'} = 1 \end{aligned} \tag{B.6}$$

Mean of the generalized DK estimator

In order to determine the mean of the generalized DK estimator we assume that the m samples of set S are distributed according to a density $p_S(\mathbf{x})$ which must be independent of $\mathbf{x}_i$. The expectation of the DK estimator (the mean) is calculated with respect to random variable $\mathbf{x}_i$, distributed according to the unknown density $p(\mathbf{x})$, and random variable $\mathbf{x}_s$ distributed according to $p_S(\mathbf{x})$. Then:

$$\begin{aligned} E[\hat{\bar{p}}(\mathbf{x})] &= \bar{\bar{p}}(\mathbf{x}) \\ &= E_{(\mathbf{x}_s,\mathbf{x}_i)}\left[\frac{1}{m'}\sum_{s=1}^{m}\sum_{i=1}^{n}\frac{N_{S,\Delta}(\mathbf{x}-\mathbf{x}_s)}{V_N(\mathbf{x}-\mathbf{x}_s)}M_{I,B}(\mathbf{x}_s-\mathbf{x}_i)\right] \\ &= \frac{1}{m'}\sum_{s=1}^{m}\sum_{i=1}^{n}E_{(\mathbf{x}_s,\mathbf{x}_i)}\left[\frac{N_{S,\Delta}(\mathbf{x}-\mathbf{x}_s)}{V_N(\mathbf{x}-\mathbf{x}_s)}M_{I,B}(\mathbf{x}_s-\mathbf{x}_i)\right] \end{aligned} \quad \text{(B.7)}$$

substituting $\mathbf{v}$ for $\mathbf{x}_s$ and $\mathbf{w}$ for $\mathbf{x}_i$, and noting that $p(\mathbf{v},\mathbf{w}) = p_S(\mathbf{v})p(\mathbf{w})$:

$$\begin{aligned} \bar{\bar{p}}(\mathbf{x}) &= \frac{1}{m'}\sum_{s=1}^{m}\sum_{i=1}^{n}\int\int\frac{N_{S,\Delta}(\mathbf{x}-\mathbf{v})}{V_N(\mathbf{x}-\mathbf{v})}M_{I,B}(\mathbf{v}-\mathbf{w})p_S(\mathbf{v})p(\mathbf{w})d\mathbf{w}d\mathbf{v} \\ &= \frac{mn}{m'}\int\frac{N_{S,\Delta}(\mathbf{x}-\mathbf{v})}{V_N(\mathbf{x}-\mathbf{v})}\left[\int M_{I,B}(\mathbf{v}-\mathbf{w})p(\mathbf{w})d\mathbf{w}\right]p_S(\mathbf{v})d\mathbf{v} \end{aligned} \quad \text{(B.8)}$$

using $*$ to denote a convolution, this can be written as:

$$\begin{aligned} \bar{\bar{p}}(\mathbf{x}) &= \frac{mn}{m'}\int\frac{N_{S,\Delta}(\mathbf{x}-\mathbf{v})}{V_N(\mathbf{x}-\mathbf{v})}\left[M_{I,B}(\mathbf{v})*p(\mathbf{v})\right]p_S(\mathbf{v})d\mathbf{v} \\ &= \frac{mn}{m'}\frac{N_{S,\Delta}(\mathbf{x})}{V_N(\mathbf{x})}*\{[M_{I,B}(\mathbf{x})*p(\mathbf{x})]p_S(\mathbf{x})\} \end{aligned} \quad \text{(B.9)}$$

The estimator should then be designed such that its expectation integrates to one:

$$\begin{aligned} \int\bar{\bar{p}}(\mathbf{x})d\mathbf{x} &= \int\frac{mn}{m'}\frac{N_{S,\Delta}(\mathbf{x})}{V_N(\mathbf{x})}*\{[M_{I,B}(\mathbf{x})*p(\mathbf{x})]p_S(\mathbf{x})\}d\mathbf{x} \\ &= \int\frac{N_{S,\Delta}(\mathbf{x})}{V_N(\mathbf{x})}d\mathbf{x}\int\frac{mn}{m'}[M_{I,B}(\mathbf{x})*p(\mathbf{x})]p_S(\mathbf{x})d\mathbf{x} \\ &= \int\frac{mn}{m'}[M_{I,B}(\mathbf{x})*p(\mathbf{x})]p_S(\mathbf{x})d\mathbf{x} \end{aligned} \quad \text{(B.10)}$$

hence the design criterion for a normalized expectation becomes:

$$\int\frac{mn}{m'}[M_{I,B}(\mathbf{x})*p(\mathbf{x})]p_S(\mathbf{x})d\mathbf{x} = 1 \quad \text{(B.11)}$$

Variance of the generalized DK estimator

Since the generalized DK estimator is a sum of functions of statistically independent random variables $\mathbf{x}_i$, its variance is equal to the sum of variances of the terms:

$$\begin{aligned} Var[\hat{\hat{p}}(\mathbf{x})] = \\ &= \sum_{i=1}^{n} E_{(\mathbf{x}_s,\mathbf{x}_i)} \left[\left(\frac{1}{m'} \sum_{s=1}^{m} \frac{N_{S,\Delta}(\mathbf{x}-\mathbf{x}_s)}{V_N(\mathbf{x}-\mathbf{x}_s)} M_{I,B}(\mathbf{x}_s-\mathbf{x}_i) \right)^2 - \frac{1}{n^2}\bar{\bar{p}}^2(\mathbf{x}) \right] \\ &= nE_{(\mathbf{x}_s,\mathbf{x}_i)} \left[T \left(\frac{1}{m'} \sum_{s=1}^{m} \frac{N_{S,\Delta}(\mathbf{x}-\mathbf{x}_s)}{V_N(\mathbf{x}-\mathbf{x}_s)} M_{I,B}(\mathbf{x}_s-\mathbf{x}_i) \right) - \frac{1}{n^2}\bar{\bar{p}}^2(\mathbf{x}) \right] \end{aligned} \tag{B.12}$$

bounding T, and substituting $\mathbf{w}$ and $\mathbf{v}$ gives:

$$\begin{aligned} Var[\hat{\hat{p}}(\mathbf{x})] = \\ &= \sup\{T\} \int\int \frac{mn}{m'} \frac{N_{S,\Delta}(\mathbf{x}-\mathbf{v})}{V_N(\mathbf{x}-\mathbf{v})} M_{I,B}(\mathbf{v}-\mathbf{w}) p(\mathbf{w}) p_S(\mathbf{v}) d\mathbf{w} d\mathbf{v} - \frac{1}{n}\bar{\bar{p}}^2(\mathbf{x}) \end{aligned} \tag{B.13}$$

dropping the second term and using (B.8) finally yields for the variance:

$$Var[\hat{\hat{p}}(\mathbf{x})] \leq \sup\{T\}\bar{\bar{p}}(\mathbf{x}) \tag{B.14}$$

To obtain a variance comparable to the Parzen Windows estimator we must design T such that:

$$\sup\{T\} = \frac{1}{nV_N(\mathbf{x}-\mathbf{x}_s)} \tag{B.15}$$

Convergence in mean-square

For convergence in mean-square it is necessary to prove that the mean and variance convergence according to specified criteria:

$$\begin{aligned} E[\hat{\hat{p}}(\mathbf{x})] &\rightarrow p(\mathbf{x}) \\ Var[\hat{\hat{p}}(\mathbf{x})] &\rightarrow 0 \end{aligned} \tag{B.16}$$

Example

As an example, consider the lattice given in figure B.1. Then M can be written as:

$$M_{S,B}(\mathbf{x}-\mathbf{x}_s) = \sum_{j=1}^{b} \mu_j \left(\frac{\mathbf{x}-\mathbf{x}_s}{H_{\mu_j}}\right) \beta_j \left(\frac{\mathbf{x}_s-\mathbf{x}_j}{H_{\mu_j}}\right) \tag{B.17}$$

where the non-symmetrical bins β_j are centered around the center positions of the kernels, such that each $\mathbf{x}_s$ corresponds to a single bin center (clearly the number of bins b equals the number of samples m). Since the sum of kernels is equal to 1, the normalization constant m' becomes equal to n. If we choose N equal to M we get for the volume $V_N(\mathbf{x}-\mathbf{x}_s)$:

$$\begin{aligned}
V_N(\mathbf{x}-\mathbf{x}_s) &= \int N_{S,B}(\mathbf{x}-\mathbf{x}_s)d\mathbf{x} \\
&= \int M_{S,B}(\mathbf{x}-\mathbf{x}_s)d\mathbf{x} \\
&= \int \sum_{j=1}^{b} \mu_j(\frac{\mathbf{x}-\mathbf{x}_s}{H_{\mu_j}})\beta_j(\frac{\mathbf{x}_s-\mathbf{x}_j}{H_{\mu_j}})d\mathbf{x} \\
&= \sum_{j=1}^{b} \int \mu_j(\frac{\mathbf{x}-\mathbf{x}_s}{H_{\mu_j}})d\mathbf{x}\beta_j(\frac{\mathbf{x}_s-\mathbf{x}_j}{H_{\mu_j}}) \\
&= \sum_{j=1}^{b} \frac{1}{V_{\mu_j}|H_{\mu_j}|}\beta_j(\frac{\mathbf{x}_s-\mathbf{x}_j}{H_{\mu_j}}) \\
&= \sum_{j=1}^{b} \frac{1}{|H_{\mu_j}|}\beta_j(\frac{\mathbf{x}_s-\mathbf{x}_j}{H_{\mu_j}}) \qquad \text{(B.18)}
\end{aligned}$$

Where the last equality comes from observing that for triangular kernels $V_{\mu_j} = 1$. Due to the choice of the sample-bins β_j,p_S can be considered locally uniform, and can be written as:

$$p_S(\mathbf{x}) = \frac{1}{m}\sum_{j=1}^{b} \frac{1}{|H_{\mu_j}|}\beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}}) \qquad \text{(B.19)}$$

such that the design criterion (B.11) for a normalized expectation becomes:

$$\begin{aligned}
&\int \frac{mn}{m'}[M_{I,B}(\mathbf{x}) * p(\mathbf{x})]p_S(\mathbf{x})d\mathbf{x} = \\
&= \int \frac{mn}{n} \int \sum_{j=1}^{b} \mu_j(\frac{\mathbf{w}-\mathbf{x}}{H_{\mu_j}})\beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}})p(\mathbf{w})d\mathbf{w}\frac{1}{m}\sum_{j=1}^{b} \frac{1}{|H_{\mu_j}|}\beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}})d\mathbf{x} \\
&= \int \int \sum_{j=1}^{b} \frac{\mu_j(\frac{\mathbf{w}-\mathbf{x}}{H_{\mu_j}})}{|H_{\mu_j}|}\beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}})p(\mathbf{w})d\mathbf{w}d\mathbf{x} \\
&= \int \sum_{j=1}^{b} \int \frac{\mu_j(\frac{\mathbf{w}-\mathbf{x}}{H_{\mu_j}})}{|H_{\mu_j}|}p(\mathbf{w})d\mathbf{w}\beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}})d\mathbf{x} \\
&= \int \sum_{j=1}^{b} \frac{\mu_j(\frac{-\mathbf{x}}{H_{\mu_j}})}{|H_{\mu_j}|} * p(\mathbf{x})\beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}})d\mathbf{x}
\end{aligned}$$

(B.20)

where we used (B.3). Suppose each kernel $\mu_j(\frac{\mathbf{w}-\mathbf{x}}{H_{\mu_j}})$ (positioned around $\mathbf{x}$ near $\mathbf{x}_j$!) is a perfect filter for $p(\mathbf{x})$ in the range defined by the bin $\beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}})$ then each of the convolutions multiplied by the bin returns a part of $p(\mathbf{x})$. The summation of all these parts is then simply $p(\mathbf{x})$ itself. Hence, in case of perfect *local* filtering the previous can be written as:

$$\begin{aligned}\int \sum_{j=1}^{b} \frac{\mu_j(\frac{-\mathbf{x}}{H_{\mu_j}})}{|H_{\mu_j}|} * p(\mathbf{x})\beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}})d\mathbf{x} &= \int \sum_{j=1}^{b} p(\mathbf{x})\beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}})d\mathbf{x} \\ &= \int p(\mathbf{x}) \sum_{j=1}^{b} \beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}})d\mathbf{x} \\ &= \int p(\mathbf{x})d\mathbf{x} \qquad \text{(B.21)}\end{aligned}$$

where we used that the sum of the bins equals one. So, from analysis of the expectation we get as a condition that:

$$\frac{\mu_j(\frac{-\mathbf{x}}{H_{\mu_j}})}{|H_{\mu_j}|} * p(\mathbf{x}) = p(\mathbf{x}) \ \forall\{\mathbf{x}|\beta_j(\frac{\mathbf{x}-\mathbf{x}_j}{H_{\mu_j}}) \geq 0\} \qquad \text{(B.22)}$$

The normalization then becomes:

$$\int \frac{\mu_j(\frac{-\mathbf{x}}{H_{\mu_j}})}{|H_{\mu_j}|} * p(\mathbf{x})d\mathbf{x} = 1 \qquad \text{(B.23)}$$

where the equality comes from observing that $p(\mathbf{x})$ is a density function, and from observing that also all μ_j are normalized by their volumes $|H_{\mu_j}|$. Since N is the same lattice as M, the additional convolution of N in the expectation value (B.9) is exactly the same as the convolution of M with $p(\mathbf{x})$, again resulting in $p(\mathbf{x})$.

Finally turning to the variance we have for T

$$T = \sum_{s=1}^{m} \frac{1}{m'} \frac{N_{S,\Delta}(\mathbf{x}-\mathbf{x}_s)}{V_N(\mathbf{x}-\mathbf{x}_s)} M_{I,B}(\mathbf{x}_s - \mathbf{x}_i) \qquad \text{(B.24)}$$

Due to $m' = n$ and the fact that the summation of the kernels from one lattice is one, the supremum of T becomes:

$$\sup\{T\} \leq \frac{1}{nV_N(\mathbf{x}-\mathbf{x}_s)} \qquad \text{(B.25)}$$

Since V_N is a function of the position in the feature space (and not all volumes are identical) we may get *locally* smaller variance than using a fixed kernel in lattice N (the fixed-bandwidth approach). Off course the real difficulty with these choices is finding the local perfect pre-filters μ_j, which is the general problem addressed in *locally adaptive* estimators.

Appendix C

Allowed Kernels

In order to find which kernels which satisfy (see 3.2.1):

$$\sum_{s=1}^{m} \mu(\frac{\mathbf{x}_i - \mathbf{x}_s}{H_\mu}) = \frac{V_\mu |H_\mu|}{\Delta_S} \; \forall \; \mathbf{x}_i \tag{C.1}$$

we will analyze the Fourier transform for the left part of the above equation:

$$\mathcal{F}[\sum_{s=1}^{m} \mu(\frac{\mathbf{x} - \mathbf{x}_s}{H_\mu})] = |H_\mu| M(\omega^T H_\mu) \sum_{s=1}^{m} e^{-j\omega^T \mathbf{x}_s} \tag{C.2}$$

Where ω is a d-dimensional frequency vector, ω^T the transpose of ω (T is used here as the transpose operator). Since $\mu(\mathbf{x})$ is a separable function, there always exists some principle axes such that $\mu(\frac{\mathbf{x}}{H_\mu})$ is also separable. In order to see this we use the coordinate-transformation $\mathbf{u} = \frac{\mathbf{x}}{H_\mu}$ we obtain:

$$\mu(\frac{\mathbf{x}}{H_\mu}) = \mu(\mathbf{u}) \tag{C.3}$$

Since μ is separable we may write:

$$\mu(\mathbf{u}) = \mu_x(u_x)\mu_y(u_y)... \tag{C.4}$$

and we use:

$$\int \mu(\mathbf{u}) d\mathbf{u} = \int \mu_x(u_x)\mu_y(u_y)...du_x du_y... = V_{\mu_x} V_{\mu_y} ... = V_\mu \tag{C.5}$$

Since the Fourier transform of a separable function is a product of the Fourier transforms of each function (see [86]) we can write for the Fourier transform of $\mu(\frac{\mathbf{u}}{L})$:

$$\mathcal{F}[\mu(\mathbf{u}] = M_x(\omega_x)M_y(\omega_y)... \tag{C.6}$$

Where we have used ω_x and ω_y for the one-dimensional frequency components of ω. Continuing with each individual one-dimensional function, using the Taylor-series expansion for $M_x(\omega_x)$, we obtain for ω near 0:

$$M_x(\omega_x) = M_x(0) + \frac{dM_x(0)}{d\omega_x}\omega_x + O(\omega^2) \tag{C.7}$$

If μ has finite moments m_n:

$$m_n = \int {u_x}^n \mu(u_x) du_x \tag{C.8}$$

then it is proven [106] that the expansion of $M(\omega_x)$ can be written as a series of its moments:

$$M_x(\omega_x) = M_x(0) + -jm_1\omega_x + O({\omega_x}^2) \tag{C.9}$$

Since the kernel μ_x is a symmetric function, its Fourier transform is strictly real-valued, and all odd moments are zero. A first order approximation of M_x then becomes:

$$\hat{M}_x(\omega_x) = M_x(0) + O(\omega_x^2) \tag{C.10}$$

For non-symmetric kernels this would be a $O(\omega_x)$ approximation. In general it will be a good approximation for *reasonable* kernels. However, since it is only a good approximation for small values of ω_x, we need to find a suitable bandwidth Δ_{ω_x} for $\hat{M}_x(\omega_x)$ we write:

$$\hat{M}_x(\omega_x) = M_x(0) b(\frac{\omega}{0.5\Delta_{\omega_x}}) \tag{C.11}$$

where $b(\frac{\omega_x}{0.5\Delta_{\omega_x}})$ is a block-function, or bin, with width Δ_{ω_x}. Requiring that:

$$\int \hat{M}_x(\omega_x) d\omega_x = \int M_x(\omega_x) d\omega_x \tag{C.12}$$

yields

$$\Delta_{\omega_x} = \frac{1}{M_x(0)} \int M_x(\omega_x) d\omega_x = \frac{2\pi\mu_x(0)}{M_x(0)} = \frac{2\pi}{M_x(0)} = \frac{2\pi}{V_{\mu_x}} \tag{C.13}$$

which is a well-know measure of the bandwidth for a function $M_x(\omega_x)$ (see [106]), and it has an equivalent measure for the *duration* Δ_{u_x} according to:

$$\Delta_{u_x} = \frac{1}{\mu_x(0)} \int \mu_x(u_x) du_x = \frac{(V_{\mu_x})}{\mu_x(0)} = \frac{M_x(0)}{\mu_x(0)} = \frac{2\pi}{\Delta_{\omega_x}} \tag{C.14}$$

such that:

$$\Delta_{u_x} \Delta_{\omega_x} = 2\pi \tag{C.15}$$

Note that for convenience we have used here that $sup\{\mu_x\} = \mu_x(0) = 1$, however (C.15) holds independent of the supremum of μ_x. Repeating this exercise for all one-dimensional functions and substituting in (C.6) gives:

$$\begin{aligned}\mathcal{F}[\mu(\mathbf{u})] &= M_x(0)b(\frac{\omega_x}{0.5\Delta_{\omega_x}})M_y(0)b(\frac{\omega_y}{0.5\Delta_{\omega_y}})... \\ &= V_\mu b(\frac{\omega}{0.5D_\omega}) \\ &= M(\omega) \end{aligned} \tag{C.16}$$

Where D_ω is a d-dimensional diagonal matrix with $\Delta_{\omega_x}, \Delta_{\omega_y}...$ on its diagonal. This matrix can be thought of as being the multidimensional bandwidth. If all one-dimensional functions are equal $\mu_x = \mu_y = ...$, then all one-dimensional volumes are equal to $V_\mu^{\frac{1}{d}}$, the multidimensional bandwidth then becomes:

$$D_\omega = \frac{2\pi}{V_\mu^{\frac{1}{d}}} Id \tag{C.17}$$

Where Id is the d-dimensional unity matrix. Since:

$$\mathcal{F}[\mu(\frac{\mathbf{x}}{H_\mu})] = |H_\mu| M(\omega^T H_\mu) \tag{C.18}$$

We get by using (C.16):

$$\begin{aligned}\mathcal{F}[\mu(\frac{\mathbf{x}}{H_\mu})] &= V_\mu |H_\mu| b(\frac{\omega}{0.5D_\omega}) \\ where \quad D_\omega &= \frac{2\pi}{V_\mu^{\frac{1}{d}} H_\mu}\end{aligned} \tag{C.19}$$

Using (C.19) in (C.2) gives:

$$\mathcal{F}[\sum_{s=1}^{m} \mu(\frac{\mathbf{x} - \mathbf{x}_s}{H_\mu})] = V_\mu |H_\mu| b(\frac{\omega}{0.5D_\omega}) \sum_{s=1}^{m} e^{-j\omega^T \mathbf{x}_s} \tag{C.20}$$

where, since $\mathbf{x}_s = S\mathbf{n}_s$ and $\mathbf{n}_s \in N$ (see 3.2.1), the summation can be written as:

$$\sum_{s=1}^{m} e^{-j\omega^T \mathbf{x}_s} = \sum_{s=1}^{m} e^{-j\omega^T S\mathbf{n}_s} \tag{C.21}$$

It is well-known from sampling theory that this can be thought of as a series of m replica's of the Fourier-transform of a (virtual) uniform block-function (sometimes denoted as the sampling window) defined over the Lattice L. each replica having magnitude m in the frequency domain. These replica's are positioned in the d-dimensional frequency domain according to a reciprocal lattice defined by R and coefficients $\mathbf{n}_s \in N$ with

$$R = \frac{2\pi}{S} \tag{C.22}$$

Multiplying these replica's with the Fourier transform of μ leads to a single replica centered at $\omega = 0$ with magnitude $mV_\mu|H_\mu|$ if

$$\begin{aligned} D_\omega &= \frac{2\pi}{S} \\ mV_\mu|H_\mu| &= m|S| \end{aligned} \tag{C.23}$$

substituting (C.19) gives:

$$\begin{aligned} \frac{2\pi}{V_\mu^{\frac{1}{d}} H_\mu} = \frac{2\pi}{S} \Rightarrow \\ S = V_\mu^{\frac{1}{d}} H_\mu \end{aligned} \tag{C.24}$$

note that the sample-volume, also known as the *Nyquist density*, equals:

$$\Delta_S = |S| = V_\mu|H_\mu| \tag{C.25}$$

such that (C.23) is also satisfied.

The single remaining replica with magnitude corresponds to a uniform function in the spatial-domain defined over the complete space where we have samples $\mathbf{x}_s$. Hence, if we require that the sampling takes places over the complete space of data where $\mu(\frac{\mathbf{x_i}-\mathbf{x}}{H_\mu})$ exists then

$$\sum_{s=1}^{m} \mu(\frac{\mathbf{x_i} - \mathbf{x}_s}{H_\mu}) = \frac{V_\mu|H_\mu|}{\Delta_S} = 1 \; \forall \; \mathbf{x}_i \tag{C.26}$$

This means that if the kernels (1) are *reasonable* kernels, and (2) have a duration density that equals a sampling volume, then C.1 holds. Note that second condition can also be reversed: the sample-volume should equal the duration density.

As a final result, we observe that the Nyquist Density equals the duration density of $\mu(\mathbf{x})$:

$$\Delta_{\mathbf{x}} = \frac{1}{\mu(\mathbf{0})} \int \mu(\frac{\mathbf{x}}{H_\mu}) d\mathbf{x} = V_\mu|H_\mu| = \Delta_S \tag{C.27}$$

Therefore, the relation between bandwidth, duration and sampling matrix in the d-dimensional case can be stated as

$$\begin{aligned} D_{\mathbf{x}} D_\omega = S D_\omega = 2\pi Id \Rightarrow \\ \Delta_{\mathbf{x}} \Delta_\omega = \Delta_S \Delta_\omega = (2\pi)^d \end{aligned} \tag{C.28}$$

Where it is known from Fourier analysis that the relation between bandwidth and duration (equal to (C.15) in one-dimension) always holds, and where we have shown that:

$$D_{\mathbf{x}} = S \tag{C.29}$$

only holds for *reasonable* kernel functions if (C.26) is satisfied.

An example of a summation of Gaussian kernels in one-dimension for which $o(\omega^2)$ is not negligible is given in Fig. C.1. It can be seen that even in this case the approximation error is still small (in the order of 10 percent of the maximum). Hence, even for non-reasonable kernels, the above analysis holds fairly well.

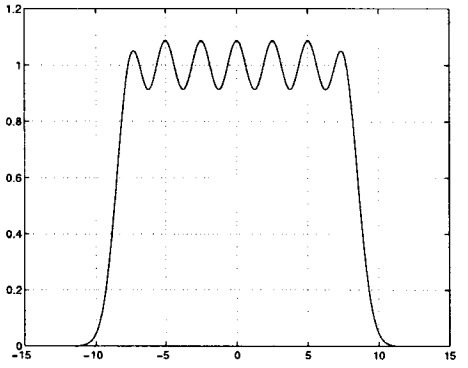


Figure C.1: *A summation of Gaussians at sampling distance* $(2\pi)^{0.5}$.

Appendix D

Order Independence

In this section we proof that:

$$h(x)[f(x) * g(x)] = [h(x)f(x)] * g(x) \tag{D.1}$$

Straightforward Fourier transforming of the left-hand side gives:

$$\mathcal{F}(h(x)[f(x) * g(x)]) = H(\omega) * [F(\omega)G(\omega)] \tag{D.2}$$

Applying the Fourier definition we obtain:

$$\begin{aligned} \mathcal{F}(h(x)[f(x) * g(x)]) &= \int (h(x)[f(x) * g(x)])e^{-j\omega x}dx \\ &= \int\int h(x)f(x-u)g(u)du\ e^{-j\omega x}dx \\ &= \int\int h(x)f(x-u)e^{-j\omega x}dx\ g(u)du \end{aligned} \tag{D.3}$$

noting that the second integral is simply the Fourier transform of a product of functions, we get

$$\begin{aligned} \int [H(\omega) * F(\omega)]e^{-j\omega u}g(u)du &= [H(\omega) * F(\omega)] \int g(u)e^{-j\omega u}du \\ &= [H(\omega) * F(\omega)]G(\omega) \end{aligned} \tag{D.4}$$

(D.5)

Hence,

$$H(\omega) * [F(\omega)G(\omega)] = [H(\omega) * F(\omega)]G(\omega) \tag{D.6}$$

and inverse transforming yields:

$$h(x)[f(x) * g(x)] = [h(x)f(x)] * g(x) \tag{D.7}$$

Which means that the order of a convolution and a product is interchangeable.

Appendix E

MISE Analysis

The MISE is defined as:

$$MISE = E\int [p_S(\mathbf{x}) - \hat{p}_S(\mathbf{x})]^2 d\mathbf{x} \tag{E.1}$$

which can be written as:

$$MISE = E\int [p_S(\mathbf{x}) - \bar{p}_S(\mathbf{x}) + \bar{p}_S(\mathbf{x}) - \hat{p}_S(\mathbf{x})]^2 d\mathbf{x} = E\int [\epsilon_1 + \epsilon_2]^2 d\mathbf{x} \tag{E.2}$$

assuming that $\epsilon_1 << \epsilon_2$

$$\begin{aligned} MISE &= E\int [\epsilon_2]^2 d\mathbf{x} \\ &= \int E[\epsilon_2]^2 d\mathbf{x} \\ &= \int var[\hat{p}_S(\mathbf{x})] d\mathbf{x} \\ &= \int E[\hat{p}_S^2(\mathbf{x})] - [\bar{p}_S(\mathbf{x})]^2 d\mathbf{x} \end{aligned} \tag{E.3}$$

noting that the integration takes place over the space V where $\bar{p}_S(\mathbf{x})$ exists, then according to Cauchy-Schwartz:

$$\int [\bar{p}_S(\mathbf{x})]^2 d\mathbf{x} \geq \frac{1}{V}\left[\int \bar{p}_S(\mathbf{x}) d\mathbf{x}\right]^2 \geq \frac{1}{V} \tag{E.4}$$

and since:

$$\int E[\hat{p}_S^2(\mathbf{x})] d\mathbf{x} < sup\{\hat{p}_S\}\int E[\hat{p}_S(\mathbf{x})] d\mathbf{x} < sup\{\hat{p}_S\} \tag{E.5}$$

We finally obtain:

$$MISE \leq sup\{\hat{p}_S\} - \frac{1}{V} \tag{E.6}$$

Appendix F

Proof of Equivalence

Given a dataset containing n examples $\mathbf{x}_i$ with labels y_i with $y_i \in \{C1, ..., C_c\}$. Further, given a specific rule consisting of r specific rules, for which holds that:

$$\sum_{s=1}^{r} \mu_{R_s}(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in X^d. \tag{F.1}$$

TO BE PROVEN:

Given a general H_g^k then:

$$P(C_k|R_g) = \frac{\sum_{R_s \subset R_g} P(C_k|R_s)P(R_s)}{\sum_{R_s \subset R_g} P(R_s)} \tag{F.2}$$

PROOF:

We will proof this for the two-dimensional case, and extensions to more dimensions is only trivial. Notations here are the same as used in the section 5.1. Suppose we take the general rule:

H_u: *If* x_1 *is* A_u *then* c *is* C_k

Then according to equation 5.7:

$$P(C_k|R_u) = \frac{\sum_{i=1}^{n} \mu_{R_u}(\mathbf{x}_{1i})\mu_{C_k}(y_i)}{\sum_{i=1}^{n} \mu_{R_u}(\mathbf{x}_i)} \tag{F.3}$$

Since $\mu_{R_u}(\mathbf{x}) = \mu_{A_u}(x_1)$, substituting gives:

$$P(C_k|R_u) = \frac{\sum_{i=1}^{n} \mu_{A_u}(x_{1i})\mu_{C_k}(y_i)}{\sum_{i=1}^{n} \mu_{A_u}(x_{1,i})} \tag{F.4}$$

Since the summation of all specific premises equals one, and all premises are separable, then also:

$$\sum_{u=1}^{a}\sum_{v=1}^{b}\mu_{A_u}(x_{1i})\mu_{B_v}(x_{2i}) = 1 \quad \forall x_i \in \mathcal{R}^1 \tag{F.5}$$

Therefore

$$P(C_k|R_u) = \frac{\sum_{i=1}^{n}\sum_{v=1}^{b}\mu_{A_u}(x_{1i})\mu_{B_v}(x_{2i})\mu_{C_k}(y_i)}{\sum_{i=1}^{n}\sum_{v=1}^{b}\mu_{A_u}(x_{1i})\mu_{B_v}(x_{2i})} \tag{F.6}$$

because the summation over v is indeed a summation over all premises covered by R_u, the latter can be written as:

$$P(C_k|R_g) = \frac{\sum_{i=1}^{n}\sum_{R_{uv}\subset R_u}\mu_{R_{uv}}(\mathbf{x}_i)\mu_{C_k}(y_i)}{\sum_{i=1}^{n}\sum_{R_{uv}\subset R_u}\mu_{R_{uv}}(\mathbf{x}_i)} \tag{F.7}$$

noting that:

$$P(C_k|R_s) = \frac{\sum_{i=1}^{n}\mu_{R_{uv}}(\mathbf{x}_i)\mu_{C_k}(y_i)}{\sum_{i=1}^{n}\mu_{R_{uv}}(\mathbf{x}_i)} \tag{F.8}$$

and that:

$$P(R_{uv}) = \sum_{i=1}^{n}\mu_{R_{uv}}(\mathbf{x}_i) \tag{F.9}$$

equation C.26 can indeed be written as:

$$P(C_k|R_u) = \frac{\sum_{R_{uv}\subset R_u}P(C_k|R_{uv})P(R_{uv})}{\sum_{R_{uv}\subset R_u}P(R_{uv})} \tag{F.10}$$

Q.E.D.

Appendix G

Example Rule Base

This appendix shows a rule-base derived for the case-study in the Intelligent Anesthesia Monitor Project. The rule-base contains 42 rules, for 6 features and 3 classes. The rule-base was made out of 7 qualifications for each feature: too High, Very High, High, Normal, Low, Very Low and Too Low. These qualifications have been found by using a clustering algorithm as a preprocessing step for FILER (K-means clustering, K=7).

Blood pressure features

The number behind every feature indicates the number of times that it is used in the rule-base, and is a measure for its importance. In this rule-base the three most important features are a0, a4 and a5.

- a0: 30 minute moving average (34)
- a1: difference between momentary value and 30 minute average (15)
- a2: 3 minute trend (LR) (7)
- a3: 7 minutes trend (14)
- a4: 15 minutes trend (21)
- a5: 30 minutes trend (17)

Intervention classes

- class 0: no intervention
- class 1: Phenylefrine intervention
- class 2: Sufentanyl intervention

Rule base

a rule like
a1 Very High
class: 0 2.366270e+01 class: 1 5.599137e+00 class: 2 4.440931e+01
should be read as: "If the momentary value of the blood pressure is very high then a sufentanyl intervention is most probable"

The most probable case can be found by taking the maximum of the three numbers indicated behind "class:". These numbers indicate the absolute number of clear cases for that class (which is not an integer due to fuzzy weighting), which is a measure of the total number of cases of that class. Simply put: dividing one number by the sum of all numbers leads to a probability for that class given the premise).

a0 high a1 very low
class: 0 1.163199e+01 class: 1 1.183178e-01 class: 2 9.272606e+00
a0 very low a1 very low
class: 0 8.521908e+00 class: 1 6.343450e+01 class: 2 2.823062e+00
a0 too low a1 very low
class: 0 2.049554e+00 class: 1 5.513427e+01 class: 2 2.161574e+00
a5 normal
class: 0 1.558335e+02 class: 1 3.615412e+01 class: 2 1.341911e+02
a5 high
class: 0 8.489674e+00 class: 1 1.466261e+00 class: 2 5.063158e+01
a5 very low
class: 0 1.576841e+01 class: 1 9.225681e+01 class: 2 1.500832e+01
a3 normal
class: 0 1.317410e+02 class: 1 4.051878e+01 class: 2 1.621139e+02
a3 very low
class: 0 1.041042e+01 class: 1 8.501967e+01 class: 2 6.487026e+00
a1 very low
class: 0 4.603782e+01 class: 1 1.392929e+02 class: 2 3.792829e+01
a1 normal
class: 0 4.639794e+01 class: 1 1.282770e+01 class: 2 1.016855e+02
a1 low
class: 0 2.028427e+02 class: 1 5.210011e+01 class: 2 1.528519e+02
a1 too low
class: 0 5.721246e+00 class: 1 9.377926e+01 class: 2 5.531390e+00
a1 low a4 normal
class: 0 1.530308e+02 class: 1 3.622784e+01 class: 2 8.362950e+01
a0 very low a5 low
class: 0 2.548640e+01 class: 1 6.635520e+01 class: 2 5.452350e+00
a0 normal a5 low
class: 0 2.863652e+01 class: 1 8.770314e+00 class: 2 1.767939e+01
a0 too low a5 low
class: 0 8.093677e+00 class: 1 6.333472e+01 class: 2 4.273518e+00

a4 low a5 high
class: 0 2.744008e+00 class: 1 2.834246e-04 class: 2 6.406971e-02
a4 high a5 high
class: 0 4.159688e+00 class: 1 4.436253e-01 class: 2 4.736035e+01
a0 too low
class: 0 2.527399e+01 class: 1 8.774475e+01 class: 2 9.150882e+00
a0 very low
class: 0 7.040976e+01 class: 1 1.133527e+02 class: 2 1.513789e+01
a0 high
class: 0 4.753977e+01 class: 1 7.250985e+00 class: 2 8.033635e+01
a0 too high
class: 0 8.793281e+00 class: 1 2.193219e-08 class: 2 1.842056e+01
a0 normal
class: 0 6.184933e+01 class: 1 2.796693e+01 class: 2 7.386781e+01
a0 very high
class: 0 1.846845e+01 class: 1 1.036654e+00 class: 2 5.811315e+01
a0 high a3 normal
class: 0 1.760851e+01 class: 1 1.115570e-02 class: 2 4.538472e+01
a4 normal
class: 0 2.019216e+02 class: 1 8.014142e+01 class: 2 1.328624e+02
a4 very high
class: 0 2.881312e-04 class: 1 4.110590e-07 class: 2 1.334571e+01
a4 high
class: 0 3.292405e+01 class: 1 1.124598e+01 class: 2 9.555361e+01
a4 very low
class: 0 5.436791e+00 class: 1 6.898084e+01 class: 2 3.870895e+00
a3 normal a4 high
class: 0 1.924602e+01 class: 1 4.425953e+00 class: 2 7.321201e+01
a3 normal a5 low
class: 0 6.107164e+01 class: 1 1.612282e+01 class: 2 2.812387e+01
a3 normal a5 high
class: 0 3.593516e+00 class: 1 3.800267e-01 class: 2 4.087532e+01
a3 low a5 normal
class: 0 8.819509e+01 class: 1 1.723559e+01 class: 2 3.591253e+01
a3 normal a5 normal
class: 0 5.796359e+01 class: 1 1.126857e+01 class: 2 9.001488e+01
a1 very high a2 very high
class: 0 2.184258e-08 class: 1 9.999942e-01 class: 2 4.790343e-06
a1 normal a2 high a4 low
class: 0 3.387117e+00 class: 1 1.632477e-02 class: 2 4.325167e-01
a1 very low a2 normal a4 high
class: 0 1.975120e+00 class: 1 1.595885e-04 class: 2 2.749869e-05
a1 normal a3 low
class: 0 9.776712e+00 class: 1 5.891989e-01 class: 2 3.058414e+00
a0 too low a2 normal
class: 0 1.569192e+01 class: 1 7.333161e+01 class: 2 5.952753e+00

a2 high
class: 0 7.767218e+01 class: 1 4.639984e+01 class: 2 1.010958e+02
a0 too low a2 high a3 normal
class: 0 7.525588e+00 class: 1 2.729027e+00 class: 2 2.444520e+00
a0 very low a4 low
class: 0 1.091771e+01 class: 1 5.889176e+01 class: 2 3.388290e+00
a0 very low a1 very low a2 low a3 very low a5 normal
class: 0 1.715626e-01 class: 1 1.519920e-01 class: 2 6.562570e-01

Leren en Redeneren op basis van de Vage Kansrekening

Dit proefschrift behandelt het probleem van kennisacquisitie voor beslisondersteuning in veeleisende omgevingen. In veeleisende omgevingen is het noodzakelijk om expliciete kennis te verkrijgen waarmee de juiste beslissingen kunnen worden genomen *en* kunnen worden uitgelegd. Deze kennis kan gebruikt worden in een beslisondersteuningssysteem om het besluitvormingsproces van experts te verbeteren. Een voorbeeld van een veeleisende omgeving is de patientbewaking in de anesthesie, hetgeen onderwerp van studie is geweest in het intelligente anesthesie monitor project van de Technische Universiteit Delft in samenwerking met het Academisch Medisch Centrum in Amsterdam. Dit project heeft veel van het werk in dit proefschrift gestimuleerd.

De aanpak voor kennisacquisitie in dit proefschrift is gebaseerd op het leren van regels uit voorbeelden. Er wordt beargumenteerd dat het resultaat van leren een set regels moet zijn die enerzijds goed past bij de data (voorbeelden) van het beslisprobleem (data-fit) en anderzijds past bij het referentie kader van de expert (mentale-fit). De reden hiervoor is dat beide aspecten noodzakelijk zijn voor het maken van en uitleggen van de juiste beslissingen. Hiertoe wordt een synthese gemaakt tussen kansdichtheidschatting, dat een nadruk heeft op data-fit, en vage regelinductie, dat een nadruk heeft op mentale-fit. Om deze synthese mogelijk te maken is een generiek raamwerk voor onzekerheidscalculus ontwikkeld: de vage kansrekening. De vage kansrekening is gebaseerd op de kans op een vage gebeurtenis en is zeer geschikt voor leren en redeneren met onzekerheid. Deze vage kansrekening is een van de belangrijke bijdragen van dit proefschrift.

Een van de validaties voor de vage kansrekening is dat een nieuwe en efficiente kernel-gebaseerde dichtheid schatter kan worden afgeleid: de dubbele-kernel schatter. Er wordt aangetoond hoe deze schatter wiskundig samenhangt met de welbekende Parzen schatter. Experimenten laten zien dat de dubbele-kernel schatter in beslisproblemen (zoals klassificatie) nauwkeuriger kan zijn met minder kernels dan de Parzen schatter. De dubbele kernel schatter is een van de interessante additionele bijdragen van dit proefschrift.

Nog een belangrijke bijdrage van dit proefschrift is een nieuw regelinductie algoritme: "Fuzzy Probabilistic Rule Induction". Dit algoritme, gebaseerd op de vage kansrekening, volgt het "covering-paradigma" voor regelinductie. De regels worden geselecteerd met behulp van de J-informatiemaat, welke gerelateerd is aan de wederzijdse informatie die ook wordt gebruikt voor het opstellen van beslisbomen. Experimenten met een implementatie van dit algoritme genaamd FILER laten zien dat, in vergelijking met andere algoritmen, zeer nauwkeurige beslissingen kunnen worden genomen die met slechts weinig algemene regels kunnen worden uitgelegd. Het blijft echter een probleem om met generieke regels rekening te houden met de covariantie van de data. Zonder generalisatie kan deze covariantie weliswaar worden meegenomen maar dan degenereert de regelinductie op basis van de vage kansrekening tot de dubbele-kernel schatter techniek.

De uiteindelijke bijdrage van dit proefschrift is de toepassing van de regelinductie op basis van de vage kansrekening voor patientbewaking in de anesthesie. Bewaking in de anesthesie is een voorbeeld van een veeleisende omgeving waarin veel bronnen van complexe informatie moeten worden verwerkt in een relatief kort tijdsbestek. Een complicerende factor in de patientbewaking is de tijdsafhankelijkheid van de fysiologische signalen. De aanpak die wordt gevolgd in dit proefschrift is het representeren van de veranderingen van de parameters in de tijd door een aantal trends. Op basis van deze trends en andere kenmerken kunnen regels geleerd worden uit voorbeelden van "alarm" situaties verkregen van experts. Met deze regels is het mogelijk dat een systeem redeneert zodat (1) de anesthesist gewaarschuwd kan worden en (2) een uitleg gegeven kan worden voor een dergelijke waarschuwing. Een case-study wordt behandeld waarin op basis van ongeveer 1000 voorbeelden, verkregen met toestemming van de Rijksuniversiteit Groningen, ongeveer 40 regels werden afgeleid. Op grond van cross-validation werd bepaald dat de regels bijna 80 % van de niet geobserveerde voorbeelden op juiste wijze zouden moeten kunnen herkennen en zinvol kunnen uitleggen. Een panel van experts bevestigde dat dit de verwachte prestatie zou zijn van een expert en tevens konden zij zich vinden in vele door het systeem geleerde generieke regels. Op basis van deze resultaten wordt geconcludeerd dat de regelinductie op basis van de vage kansrekening een bruikbare techniek is voor een beslisondersteuningsysteem in de anesthesie. De uiteindelijke prestatie van het systeem hangt echter af van de kwaliteit van de voorbeelden die door de expert worden gegeven.

Acknowledgments

Research cannot be done without standing on the shoulders of giants (Einstein believed so, and who am I to dispute this). I have stood on many shoulders, and I hope that in the process I did not step on the giants' heads and toes as well. If so, I must apologize. One reason for writing this thesis in the plural "we" is to acknowledge all these giants. Apart from all the unnamed giants, I like to thank a few in particular.

First of all, I like to thank Eric Backer, mon professeur, with whom I had many enlightening discussions. I am proud to say that I consider him as my mental father. I hope this thesis is also a mental fit to his great mind. Also, I like to thank Jan van der Lubbe; his critical remarks on my mathematical ideas as well as on my more philosophical ideas have sharpened me much. I also like to thank my supervisor of the first hour: Jan Gerbrands. Not only because of his supervision, but also because of his work for the IAM project.

I like to thank all my collegues for the many fruitful hours of discussion. Especially I like to thank Marcel Reinders for his support during my darkest hours. Further, I like to thank my students, in particular Helene van der Sluis and Marco Bravenboer, for their enthusiasm and their contributions to FILER. I also like to give special thanks to Mirjam Nieman for proofreading the manuscript.

Much of the IAM project could not have been carried out without the experts in the field: Cor Kalkman of the AMC in Amsterdam, and Rolf Gallant Huet, Bert Ballast, and Fred de Geus of the University of Groningen. I sincerely hope that, one day, they will be able to reap the fruits of the project. Of course I like to thank all members of the IAM project team, especially Philip de Graaf and Erik Vullings: Luctamur et Emergimus!

I could not have done my research without support from my family and friends. Long-lasting support I have had from my parents and my brother, their understanding and their reflections have formed the foundations of this thesis. I am also indebted to my family in Lochem, who gave me a second home where I could relax. However, I am most indebted to Carolien, my guardian angel, who sheltered me from the turmoil of life. To you Carolien, for standing beside me and for giving me the freedom to be myself.

Curriculum Vitae

Gerard C. van den Eijkel was born in Alphen aan den Rijn, on the 14th of February in 1970. In 1988 he obtained his Gymnasium β diploma from the Ichthus College in Enschede, the Netherlands. From August 1988 to April 1994 he studied Applied Physics at the Twente University from where obtained his Master's degree (ir.). His final dissertation was on modeling and control of a cooling system for a superconducting quantum interference device. During his study he did a traineeship (stage) at the Shell research laboratory in Oakville, Canada.

From May 1994 to November 1997, he worked as a Ph.D. student at the Information and Communication Theory group of the Department of Electrical Engineering at the Delft University of Technology. His Ph.D. was carried out within the Intelligent Anesthesia Monitor Project, funded by the "Beek Committee" sponsoring program. In December 1997, he became assistant professor (UD) in the Information and Communication Theory group.