On Classification of Logging Data

DISSERTATION

zur Erlangung des Grades eines Doktors
der Naturwissenschaften

vorgelegt von

Ralf Gelfort
aus Karlsruhe

Genehmigt von der
Fakultät für Energie- und Wirtschaftswissenschaften
der Technischen Universität Clausthal

Tag der mündlichen Prüfung:
19.12.2006

Vorsitzender der Promotionskommission: Prof. Dr. N. Meyer
Hauptberichterstatter: Prof. Dr. H.-J. Kümpel
Berichterstatter: Prof. Dr. A. Weller

Abstract

The identification of stratigraphical sequences within marine sediment deposits can be accomplished by the combined interpretation of seismic, wireline and core data. If seismic data are ambiguous and core recovery is poor, the accurate detection of sequence boundaries may become difficult. It is desirable to distinguish such sequences automatically, thus avoiding any potentially wrong human interpretation. This task can be performed by learning algorithms that classify logged intervals into stratigraphical sequences.

Seven such supervised learning algorithms (linear and quadratic discriminant analysis, logistic regression, k-nearest neighbour, support vector machine, backpropagation neural network and probabilistic neural network) are applied to logging data. These include borehole wireline, multi-sensor core scan and fluorescence X-ray data sets that have been acquired during the PROMESS-1 cruise in the Mediterranean Sea (Gulf of Lion and the Adriatic Sea).

The algorithms are compared in order to find the best suitable algorithm for the specific task of stratigraphical sequences identification. Those classifiers that are both non-parametric and discriminating (support vector machine, k-nearest neighbour and probabilistic neural network) yield the best results. Given that ample training data are available, quadratic discriminant analysis and backpropagation neural network also perform well. The logging data are found to be non-Gaussian distributed. Non-parametric algorithms seem to adjust better to this than parametric algorithms. Linear classifiers such as linear discriminant analysis and logistic regression cannot be recommended as the data are not linearly separable into classes representing stratigraphical sequences.

It is shown that the classification performance improves if each class is properly represented within the training data set. For the available data, the mean Dipmeter micro-resistivity proves to be the best discriminating log curve as it possibly reflects compaction effects near the borehole wall. The logged stratigraphical units are thus primarily determined by compaction.

The feasibility of automated classification is demonstrated on the basis of PROMESS-1 data. Although the identification of the most appropriate algorithms seems to be specific to this data set, the results may be transferred to other classification goals and geological settings.

Acknowledgements

Although this thesis is my independent work, it would not exist without a lot of people to whom I would like to express my gratitude.

First, I thank Stefan Bornholdt for setting me on the track of support vector machines. With his email message he started my unexpected and wondrous walk through learning machines, feature spaces, neurons, error surfaces and many more things within the realm of pattern recognition.

Thanks go to Thomas Wonik not only for accompanying the EC project in which this work was embedded but for shielding me from the equally challenging jungle of administration. The time and exasperation he saved me have been invaluable to me. Ferdinand Hölscher's patient support especially during the set-up phase of PROMESS-1 is also warmly acknowledged.

I am grateful to Agnes Schumann for our discussions in Berlin and later on by email. Her work has been an inspiration for me. It was very encouraging to see that there *are* other people out there who find this topic equally worthwhile to be explored. There are not many.

I am indebted to Prof. Kümpel and Prof. Weller for their commitment and supervision of this dissertation. Their comments considerably improved the manuscript.

My sincere gratitude goes to Serge Berné. His patient, friendly and amiable way of leading PROMESS-1 to success revealed a genuine scientist dedicated to the questions Nature poses. He was always a source of motivation and support to me.

Thanks a lot to the great PROMESS-1 scientific party. Let me mention master Sergey Pyatovskiy and his fantastic Russian crew, Andrey Shaposhnikov and his skilled drillers and Martin Galavazi and his engineers, but the entire team was awesome, too. Special thanks to Graham Tulloch for being both understanding and supporting and Tina Schoolmeester for ... literally almost everything (that includes Santa Madrona times, too).

Christian Bücker and especially Thomas Günther invested considerable time and effort to provide detailed comments on this thesis. Thank you both so much.

I thank Jens Orzol for those fundamentally important daily coffee breaks, for all those discussions and their repetitions, for nights out and days in, and for leaving Hannover every weekend. The elephant will surely miss us.

This research has profited from the encouragement, advice, support and friendship of Bernard Dennielou, Gwénaël Jouet, Jaime Frigola, Antonio Cattaneo, Gerhard Kurz, Andrei Zschocke, Stefan Weßling, Slavica Babinka, Hanna-Maria Rumpel, Janice Wilke, Stefanie Kischkies, Margarete Linek, Jan Igel, Thomas Grelle and Thies Beilecke. Thanks to all of you.

In Hermann Buness and Franz Binot I found colleagues who were always helping and cordial. This has not gone unnoticed. Keep it up.

Finally, I would like to thank Milena Kmet and this only marks her as special. Thank you for running the first floor so smoothly.

This work has been part of the PROMESS-1 project funded by the European Community under contract no. EVR1-CT-2002-40024.

Contents

1 Introduction **14**

2 Geological Setting **18**

2.1 Gulf of Lion 18
2.2 Adriatic Sea 22
2.3 PROMESS-1 Project 23

3 Data Acquisition **25**

3.1 Core Logging 26
    3.1.1 Depth control 26
    3.1.2 Multi-Sensor Core Logging (MSCL) 27
    3.1.3 X-Ray Fluorescence (XRF) Core Scanning 29
3.2 Wireline Logging 29
    3.2.1 Depth Control 29
    3.2.2 Mud System 30
    3.2.3 Tools 31

4 Classification Techniques **35**

4.1 Pre-Processing 36
    4.1.1 Data distribution 36
    4.1.2 Normalisation 37
    4.1.3 Dimensionality Reduction 37
    4.1.4 Class Labels 38
4.2 Bayesian Decision Theory 38
4.3 Algorithms 40
    4.3.1 Maximum-Likelihood Estimation 41
    4.3.2 Discriminant Analysis 42
    4.3.3 Logistic Regression 44
    4.3.4 Parzen Windows 45
    4.3.5 k-Nearest Neighbour Estimation 47
    4.3.6 Artificial Neural Networks 49

4.3.7 Support Vector Machine 53
4.4 Other Considerations 58
4.4.1 Bagging 58
4.4.2 Bias and Variance 58
4.4.3 Computational Performance and Software Used 60

5 Stratigraphical Classification 63
5.1 Gulf of Lion 63
5.1.1 Conventional Interpretation 63
5.1.2 Data Used 65
5.1.3 Factor Analysis 66
5.1.4 Data Distribution 67
5.1.5 Supervised Learning 68
5.1.6 Input Data Selection 78
5.2 Adriatic Sea 81
5.2.1 Conventional Interpretation 81
5.2.2 Data Used 82
5.2.3 Factor Analysis 83
5.2.4 Data Distribution 83
5.2.5 Supervised Learning 84
5.2.6 Input Data Selection 91

6 Discussion & Conclusions 94

Appendix 109
A. Logging Data (Gulf of Lion) 110
B. Logging Data (Adriatic Sea) 112
C. Tool Parameters 114
D. Data Distribution (Gulf of Lion) 115
E. Data Distribution (Adriatic Sea) 117
F. Classification Results with 50 % Training Data (Gulf of Lion) 119
G. Classification Results with 50 % Training data (Adriatic Sea) 120

Bibliography 121

Index 128

List of Figures

1.1 Classifier design cycle 14
1.2 Training and generalisation 15

2.1 Mediterranean Sea 19
2.2 Gulf of Lion 20
2.3 Gulf of Lion shelf profile 20
2.4 PRGL-1 seismic section 21
2.5 Seismic profile at PRAD-1 22
2.6 Adriatic Sea 23

3.1 Sonic data generation 28
3.2 Borehole vicinity 31

4.1 Visualisation of the learning process 35
4.2 Overview of classification algorithms 40
4.3 Linear discriminant analysis 43
4.4 Parzen windows density estimation and kNN 46
4.5 Bounds on nearest neighbour error rate 48
4.6 Neural network topology 49
4.7 PNN topology 52
4.8 Support vector machine 55
4.9 Bagging and component classifiers 59
4.10 Bias and variance 60
4.11 Computing times 61

5.1 Ca/Fe plot and sequence boundaries (PRGL-1) 64
5.2 Performance vs. training data size plot (PRGL-1) 69
5.3 kNN performance with changing parameter (PRGL-1) 70
5.4 RBF-kernel SVM performance with changing parameters (PRGL-1) . . 71
5.5 ANN performance with changing parameters (PRGL-1) 72
5.6 PNN performance with changing parameter (PRGL-1) 72
5.7 Classification columns of hole PRGL-1 74

5.8 Smoothed classification columns of hole PRGL-1 75
5.9 Performance cross matrix (PRGL-1) 76
5.10 Performance variances due to changing training data sets (PRGL-1) . . 77
5.11 Classification improvement by bagging (PRGL-1) 78
5.12 Performance gain dependence on input data (all data, PRGL-1) 79
5.13 Performance gain dependence on input data (R and Cu data, PRGL-1) 80
5.14 Performance gain dependence on input data (b&w-plot, PRGL-1) . . . 81
5.15 Ca/Fe plot and sequence boundaries (PRAD-1) 82
5.16 Performance vs. training data size plot (PRAD-1) 85
5.17 RBF-kernel SVM performance with changing parameters (PRAD-1) . . 86
5.18 Classification columns of hole PRAD-1 88
5.19 Performance cross matrix (PRAD-1) 89
5.20 Performance variances due to changing training data sets (PRAD-1) . . 90
5.21 Classification improvement by bagging (PRAD-1) 91
5.22 Performance gain dependence on input data (all data, PRAD-1) 92
5.23 Performance gain dependence on input data (b&w-plot, PRAD-1) . . . 92

6.1 Regions of training-to-test data ratio performance 96
6.2 Performance gain dependence on input data (incl. DLL, PRGL-1) . . . 101
6.3 Distribution of resistivity logs by classes (PRGL-1) 102
6.4 Resistivity logs of hole PRGL-1 103
6.5 Classification results including depth information (PRGL-1) 105
A.1 Logging data PRGL-1 (a) 110
A.2 Logging data PRGL-1 (b) 111
B.1 Logging data PRAD-1 (a) 112
B.2 Logging data PRAD-1 (b) 113
D.1 Normal probability plots PRGL-1 115
D.2 Histograms PRGL-1 116
E.1 Normal probability plots PRAD-1 117
E.2 Histograms PRAD-1 118
F.1 Classification columns of hole PRGL-1 (50 % training data) 119
G.1 Classification columns of hole PRAD-1 (50 % training data) 120

List of Tables

5.1 Factor analysis: Eigenvalues of PRGL-1 log curves 66
5.2 Tests of distribution on PRGL-1 log curves 67
5.3 Training data generation (PRGL-1) 68
5.4 Performance gain dependence on input data (PRGL-1) 80
5.5 Factor analysis: Eigenvalues of PRAD-1 log curves 83
5.6 Tests of distribution on PRAD-1 log curves 84
5.7 Training data generation (PRAD-1) 85
5.8 Performance gain dependence on input data (PRAD-1) 93

6.1 Suitability of classifiers 101
6.2 ANOVA results 106
C.3 Tool parameters 114

List of Symbols

General notations:

c	Scalar
$\mathbf{x}$	Numerical vector (assembled scalars)
$\hat{\mathbf{x}}$	Estimate of vector $\mathbf{x}$
$\mathbf{x}^{\mathrm{T}}$	Transpose of vector $\mathbf{x}$
$\lVert\mathbf{x}\rVert$	Euclidean norm of vector $\mathbf{x}$
$\mathbf{A}$	Matrix
$\mathbf{A}^{-1}$	Inverse matrix
$\lvert\mathbf{A}\rvert$	Determinant of matrix $\mathbf{A}$
$\ln(\cdot)$	Natural logarithm $\log_e$
$P(\cdot)$	Probability
$p(\cdot)$	Probability density function
$L(\cdot)$	Loss function
$\mathbb{R}^d$	d-dimensional Euclidean space
$\mathbb{I}$	Identity matrix

Frequently used symbols:

$\boldsymbol{\mu}$	Mean vector
$\boldsymbol{\Sigma}$	Covariance matrix
$\boldsymbol{\theta}$	Parameter vector
$\mathcal{S}$	Set
$\mathcal{R}$	Region
λ	Window length
c	Class(ification) label
k	Number of classes or number of neighbours
p	Number of variables per sample
n	Number of samples
m	Model complexity

"*Entia non sunt multiplicanda praeter necessitatem.*"

William of Ockham (1285 – 1349)

"*We think in generalities, but we live in detail.*"

Alfred North Whitehead (1861 – 1947)

1 Introduction

In recent years, the field of pattern recognition has seen a large increase of attention in the light of new computing capacities and the ability (or curse) to search through vast amounts of data. Both scientists and engineers are involved with the extension of existing and the development of new data mining techniques with diverse applications such as speech recognition, DNA sequence identification, credit card fraud detection, email spam protection or computer vision. The main goal is the feature extraction from a large data set that follows some sort of similar pattern. This pattern is likely being masked by the sheer amount of data it is extracted from or the kind of non-trivial rule it follows. There are two principal fields of pattern recognition: regression and classification. The former assigns one or more continuous valued outputs to each input object; the latter assigns each input object to one of finite classes (Vapnik, 2000). Within this work, only the latter case will be considered as the problem at hand is one of classification: n-dimensional input objects (logging curves) shall be classified into c classes (representing c stratigraphical sequences).

The classification task consists of a number of different activities: data collection, variables choice, model choice, training and evaluation (Duda et al., 2001, see Figure 1.1). Collection of good data and an appropriate variables choice (i.e. selection of logging curves) are crucial for any sound classification performance. The model choice determines the type of classification algorithm and its parameters. Selecting the best classifier for a specific problem (that of classifying logging data from unconsolidated marine sediments of the Mediterranean Sea into stratigraphical units) is the objective of this thesis. Variables and model choice may be enhanced by input of prior knowledge such as weighting of input curves, invariances, etc. Training of the

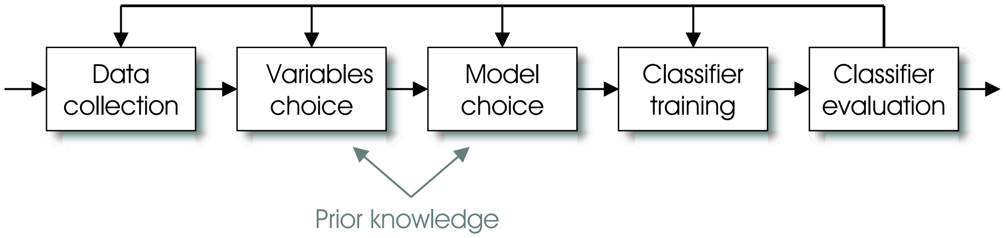


Figure 1.1: Design cycle of a classifier, from Duda et al. (2001). The evaluation process may alter all previous steps in order to achieve a better performance.

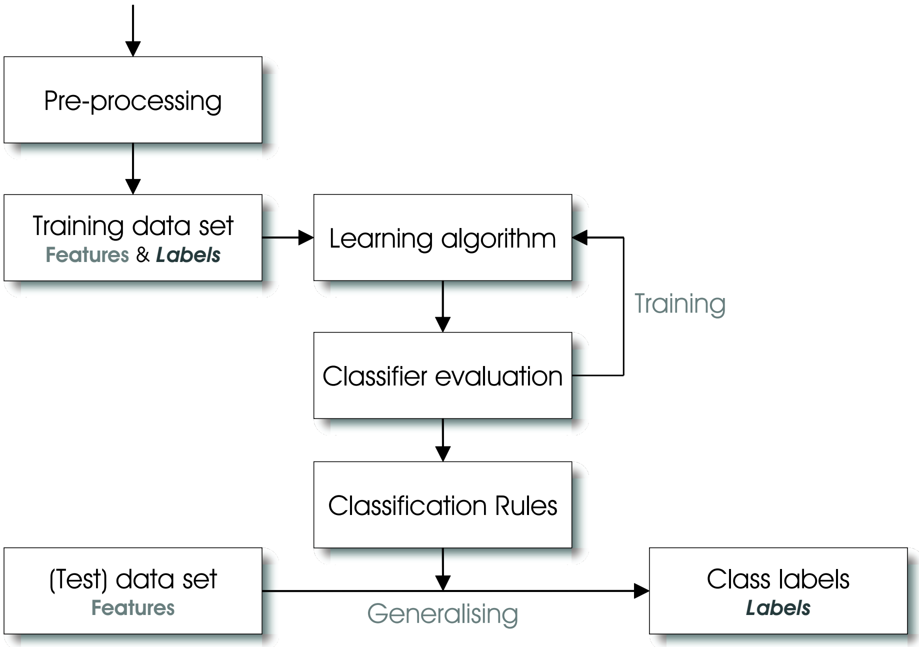


Figure 1.2: Training and generalisation cycles of a learning system. During training, a data set with variables and labels is presented to the classifier. After optimising the classifier, the algorithm found is applied to unknown data containing variables only (generalisation).

chosen classifier and evaluation of its performance complete the classification design cycle.

Classification is achieved by learning which can be divided into two groups: supervised and unsupervised learning. With unsupervised learning, there are no training data presented to the classifier but the algorithms try to find an underlying decision rule based on some measure determined by the algorithm. With supervised learning, the classifier is being trained with a training set of (e.g. logging) variables and labels (e.g. stratigraphical unit names) prior to classification of the (unlabelled) remainder of the data. The application of a trained algorithm to unknown data is called generalisation. Training performance and generalisation performance have to be distinguished, as they may be significantly different (see Figure 1.2). A small training error does not guarantee a low generalisation error, and *vice versa*. *Overfitting* occurs when a classifier is very well trained (low training error) but nonetheless performs poor on new data (large generalisation error). A good classifier avoids overfitting and maintains a good generalisation performance at the same time.

In this work automatic classification is performed after some initial training. In the given scenario of borehole data, the training data set resembles intervals where core and wireline data are complete. The trained algorithm is then applied to data where core information is missing. This situation is frequently encountered in field campaigns where coring is either too expensive or technically impossible, but wireline data are readily available. If classification can be reliably performed with only few cored intervals, these algorithms may provide a work-around in cases with poor or no core recovery.

Classification of borehole data is by no means a new approach. Since the 1960s, cross-plots of 2 or 3 parameters have been created to classify logged intervals into different lithological units. This technique has been refined to a remarkable level of perfection but as far as the industry is concerned, it has never been replaced by other, more sophisticated algorithms (White, pers. comm.). Being almost certainly biased by the interpreter or logging specialist, the results are far away from being impartial.

Statistical methods were applied to borehole data sparsely in the 1980s and 1990s (e.g. Busch et al., 1987; Schumann, 1995; Bücker et al., 2000). Baldwin et al. (1989) were the first to apply artificial neural networks (ANN) to the problem of classification of well logs. Since then, neural networks have been a frequently used tool to extract information from logs, e.g. lithofacies identification (Bhatt & Helle, 2002b), permeability and porosity (Wiener et al., 1991; Wong et al., 1995), lithology (Rogers et al., 1992) or fracture frequency (FitzGerald et al., 1999). Benaouda et al. (1999) compare neural networks with various types of discriminant analysis, some of which outperform the neural network. Derek et al. (1990) report similar results when comparing Bayes classifiers, k-nearest Neighbour (kNN) classifiers and linear discriminant analyses (LDA) with neural networks. Other classification algorithms such as support vector machines (SVM) or hidden Markov models (HMM) have been mentioned in relation to log classification in recent years (Liu & Sacchi, 2003; Wong et al., 2003; Schumann, 2002), but so far these have not yet been widely used in this field of application. In this work, conventional statistical methods, k-nearest neighbour classifiers, neural networks and support vector machines will be applied to log classification and evaluated on their performances. So will be other techniques that up-to-date have not yet been applied to this kind of classification task, namely Parzen windows and logistic regression.

Many of the conclusions of the above cited papers have in common that they recommend general improvements of classifier designs without any specific statement about how these improvements should look like. It seems to be a hope only that a vaguely postulated improvement would lift the more complex algorithms above rather simple ones such like the LDA. Intuitively, for a given problem a simple classification algorithm is preferred over a more complex one. This practice is commonly referred to as *Ockham's razor*, quoting William of Ockham's "entia non sunt multiplicanda praeter necessitatem" (entities shall not be multiplied without necessity). But as the *No Free Lunch* theorem by Wolpert (1995) states, there is no inherently best algorithm for a given problem but only a best classifier for a specific problem. The goal of this thesis is finding such best classifier for the problem of classifying a set of borehole logs into stratigraphical sequences.

Having introduced the terms pattern recognition, classification, supervised learning, training, generalisation and overfitting, the next chapter will present the data to be classified. All log data have been acquired during the PROMESS-1 cruise in the Mediterranean Sea. This project and its geological background will be illustrated. Chapter 3 sets out acquisition and processing of wireline, multi-sensor core logging (MSCL) and fluorescence X-ray (XRF) logging data. Chapter 4 introduces the theoretical foundations of selected classification algorithms. Conventional interpretations from Geologists of the log data in terms of stratigraphic sequence identification and classification results are shown in Chapter 5 along with the application of the algorithms presented in Chapter 4. Discussion on the evaluation of classification algorithms and conclusions follow (Chapter 6).

2 Geological Setting

The Mediterranean Sea is a continent-surrounded basin only linked to the ocean system through the Straits of Gibraltar. It developed during the collision of the Africa-Arabian continent with the Eurasian continent during the Oligocene and Miocene. At the end of the Miocene, the close-off of its western end caused the Messinian Salinity Crisis, a major event all over the Mediterranean Sea with the deposition of a deep-basin evaporitic sequence (Hsü et al., 1973). During the Quaternary, sea-level rise led to sedimentation characterised by margin progradation (e.g. Lofi et al., 2003; Trincardi et al., 2004). The Mediterranean Sea's present coastline extents some 46,000 km, the Sea's average depth is around 1,500 m and it covers roughly 1.7 million km^2. It is commonly divided into a number of gulfs, straits and smaller seas, two being the Gulf of Lion at the southern coast of France and the Adriatic Sea east of Italy. These two areas have long been the object of investigations of European marine geoscientists. They both exhibit continental shelf and upper slope sedimentary sequences deposited during the last 500 ka (see Figure 2.1). During the PROMESS-1 (PROfiles across MEditerranean Sedimentary Systems, part 1) project, drilling, coring and borehole logging at four locations in the Gulf of Lion and the Adriatic Sea have been conducted. The geological settings of these sites along with the project's goals will be outlined in this chapter.

2.1 Gulf of Lion

The Gulf of Lion is a passive continental margin. Present day Quaternary sediments deposited in the Gulf are mainly fed by fluvial input from the Rhône river with an estimated sediment load prior to dam construction of $2 - 8 \times 10^6$ t/a (Pauc, 1970) and to a minor extent from the Aude and Têt rivers with $1.8 - 4 \times 10^6$ t/a and 0.6×10^6 t/a, respectively (CSCF, 1984). Their origin are mainly alpine glaciers and the rivers' drainage basins. A rather constant subsidence rate of around 250 m/Ma (Rabineau, 2001) allows for ample accommodation space. This, in combination with high sediment supply rates, controls sediment deposition at the shelf edge, continental slope (Lofi et al., 2003) and the Rhône deep sea fan (Droz & Bellaiche, 1985). Stratigraphically, the Quaternary sequences on the outer shelf and upper slope exhibit

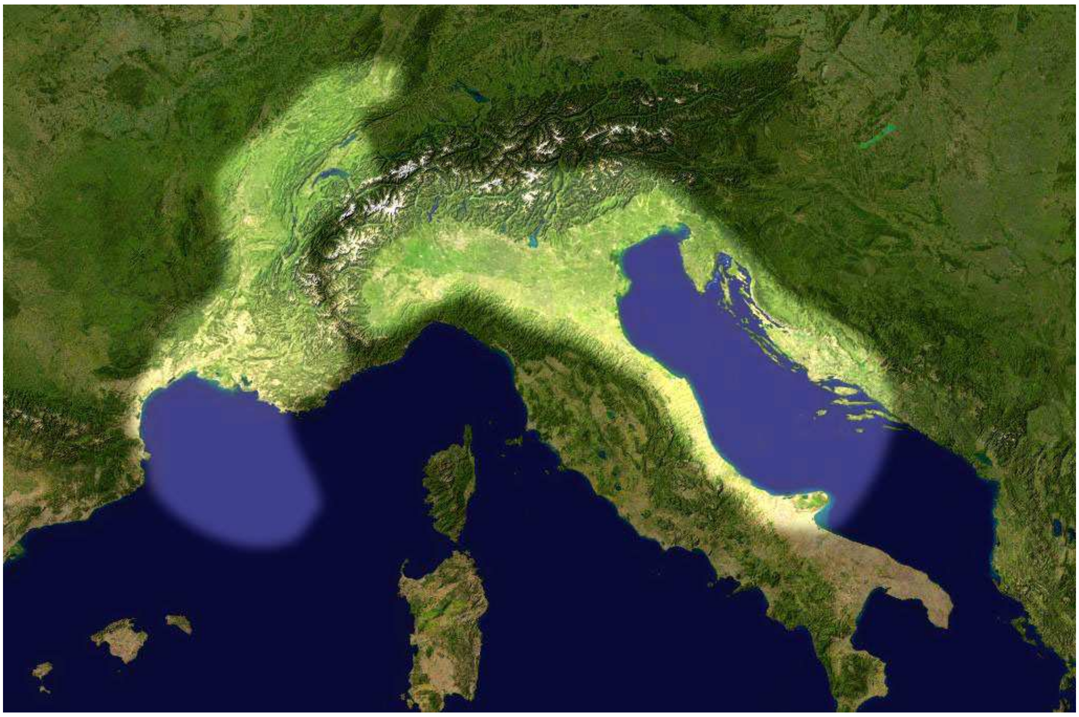


Figure 2.1: Northern Mediterranean Sea and drainage basins of the rivers Rhône and Po (highlighted). Satellite image from USGS (2000).

a combination of several prograding wedges confined by discontinuities. Rabineau et al. (2005) define a motif describing their depositional pattern. It consists of 2-3 horizontally juxtaposed prisms PI, PII and PIII capped by those discontinuities. In their paper, they identify at least five such major erosional surfaces named D30, D40, D50, D60 and D70. Supported by stratigraphic simulations and lithological, palynological, micro-palaeontological and seismic stratigraphical data they strengthen the debated model of a depositional pattern following a 100 ka glacial/interglacial cycle (as opposed to a 20 ka cycle). One of the goals of the PROMESS-1 cruise was to test this model. At locations PRGL-1 and PRGL-2 (see Figures 2.2 and 2.3) boreholes were drilled down to 300 metres and 100 metres below seafloor (mbsf), respectively, penetrating the discontinuities D70 down to D35 (see Figure 2.4). Preliminary micro-palaeontological data show that marine isotope stage (MIS) 13 has been reached at the bottom of hole PRGL-1 (Flores, pers. comm.), thus supporting the 100 ka cycle hypothesis. Grain size distribution and fluorescence X-ray (XRF) data suggest an additional major discontinuity D45 between D40 and D50. This is supported by classification results presented later in this work.

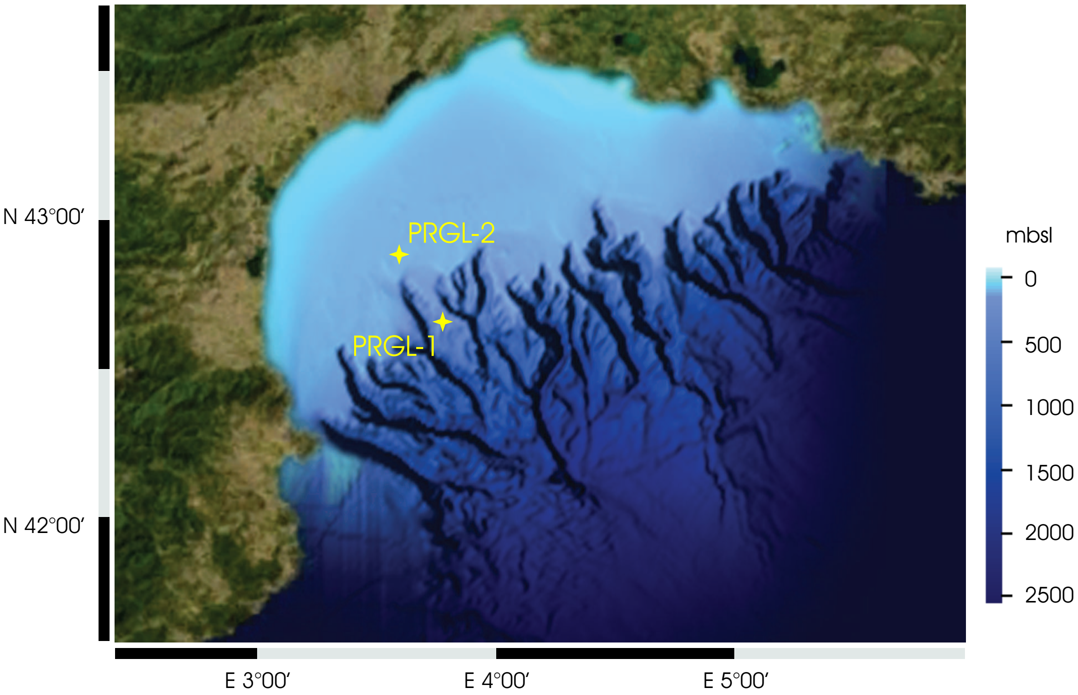


Figure 2.2: Gulf of Lion: Shelf, Rhône canyon systems and drilling locations. PRGL-1 is located on the shelf edge at a water depth of 300 m. PRGL-2 is located on the shelf at a water depth of 100 m. Bathymetry data from Berné et al. (2002), satellite image from USGS (2000).

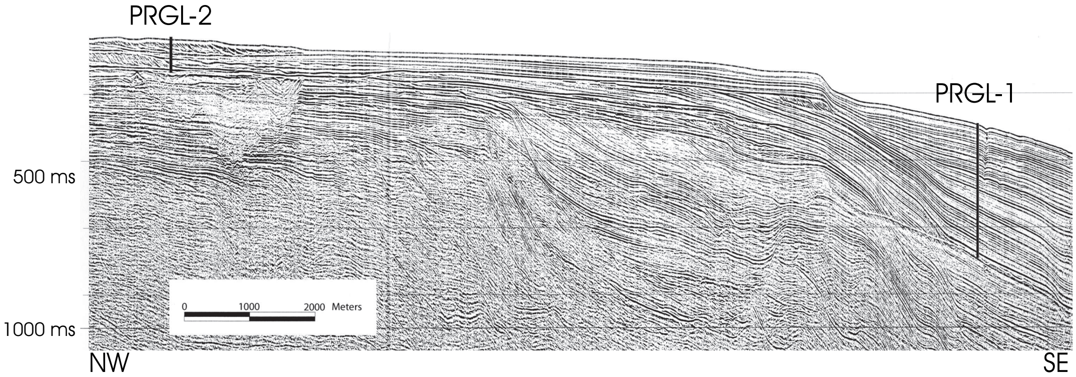


Figure 2.3: Projected drill holes PRGL-1 and PRGL-2 at the shelf and upper slope of the Gulf of Lion shelf. Seismic data courtesy of IFREMER.

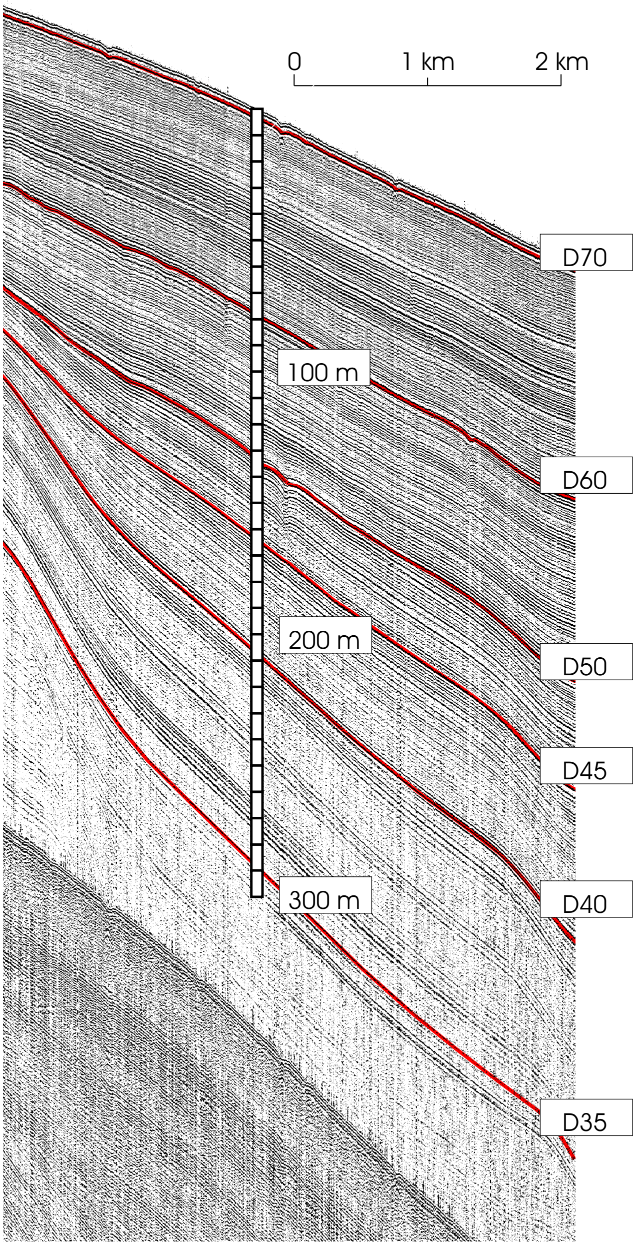


Figure 2.4: Borehole PRGL-1 at the upper slope of the Gulf of Lion shelf. Major discontinuities are shown in red. Water depth is 300 m. Depth scale (mbsf) is an approximation based on seismic velocities. High resolution sparker data courtesy of IFREMER.

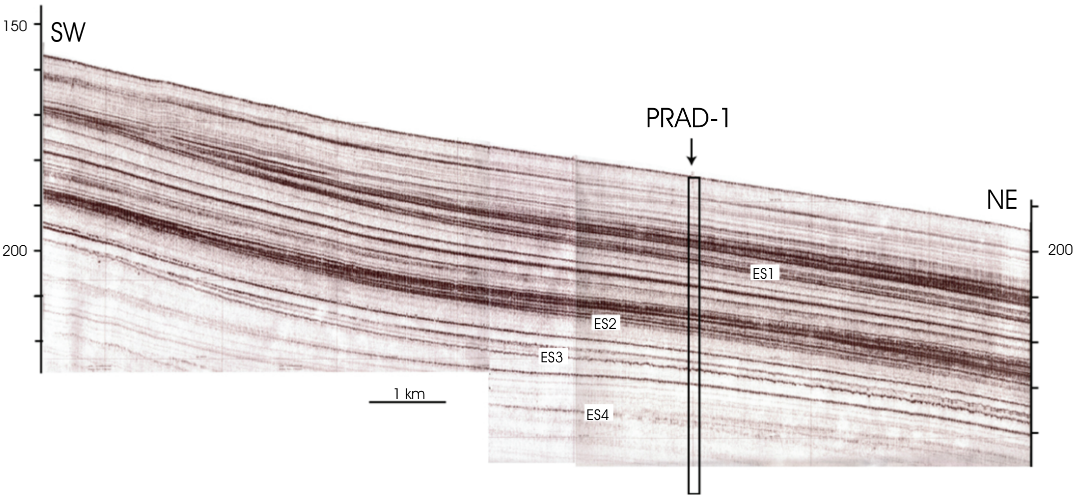


Figure 2.5: Chirp seismic profile AMC-236 intersecting borehole PRAD-1. Depth in mbsl. Courtesy of ISMAR-CNR.

2.2 Adriatic Sea

The Adriatic Sea is a Plio-Quaternary foreland basin of the Apennine chain. During the Pliocene and Quaternary, sediment flux, depositional patterns and direction of progradation changed considerably (Ori et al., 1986). The Po Plain is the basin's main drainage area for sediment input, supplemented by smaller drainage basins of the Apennine chain (Trincardi et al., 1996). Southward sediment transport is controlled by wind-driven currents and waves. Sediment discharge of the smaller rivers from the Apennine coalesces to create accretionary features known as clinoforms (Cattaneo et al., 2004; Nittrouer et al., 2004). Four progradational units correlating to 100–120 ka cycles have been identified for the Central Adriatic basin. Each of them developed during highstand to falling sea-level conditions and is topped by erosional surfaces (ES1 to ES4, see Figure 2.5). The preservation of forced regression deposits was controlled by relative sea-level changes, the eustatic signal with short-term but large magnitude rises and fifth-order cyclic sea-level falls of short duration (Trincardi & Correggiari, 2000). The sequences vary laterally in terms of thickness and depositional geometry which is due to heterogeneous regional deposition prerequisites. Drilling through these four sequences was accomplished during the PROMESS-1 cruise, therefore providing new information about sedimentation processes, age definition and tectonic subsidence (see Figure 2.6).

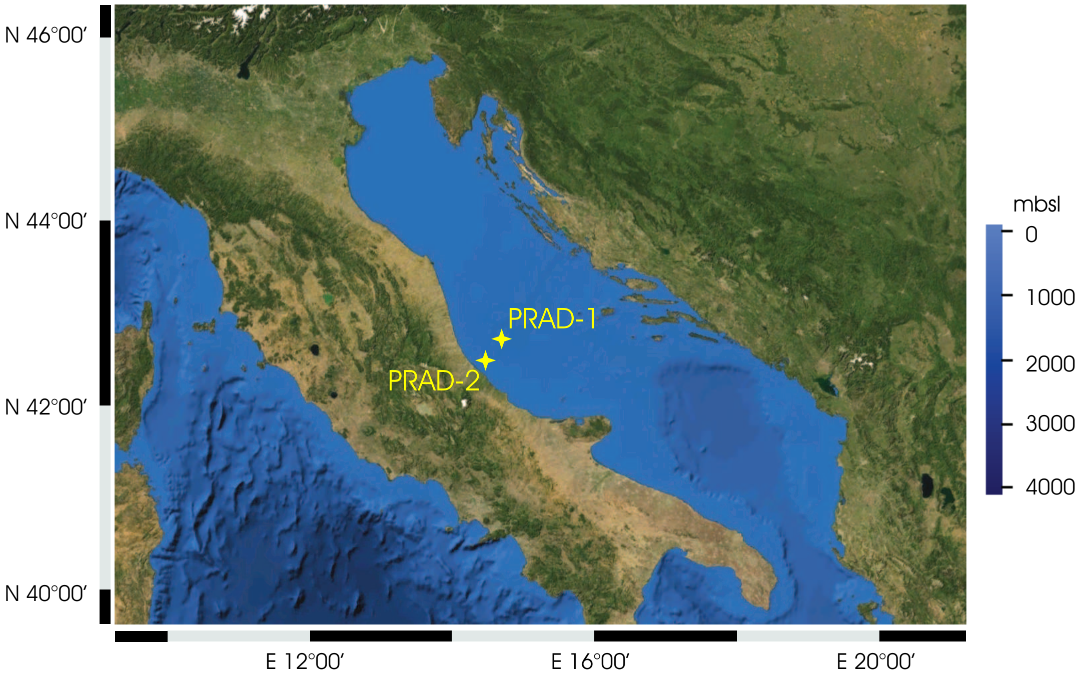


Figure 2.6: Adriatic Sea with drilling locations PRAD-1 and PRAD-2 north of the Gargano Peninsula. Satellite image from USGS (2000).

2.3 PROMESS-1 Project

In order to enhance the existing data base of the two above mentioned Mediterranean regions with ground-truth core and logging data, the European Community launched the PROMESS-1 project involving 12 universities and research institutions across Europe. In summer 2004, a 4-weeks cruise with an industrial drilling vessel was accomplished. During its course, boreholes at four locations were drilled and cored. In addition, geotechnical and wireline logging measurements were performed downhole. The drilled intervals were targeted to cover approximately the last 500 ka. PROMESS-1 was designed as a "source to sink"-approach to investigate two continental margins in the Mediterranean Sea and the associated sea-level changes, sediment fluxes, canyon history and slope failures. Scientific objectives included the investigation of

- Stratigraphic evidence of glacio-eustatic (glacial/interglacial) cycles and correlation with Milankovitch and Dansgaard-Oeschger scales;
- Depositional processes of sediments at the shelf, slope and deep-sea environment;
- Slope stability and the impact of regime and sediment supply variability on

slope failures;

- Segregation of eustatic changes from tectonic deformation at short time scales.

Results from classification of logging data acquired during PROMESS-1 contribute to the first two objectives. In the first case, stratigraphic sequences related to 100 ka glacio-eustatic cycles are identified. From the four drilled sites of the PROMESS-1 project, the most extensive data sets from downhole, XRF and MSCL logging have been recorded at sites PRGL-1 and PRAD-1. Hence, input data and classification will be presented for these two sites.

3 Data Acquisition

Geo-scientists seek to gather ground-truth data from the subsurface as accurate and comprehensive as possible. At times, non-invasive methods such as reflection seismics or swath bathymetry already deliver substantial data sets. However, it may also be crucial to get closer to the target under investigation. Drilling, coring and logging a borehole will provide the scientist with information impossible to obtain by remote techniques.

Intuitively, obtaining a core from a drilled hole looks like pulling all extractable data and information up to the surface. In fact, core data provide fundamental insights into the genesis and history of the sediments and rocks, as well as palaeontological, chemical and physical parameters. But retrieving a core and bringing it up to the surface means disturbing the sample and removing it from its natural equilibrium state. Temperature and pore pressure will irrevocably change at surface. Here, wireline logging inside the borehole can add valuable information by means of measuring *in situ* properties of the formation surrounding the hole.

During the PROMESS-1 project, at all drilled sites a full core recovery was attempted. Onboard, all cores were scanned by a containerised multi-sensor core logging (MSCL) unit owned and operated by IFREMER Brest. Magnetic susceptibility, gamma density and compressional wave velocity were recorded. After the cruise, the cores were split and one half was sampled by a fluorescence X-ray scanner (XRF) at the Bremen Core Repository. At sites PRGL-1 and PRAD-1, an extensive wireline logging suite was run, consisting of mainly open hole and few cased hole measurements. Hence, the logging data set comprises MSCL, XRF and wireline logging measurements. Because this set is exhaustive and of good quality the application and comparison of existing and new classification techniques were made possible. This enabled the investigation of the classifiers' behaviour on different training and testing data sets with real data. Unlike often published data with simple, simplified or synthetic geological settings, the data used here are from rather homogeneous marine Quaternary sediments (i.e. silty clay with clayey silt), which makes the classification task difficult and challenging at the same time. Simple data sets are no defiance for most classifying algorithms but delicate data sets like the ones presented are highly capable of discriminating classifiers by their performance.

In this chapter, data acquisition aspects such as principles of methods, resolution, accuracy issues, possible sources of errors and depth matching will be briefly discussed. This shall also emphasise the fact that real data are always noisy and imperfect and that this has to be kept in mind during all subsequent processing and classification steps. Tool output data are shown in Appendices A and B.

3.1 Core Logging

Coring operations onboard the PROMESS-1 drilling vessel *SRV Bavenit* utilised three kinds of corers, namely WIP Mk III, PISTON Mk III and Fugro Corer. The majority of coring in holes PRGL-1 and PRAD-1 was done by the former two with occasional supplement of the Fugro Corer. WIP and PISTON corers are operated by hydraulic oil pressure through an umbilical cable. The WIP corer takes a cylindrical sample by applying a monotonic thrust. The sample tube is fitted with a liner. The core is retained by a watertight ball valve in the sampler head, which causes suction if the sample moves downward. The PISTON sampler is similar except that a stationary piston is fitted in the tool. This piston forms a seal directly above the sample and enhances recovery especially in soft cohesive soils. The Fugro Corer is a free-fall device operated by mud pressure and retrieved by wireline. For soft sediments, the tool acts as a push sampler (as opposed to a percussion mode when used in hard soils). The pushing is achieved by building pressure behind a hammer at the top of the device. Pressure is attained by pumping the drilling mud down the string using the mud pumps (Fugro Engineers B.V., 2003).

All sample inner diameters (IDs) are 67 mm. The maximum stroke (i.e. length of a recovered core sample) of the WIP and PISTON corer is 95 cm and 85 cm, respectively. The Fugro Corer's maximum liner length is 188 cm. During PROMESS-1, stroke lengths were restricted to 80 cm for WIP and PISTON coring. The core liners of the Fugro Corer were divided into 100 cm and 88 cm cores.

After recovery, every core received top and end caps, was labelled and stored vertically at 13°C. Because of outgassing effects cores tended to expand. In extreme cases, end caps had to be perforated in order to avoid gas pressure building up inside the core liner.

3.1.1 Depth control

Despite extraordinarily good core recovery (above 95 % for either hole PRGL-1 and PRAD-1), cores may have been shifted or squeezed due to outgassing and missing intervals. Absolute depth match is therefore likely within the order of decimetres. Claiming a more precise depth accuracy may be questionable. To minimise depth offsets, the depth matching procedure consisted of two steps. First, a relative depth conversion was applied to convert "depth within one core" to "depth within the borehole". For this conversion, the following criteria applied (from Dennielou, pers. comm.):

- The top of a push corresponds to the top of a core;
- The sample retrieved during one push is the core plus the core shoe;
- The sampled length in a borehole during a sequence is the distance between the top of two consecutive pushs;
- If the length of the core plus core shoe is larger than the sampled length, the section is linearly squeezed to fit the sampled length;
- If the length of the core plus core shoe is smaller than the sampled length, the distance between the base of the core plus core shoe and the top of the next push is defined as lack of recovery;
- Voids described in the core (mainly due to outgassing) are excluded from the recovery length;
- The intervals have been resampled at 1 cm resolution.

Second, the entire resulting intrinsically corrected core interval was then depth-matched to the reference datum, the sea-floor. Here, the XRF measurement of Potassium (^{40}K) count rates was matched with the wireline spectral Gamma-ray count rates for Potassium. The latter curve had been previously tied in to the sea-floor level (see below).

3.1.2 Multi-Sensor Core Logging (MSCL)

Cores were scanned by the MSCL unit onboard the *SRV Bavenit* shortly after retrieval. The unit is basically a conveyor system that pushes each core section past several sensors. As for PROMESS-1 cores, these sensors were

- Rectilinear displacement transducers for core diameter measurements;
- Piezo-electric ceramic transducers for acoustic P-wave measurements;
- ^{137}Cs Gamma-ray source in a lead shield for Gamma-ray attenuation (bulk density) measurements;
- Loop and point sensors for magnetic susceptibility measurements;

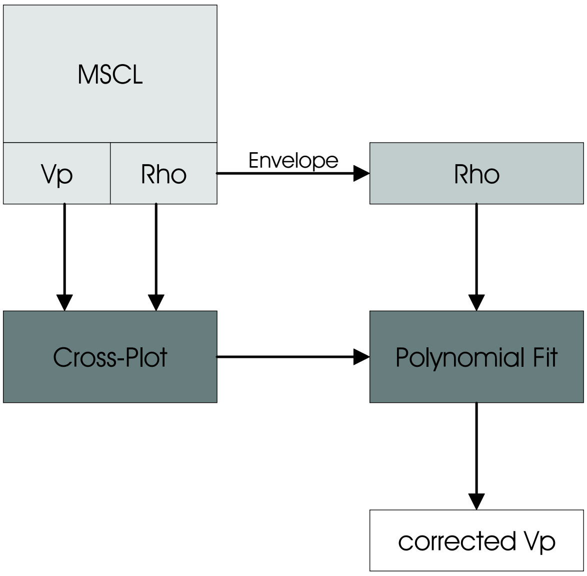


Figure 3.1: Sonic data were generated by utilising density data and a polynomial fit of a sonic-density cross-plot.

- Thermometer for core temperature (used for temperature correction).

All these measurements were performed on whole core sections with a sampling interval of 1 cm. During a subsequent sampling party, additional high resolution susceptibility measurements were carried out on split cores. All MSCL data were resampled at 5 cm resolution to match the wireline logging data. In terms of data processing, erroneous data such as voids, cracks or core gaps were removed. This is specifically true for the core extremities where sediment disturbance was commonly encountered.

Unfortunately, it turned out that sonic velocity data were severely affected by gas effects, i.e. the outgassing of cores after retrieving them from the subsurface. To establish a relationship between sonic velocity and density for the investigated area, a ρ vs. V_P cross-plot was generated. For a polynomial fit, it was decided to best use MSCL data from PRGL-1 (upper 21.5 m with still valid sonic data) and other PROMESS-1 boreholes, namely PRGL-2 (fraction of the data with most confidence) and PRAD-1 (upper 56.5 m). MSCL density data were corrected by extracting the upper envelope of the curve to remove any gas effects. A polynomial of second order was used to convert this envelope into sonic velocity values (see Figure 3.1). MSCL sonic data and this fit were spliced at 21.5 mbsf. Two major assumptions were thus made: V_P is correlated to the upper envelope of density; and the polynomial fit is sufficient for a mapping function $\rho \rightarrow V_P$.

3.1.3 X-Ray Fluorescence (XRF) Core Scanning

Split core halves of PRGL-1 and PRAD-1 were analysed by the XRF core scanner at the Bremen core repository. X-rays are generated by a 50 kV Molybdenum X-ray source and emitted to the sediment. A Peltier cooled Si-detector records count rates and energy of the backscattered X-rays. Spectral analyses yield relative abundance values (in cps, counts per second) for the elements K, Ca, Ti, Mn, Fe, Cu, Sr, V, Cr, Fe, Co, Ni, Zn and Pb. Ca/Fe and Sr/Ca ratios may be used to reveal detrital to biogenic compositional fluctuations in the sediments (Rothwell et al., 2005). Sample interval for PRGL-1 was 2 cm for the uppermost 49 m and 4 cm below. PRAD-2 was sampled at 2 cm resolution. All XRF data were resampled at 5 cm resolution to match wireline logging data. Again, erroneous data such as voids, cracks or core gaps were removed. Other possibly disturbed data points such as those close to the core top or bottom, at shells etc. were kept within the data set. However, smoothing of the data was performed prior to classification by means of a symmetrical, non-recursive filter.

3.2 Wireline Logging

Contrary to running memory sondes on slickline or LWD (Logging While Drilling) tools as part of the drill string (PCL, Pipe Conveyed Logging), wireline logging tools are conveyed on an electrical cable. One to seven inner electrical conductors are armoured by generally two outer layers of steel wire. This type of conveyance allows for fast and easy entry into the hole, good depth control and real-time data acquisition. On the other hand, good hole conditions are necessary to lower the tools all the way down to the total depth; if the hole is bridging or collapsing, no other force than the tools' weight will support the tool string to get down past the obstacle. With the cable being flexible, its weight does not contribute to that force. While logging up, a collapsed hole can still be logged as long as the maximum safe tension (defined by the cable strength) is not exceeded.

3.2.1 Depth Control

The tool's position relative to a reference datum can only be measured indirectly, i.e. by measuring the cable length between tool and a surface station. This is typically done by means of a wheel touching the wireline cable that rotates as the cable is lowered into or pulled out of the hole. With this kind of depth control the following issues have to be considered:

- **Cable stretch**. Provided with enough weight on its lower end, steel armoured cable will stretch. While the weight of the tool string itself may be rather small, the cable's own weight will eventually cause cable stretch. This stretch can be accounted for by either cable stretch charts, an empirical formula or by comparing a down-log vs. up-log near the bottom of the hole (TD, total depth).

- **Cable slack**. At a typical logging setup, the wireline cable has some slack in between the cable drum and the sheave wheels. The slack will vanish with tool depth and can be corrected for by measuring the surface cable length once when the tool is near the surface and once when it is near TD.
- **Cable slippage**. The cable may slip when touching the depth-control wheel. This can be recognised by using two wheels and only considering the faster one for depth control. The depth control equipment used during PROMESS-1 only features one wheel, therefore having no means to detect cable slippage.
- **Tool drag**. The cable may move at surface, while the tool downhole in fact is being held up at a bridge (on the way down) or dragged along the borehole wall. This can be easily seen on the cable tension meter. If the tool has a built-in accelerometer , this "jojo"-effect may be removed from the logs, but typically the logs are not corrected.

Nonetheless, wireline logging is one of the most accurate conveyance type in terms of depth control, and depth accuracy within a decimetre can be achieved.

During PROMESS-1, wireline logs were recorded moving up-hole. The drop of Gamma-ray count rates to zero at the seafloor was set as the reference datum (0 mbsf). All logs were tied in to this datum subsequently.

3.2.2 Mud System

All sondes measure either parameters of the borehole, the borehole wall or the formation surrounding the borehole. Unfortunately, this part of the formation is affected by the drilling process. The drilling mud enters the formation matrix and migrates from the borehole away, thus increasing the invaded zone (see Figure 3.2). The formation matrix acts as a sieve: suspended particles of the drilling mud will stick to the borehole wall, forming a mud cake, while the mud filtrate enters the formation and displaces the formation fluid. This complex system consisting of mud, mud cake and mud filtrate is difficult to account for when applying borehole corrections to the recorded data. When possible, the wireline sondes' target depth of investigation (DOI) is set to the virgin zone but often the tool actually measures parameters of the invaded zone. Most of the time it is even completely unknown which part of the

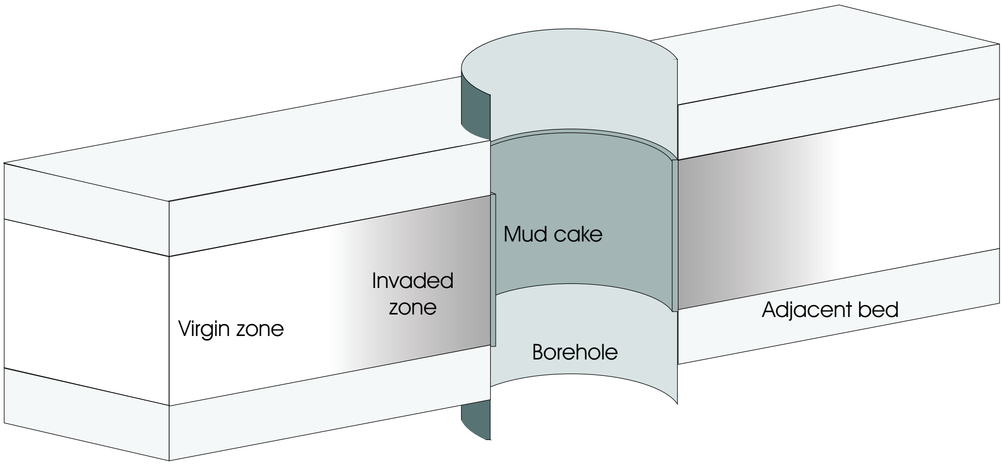


Figure 3.2: Mud invasion defines different zones of the mud system, the virgin zone being the most desirable one for measurements.

formation the data come from, as there exist only few tools that can give hints as to how deep the invaded zone penetrates the formation (e.g. array resistivity, if both mud and formation salinity is known). For small scale measurements such as micro-resistivity, the mud cake may affect data quality in the same way, forging the data that are meant to be recorded with an undisturbed borehole wall. Some sondes may overcome this issue by exerting an eccentralising force to push the sensors through the mud cake and into the formation.

At sites PRGL-1 and PRAD-1, the drilling mud used consisted of sea-water, salt water gel (attapulgite clay) as a viscosifier and PAC (Polyanionic Cellulose) to control possible fluid-loss and to stabilise the borehole. Dual-Laterolog measurements suggested that mud penetration was small and hence the mud cake was very thin (hinting at a low porosity regime as to be expected with marine clays). Therefore, the recorded data were considered to be unaffected by the drilling mud and no correction had been applied.

3.2.3 Tools

The following is a list of the wireline logging tools run during PROMESS-1. All tools record the data digitally and send them over the wireline to a PC based real-time acquisition system at the surface. The tools are not combinable but have to be run one-by-one. Thus, depth matching and preparation of composite logs have to be carried out with utmost care. A summary of logging parameters and technical details are given in Appendix C. What follows is a short overview of the principles and physics behind the measurement types and their technical implementation. It is not the aim of the author to give a thorough tract of each and every tool. For more details, see e.g. Serra (1984).

Calliper Sonde (CALI)

The calliper tool measures the borehole diameter by means of a calliper arm pressed against the borehole wall. The measurement is uni-directional, i.e. only the diameter in one direction is measured. When the borehole is oval, the calliper reading is assumed to represent the major axis of the ellipse, i.e. the largest possible distance. Diameter readings give information about borehole stability, break-outs, zones of swelling clay or fractures. It can also be used to locate the casing shoe or drill bit.

Spectral Gamma-Ray Tool (SGR)

Naturally occurring radioactive elements in the subsurface formations are primarily Thorium (Th), Potassium (K) and Uranium (U). As part of their disintegration process, Gamma-rays are emitted into the surrounding area. These Gamma-rays are subjected to further interactions, namely Compton scattering and photoelectric absorption. Before they are captured by an atom, they can be counted by means of a crystal scintillation detector (e.g. Bismuth-Germanium oxide, BGO) that converts their energy into a light flash (photoelectric effect). These photons can be converted into an electrical current by a photomultiplier tube and amplifier circuit. The current is dependent on the Gamma-ray's energy, hence energy spectra can be recorded. As Th, K and U all have different energy spectra, the three elements can be discriminated. The tool outputs four parameters: Th, K, U and total count rates. Total count rates are then converted into API units, i.e. into a calibrated comparable output measure based on the natural radioactivity at the American Petroleum Institute's test pit in Houston. Gamma-ray emission is statistical by nature, hence at low count rate levels features may not be seen on the log. Reducing the logging speed may help to increase overall count rates. The above radioactive elements tend to concentrate in shales and volcanic rock rather than in sandstones, but there are also reported high Gamma-ray responses in mineralogically immature sandstones. The ratios Th/K and Th/U indicate changes in clay composition. Also, high Th/U ratios may suggest volcanic ashes. As happened during PROMESS-1, the sonde may also be used inside the drill string or casing, but then count rates are severely diminished. An ENCOR correction algorithm (Hendriks, 2003) for this setting was applied that requires prior calibration and re-processing of the recorded curves .

Geochemical Tool (EBS)

The geochemical or "Neutron-Gamma-" tool records the formation Gamma-ray response after being irradiated by neutrons. Relative element occurrence can be deducted from the tool. Changes in the chemical composition of the formation are easily detected and may give hints on mineral structure, provenance or weathering. When emitting high energy neutrons into the formation, inelastic scattering occurs: neutrons will collide with formation atoms and kinetic energy will be transformed into gamma rays. Eventually, the neutrons will be captured by formation atoms. During this capture process, gamma radiation will be set free as well. The energy spectrum of these gamma rays is dependent on the colliding atoms. Energy levels of gamma rays created during inelastic scattering can be associated with carbon and hydrogen. Likewise, energy levels of gamma rays created during thermal absorption can be associated with silicon, calcium, iron and oxygen. By means of two gamma-ray scintillation detectors, two spectra are recorded and give quantitative measures of the above elements. Similar to the SGR sonde, this is a statistical measurement and may thus be affected by errors when count rates are low.

Dipmeter Tool (DIP)

The main data output of the Dipmeter tool are deviation, relative bearing and azimuth of the borehole (sonde). Also recorded are the angles and dips of formation beds intersecting the borehole. These are based on micro-resistivity of the borehole wall recorded by 4 buttons at 0°, 90°, 180° and 270° positions around the sonde. The four micro-resistivity values can also be smoothed and used to generate a mean micro-resistivity curve. This curve is neither calibrated nor output in Ωm units. It nevertheless follows the conventionally recorded laterolog resistivity but its sampling rate is much higher (5 mm). For this reason and because a calibrated unit scale is not required for the classification task the mean micro-resistivity curve was used for subsequent processing instead of standard laterolog resistivity data.

Dual Laterolog Resistivity Tool (DLL)

The Laterolog principle of measurement is to confine voltage and current in a fixed region such that the resistance can be computed. Electrical resistance together with a geometric factor yields resistivity. Technically, the tool directs a current beam into the formation. This beam is focused and forced into the formation by bucking currents. They create an equipotential surface, preventing the measure current to flow up the borehole. The minimum electrode configuration for a Laterolog tool consists of two voltage measuring electrodes and one current electrode. Generally, regional changes of formation resistivity are due to changes of pore fluid rather than matrix resistivity. Where porosity and pore fluid resistivity are constant, changes of matrix resistivity are likely being reflected in a Laterolog recording. Conductive drilling mud affects the measurement that needs to be corrected. During PROMESS-1 with conductive salt-water based drilling mud, such correction was not applied as there had been no means to determine the mud salinity onboard. All resistivity data are thus presented in uncalibrated Ωm units. The DLL tool provides two output curves (hence the name dual): a deep and a shallow reading. The difference is their bucking currents and thus their depth of penetration. Usually, the deep reading is closer or equal to the true formation resistivity R_t, while the shallow reading may have been recorded within the invaded zone and hence be affected by the mud. Within this work only the deep reading is used.

Micro-Susceptibility Tool (MS)

Magnetic susceptibility is defined as the degree of magnetisation of a formation in response to a magnetic field. This ability of a sediment or rock is controlled by the amount of magnetic minerals within. Generally for sediment deposits, glacial periods are characterised by sediments with high concentration of magnetic minerals, where interglacials have low concentrations. Hence, climate variability may be deducted from a susceptibility log. In order to measure the magnetic susceptibility of the surrounding formation, this tool utilises three coils: one transmitter, one receiver and one compensating coil. The transmitter coils sends a (primary) electromagnetic field into the formation which in turn produces a secondary electromagnetic field. This field, together with the primary one, induces a current in the receiver coil. The compensating coil will cancel out the effect of the primary field on the receiver coil; the imaginary part of the remaining signal is proportional to the formation's magnetic susceptibility.

4 Classification Techniques

The principle of dividing any given quantity into several groups (a rather rude description of the classification task) seems to be so naturally and unconsciously applied to every-day tasks that in fact the roots of pattern recognition are difficult to reveal. There is historical tradition of classification by Bodhidharma, Plato and later Aristotle. In a modern mathematically-founded sense of classification, Fisher's discriminant analysis (Fisher, 1936) developed in the 1930s, started the development of algorithms and learning machines that nowadays include e.g. maximum likelihood estimation, artificial neural networks or support vector machines.

As already mentioned in Chapter 1, classification by means of supervised learning requires a training data set, that is, data samples that have been assigned to (correct) class labels (see Figure 1.2). Based on these data, the classification algorithm is trained until a discriminating (or decision) rule has been established. The remaining data samples (termed test data) are then classified by this trained algorithm (Figure 4.1). In a more formal notation, the data samples are composed of a variable vector x whose elements are, for instance, the values of several log curves at a given depth. For the learning data set, each variable vector is accurately assigned to one of k classes:

$$\mathbf{x}_i \rightarrow c_j \qquad \forall\, i = 1,..,n \quad \text{and} \quad j = 1,..,k.$$

where n is the number of training samples. The classification task is then to determine

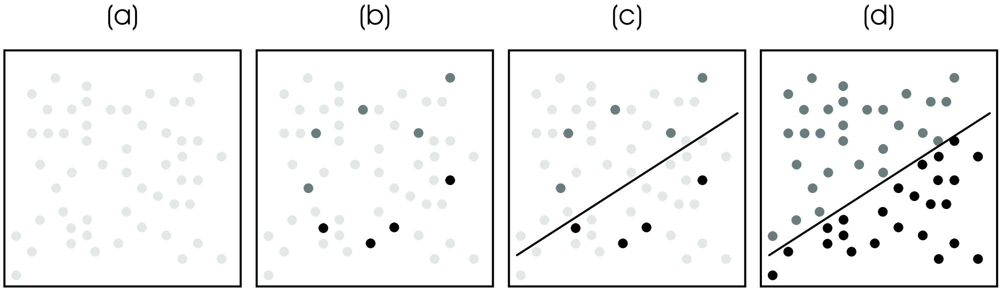


Figure 4.1: Visualisation of the learning process: (a) Complete (2-D) data set. (b) Training data set for the 2-class case (class 1 = dark grey, class 2 = black). (c) Learning of the decision boundary. (d) All other sample points are classified according to the decision rule (generalisation).

or to approximate the underlying mapping function

$$f : \mathbf{x}_i \rightarrow c_j \qquad \forall\, i = 1, .., n \quad \text{and} \quad j = 1, .., k$$

for all data samples for which the proper class labels are known (training data set). The function is then applied to all other samples with unknown class labels (test data set). This principle remains the same for all classification algorithms (or *classifiers*).

This chapter will briefly explain some of the existing algorithms and their differences. It is intended to give an overview rather than an exhaustive tract of derivations. References to more detailed essays are given in the text. The focus here clearly is on basic principles and applicability of the algorithms.

4.1 Pre-Processing

All available wireline logging, MSCL and XRF data were verified in terms of gross outliers, missing data points and drifts. Log curves to be used with the learning algorithms were then selected. Outliers most likely due to borehole conditions were omitted and replaced by an interpolation of adjacent values. Missing data were not observed. Susceptibility data showed a temperature drift. Sonic velocities increased with depth due to compaction effects. Both trends were removed under the assumption that they are linear.

4.1.1 Data distribution

Strictly speaking, some of the subsequently used learning algorithms require the input data to be Gaussian (normally) distributed. Even in cases where this prerequisite is mathematically not necessary, Gaussian distribution is assumed, as it is the most common distribution and very well studied in literature.

Input data were examined if they were normally distributed. Electrical resistivity and magnetic susceptibility were tested for a log-normal distribution (i.e. the tests were applied to the natural logarithm of the respective values). For the univariate case (i.e. treating each log curve separately), the standard tests of distribution are χ^2 (chi-square) and Kolmogorov-Smirnov. Both tests calculate a score based on the the null hypothesis that the values come from a standard normal distribution. If the score falls below a critical value, the hypothesis cannot be rejected, i.e. the data are Gaussian distributed. The χ^2-test is susceptible to deviations from the standard distribution especially at the distribution margins where frequencies are low. Here, the Kolmogorov-Smirnov is more stable and gives a second opinion (Trauth, 2003).

4.1.2 Normalisation

Normalisation is the scaling of data within a defined range. The input data in the discussed case of logging curves consist of different value ranges in different units, e.g. for Iron $3000 - 5450$ counts per seconds or susceptibility $35 - 800 \times 10^{-6}$ SI units. For the purpose of classification, all input log curves were normalised with respect to the interval $[0, 1]$ using

$$x_{i_n} = \frac{x_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$$

where x_{i_n} is the normalised log value, x_i the actually measured value and $\mathbf{x}$ the log curve vector with all x log values. The normalisation ensures evenly weighted input vectors and faster computing performance.

4.1.3 Dimensionality Reduction

When dealing with classification algorithms using large data sets with many input variables, this excessive dimensionality leads to prohibitive complex calculations, computing times and storage. Specifically, the need for a large number of training samples grows exponentially with the dimension of input space (Duda et al., 2001). This fact is also known as the *curse of dimensionality* and may seriously handicap the application of classifiers.

One approach to cope with the problem of dimensionality reduction is the linear combination of input variables. Beside other methods such as principal component analysis (PCA), projection pursuit (PP) and multiple discriminant analysis (MLA), factor analysis (FA) aims to reveal simple underlying structures within data based on similarities between observations (Davis, 2002). For the dimensionality reduction in the case of PROMESS-1 boreholes, input data space is not excessively high-dimensional with only 18 log curves for PRGL-1 and 17 curves for hole PRAD-1. In pattern recognition problems, typical input spaces have up to many hundreds of dimensions. Nonetheless, a factor analysis was performed on the normalised data sets of PRGL-1 and PRAD-1. This method not only reduces input space dimensionality but also may reveal mutual uncorrelated hidden variables (*factors*) controlling the visible input variables.

Among several methods to perform a factor analysis, the approach already applied to borehole data by Bücker et al. (2000) was used as follows. Starting with a $d \times n$ input data matrix, eigenvalues and eigenvectors were extracted from its standardised $d \times d$ variance-covariance matrix, d being the number of log curves and n the number of log values. The eigenvalues λ are set as the diagonal elements of matrix $\mathbf{\Lambda}^2$ and with eigenvector matrix $\mathbf{U}$,

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}$$

defines the factor matrix with its elements called *factor loadings*. The sum (for each variable) of the squared factor loadings yields totals called *communalities*. They describe the share of variance of the respective variable that is explained by the factors and hence link each variable to one factor. Only factors with eigenvalues >1 are considered as being meaningful. Factors with smaller eigenvalues do not contain more information than a single input variable and are therefore discarded. Moving the variables closer to the factor axes (in input space) may allow the deduction of a meaning of the factors. This goal can be achieved by a transformation called varimax rotation which is applied after a Kaiser normalisation to promote variables with higher communalities (for details see Davis, 2002). Unfortunately, factor analysis yielded unsatisfactory results (see Chapter 5) and the classification tasks were carried out with the original log curves.

4.1.4 Class Labels

With supervised learning, the classification algorithm is fed with a training data set that consists of input data vectors and class labels. In the discussed case of classifying log curves into stratigraphic sequences, every measuring point is represented by a vector of 18 log values (at PRGL-1) and 17 log values (at PRAD-1), respectively. In addition, the training data set includes a number $1, ..., k$ for each measurement point, where k is the number of sequences to be classified. The goal of each classifier is then to predict this *label* for each measurement point of the test data set.

4.2 Bayesian Decision Theory

Prior to classifying a given sample into one of k classes $c_1, ..., c_k$, there is an *a priori* or prior probability $P(c_i), i = 1, ..., k$ that the sample belongs to class c_i (e.g. because there are more samples of class c_1 than there are of class c_2). When classifying the same sample on the basis of a continuous random p-dimensional input vector $\mathbf{x}$ (where $\mathbf{x} = (x_1, ..., x_p)$; e.g. Gamma-ray, sonic velocity and susceptibility measurements at a given depth), both distribution and probability density function (PDF) of $\mathbf{x}$ depends on the class it belongs to. The latter is therefore called *class-conditional probability density function* $p(\mathbf{x}|c_i)$ for vector $\mathbf{x}$ given it belongs to class c_i.

Together with the prior probability it is used to determine the probability that the appropriate class is c_i given that the input vector $\mathbf{x}$ has been measured. This is called the *a posteriori* or posterior probability and can be expressed as

$$P(c_i|\mathbf{x}) = \frac{p(\mathbf{x}|c_i)P(c_i)}{p(\mathbf{x})}. \tag{4.1}$$

Equation 4.1 is called *Bayes* formula after the Reverend Thomas Bayes (1702–1761). $p(\mathbf{x})$ is termed evidence factor and is a mere scale factor ensuring that the posterior probabilities sum to unity. It is

$$p(\mathbf{x}) = \sum_{i=1}^{k} p(\mathbf{x}|c_i)P(c_i).$$

Classifying an input vector into a (possibly false) class is termed *action*. With a finite set of l possible actions $\{a_1, ..., a_l\}$ and the above set of k classes $\{c_1, ..., c_k\}$, the loss function $L(a_j|c_i)$ defines the loss incurred for taking action a_j when the correct class is c_i. It quantifies the error of classification. The expected loss for taking action a_j when observing variables $\mathbf{x}$ is

$$R(a_j|\mathbf{x}) = \sum_{i=1}^{k} L(a_j|c_i)P(c_i|\mathbf{x}). \tag{4.2}$$

$R(a_j|\mathbf{x})$ is called *conditional risk*. The overall risk (or error) R can be minimised by taking action a_j for which the conditional risk $R(a_j|\mathbf{x})$ is minimum. The resulting minimum overall risk is called *Bayes risk* R^* and defines the best performance achievable.

To minimise the probability of any classification error that class should be chosen that minimises the overall risk and hence maximises the posterior probability. Bayes' decision theory is coherent and powerful in the sense that it allows to calculate this posterior probability of class c_i given a measured input vector $\mathbf{x}$ only from prior probabilities $P(c_i)$ (which are typically known) and the class-conditional PDFs $p(\mathbf{x}|c_i)$. Generally, the latter are the only unknown term in Equation 4.1 for many classification applications (Duda et al., 2001). In few cases, they are multivariate Gaussian distributions with known mean vectors and covariance matrices, sometimes they can be assumed to be multivariate Gaussians but the parameters are unknown, and often, conditional densities are altogether unknown. For each case, varying algorithms exist that require different assumptions about the conditional densities (Hastie et al., 2001).

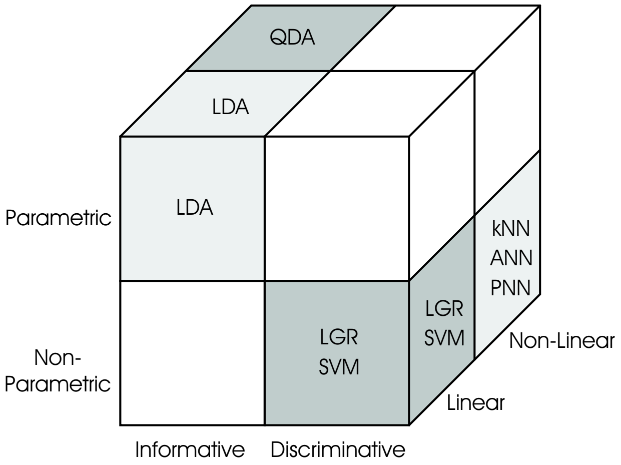


Figure 4.2: Block diagram of presented classification algorithms. The methods can be separated into parametric vs. non-parametric, informative vs. discriminative or linear vs. non-linear groups.

4.3 Algorithms

Among the several methods, there are groups of classifiers that can be distinguished. There are linear and non-linear, parametric and non-parametric, or informative and discriminative classifiers (Figure 4.2), of which some have been chosen to be adjusted to and applied to the logging data of PROMESS-1 boreholes.

The first distinguishing feature of classification algorithms is that between *informative* (also called *generative*) classifiers, and *discriminative* classifiers. Informative classifiers model the class-conditional densities $p(\mathbf{x}|c_i)$ and then use Bayes formula (Equation 4.1) as described above to determine the posterior probability $P(c_i|\mathbf{x})$. In contrast, discriminative classifiers model the decision boundaries or the class membership probabilities directly without considering any underlying class-conditional densities. Because this approach requires the consideration of all classes at the same time, these types of classifiers are more difficult to train (Rubinstein & Hastie, 1997). For a comparison between the two methodologies see Bouchard & Triggs (2004).

Regarding informative classifiers, the estimation of $p(\mathbf{x}|c_i)$ can be accomplished by estimating the parameters of these class-conditional densities (such as mean and variance in case of a Gaussian distribution). This is done by so-called parametric techniques like discriminant analysis. Contrary to this, non-parametric techniques have to be used when there is no prior knowledge of the parameters of the underlying density $p(\mathbf{x}|c_i)$. For instance, one approach is to estimate the density functions $\hat{p}(\mathbf{x}|c_i)$ from a sample data set. If satisfactory, these estimates can substitute the true class-conditional densities. This is the principle of Parzen windows density estimation. For more details see e.g. Duda et al. (2001) or Hastie et al. (2001).

Finally, the various algorithms can be distinguished by their shape of decision boundary. There are linear classifiers such as LDA and SVM, or other, such as neural networks, that are so-called non-linear algorithms.

A short presentation of algorithms and classification tools follows. Although maximum likelihood and Parzen windows density estimation are not restricted to classification problems only, they are included because their principles are often used in pattern recognition tasks.

4.3.1 Maximum-Likelihood Estimation

As outlined before, the difficult task in Bayes decision theory is to determine the class-conditional densities $p(\mathbf{x}|c_i)$. Assuming that the densities are known (e.g. Gaussian density), but their parameters are not (e.g. mean and covariance), the problem of estimating an unknown function $p(\mathbf{x}|c_i)$ is simplified to one of estimating its parameter vector $\boldsymbol{\theta}_i$. For the multivariate Gaussian case $\boldsymbol{\theta}_i$ would consist of a mean vector $\boldsymbol{\mu}_i$ and a covariance matrix $\boldsymbol{\Sigma}_i$. Two commonly applied procedures that accomplish this parameter estimation based on training samples from supervised learning are Bayesian parameter estimation and maximum-likelihood (ML) estimation. Bayesian parameter estimation considers $\boldsymbol{\theta}$ to be a random variable, whose distribution can be recursively converted into a posterior probability density by use of training data. ML estimation on the other hand considers $\boldsymbol{\theta}$ to be a fixed parameter vector and finds the parameter vector that is best for the given training data. For computational and interpretational reasons ML is often preferred to Bayesian parameter estimation which in turn is supported by theoretical and methodological arguments (Duda et al., 2001). Therefore, out of these two methods, ML parameter estimation has been chosen to be shortly outlined. It will later be used relating to discriminant analysis and logistic regression.

ML estimation assumes that $p(\mathbf{x}|c_i)$ has a known parametric form determined by its parameter vector $\boldsymbol{\theta}_i$. For the multivariate Gaussian case, $\boldsymbol{\theta}_i$ consists of the components of $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$. For each of $c_1, ..., c_k$ classes, there is a data set $\mathcal{S}$ with n i.i.d. (independent and identically distributed) random samples $\mathbf{x}_1, ..., \mathbf{x}_n$. Then the likelihood of $\boldsymbol{\theta}$ with respect to the set of samples is

$$p(\mathcal{S}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathbf{x}_i|\boldsymbol{\theta}).$$

The maximum-likelihood estimate of $\boldsymbol{\theta}$ is that value $\hat{\boldsymbol{\theta}}$ that maximises $p(\mathcal{S}|\boldsymbol{\theta})$. It can be shown (e.g. Duda et al., 2001) that for the multivariate Gaussian case, the ML estimate of the mean

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{4.3}$$

is the mean of the sample and the ML estimate for the covariance matrix is

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^{\mathrm{T}} \tag{4.4}$$

which indeed is, in the words of Moore (2001), an "ultimately unsurprising fact".

4.3.2 Discriminant Analysis

One of the most commonly applied classification procedures is the discriminant analysis. For a classification problem of assigning an input vector $\mathbf{x}$ to one of k classes $c_1, ..., c_k$, there is a set of k discriminant functions $g_i(\mathbf{x})$. Each discriminant function transforms any input vector $\mathbf{x}$ into a discriminant score. That class c_i is assigned to input vector $\mathbf{x}$ that yields the highest discriminant score. To minimise the risk of misclassification, Bayes formula (Equation 4.1) is used to relate the maximum discriminant function $g_i(\mathbf{x})$ to the maximum posterior probability:

$$g_i(\mathbf{x}) = P(c_i|\mathbf{x}).$$

It can be shown (e.g. Hastie et al., 2001) that this is equal to

$$g_i(\mathbf{x}) = \ln p(\mathbf{x}|c_i) + \ln P(c_i).$$

In the case of multivariate Gaussian density (in d dimensions) given by

$$p(\mathbf{x}|c_i) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_i|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)},$$

the discriminant functions are now

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(c_i). \tag{4.5}$$

The special case where the covariance matrices for all classes are identical ($\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}\,\forall\, i$) is called linear discriminant analysis (LDA), because Equation 4.5 can be simplified to

$$g_i(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\boldsymbol{\mu}_i^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln P(c_i) \tag{4.6}$$

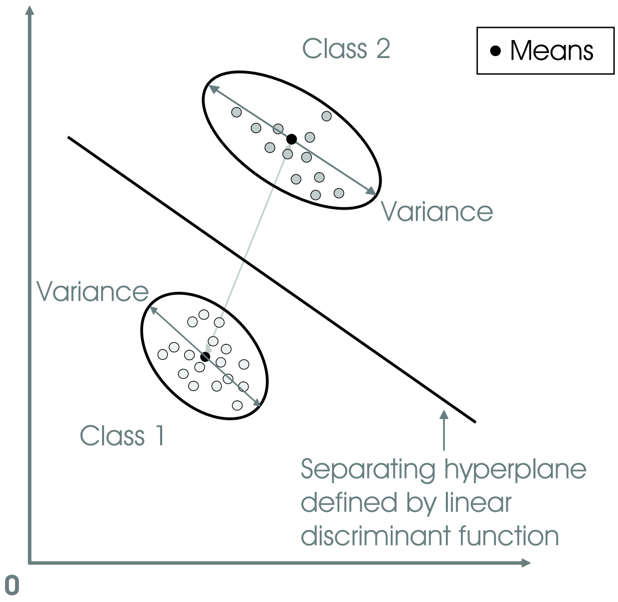

Class 2
• Means
Variance
Variance
Class 1
Separating hyperplane defined by linear discriminant function
0

Figure 4.3: Principle of linear discriminant analysis (LDA). LDA aims to minimise the within-class variances and to maximise the inter-class means at the same time.

which are linear in $\mathbf{x}$. All other terms of Equation 4.5 are independent of i and therefore constant. Visually, Equation 4.6 can be interpreted such that to maximise the posterior probability, the distances between the class means are maximised while at the same time the variances for each class are minimised. This is illustrated in Figure 4.3 for the 2-D case.

For the more general case with different covariance matrices for each class, only the $(d/2\ \ln 2\pi)$-term of Equation 4.5 can be dropped, yielding

$$g_i(\mathbf{x}) = -\frac{1}{2}\ln|\mathbf{\Sigma}_i| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \ln P(c_i)$$

which are quadratic in $\mathbf{x}$ and hence denominate the quadratic discriminant analysis (QDA). For both LDA and QDA, the parameters of the multivariate Gaussian distributions are unknown and have to be estimated from training data for each of k classes, $i = 1, ..., k$:

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_i &= \frac{1}{n_i}\sum_{j=1}^{n_i}\mathbf{x}_j \\
\hat{\mathbf{\Sigma}}_i &= \frac{1}{n_i}\sum_{j=1}^{n_i}(\mathbf{x}_j-\hat{\boldsymbol{\mu}}_i)(\mathbf{x}_j-\hat{\boldsymbol{\mu}}_i)^{\mathrm{T}} \\
\hat{P}(c_i) &= n_i/N
\end{aligned}$$

where n_i is the number of training data vectors of class i and N the number of all training data vectors. Note the recurrence of Equations 4.3 and 4.4.

Though the principles that LDA and QDA are based on, are rather simple, both techniques are widely used and have an excellent reputation. Michie et al. (1994) report a remarkable performance of LDA and QDA compared to other classifiers as do Benaouda et al. (1999) and Lee et al. (2005). Hastie et al. (2001) suggest that this may be credited to the fact that many data sets support only simple decision boundaries which can be sufficiently estimated by Gaussian distributions.

4.3.3 Logistic Regression

Logistic regression aims to model the posterior probabilities $P(c_i|\mathbf{x})$ from the Bayes formula (Equation 4.1) for each of k classes $c_1, .., c_k$. It does so by means of a set of linear functions such that the probabilities sum to one and remain in $[0;1]$:

$$\sum_{i=1}^{k} P(c_i|\mathbf{x}) = 1 \qquad \text{and} \qquad P(c_i|\mathbf{x}) \in [0;1]. \tag{4.7}$$

The reason for these conditions are that the dependent variables of classification are the class labels (here: 0 and 1 for the two-class case) and that any data sample must be allocated to one of the possible classes. The linear equations are of the form

$$\begin{aligned} f_i(\mathbf{x}) &= w_{i0} + \sum_{j=1}^{p} w_{ij} x_j \\ &= b_i + \mathbf{w}_i^{\mathrm{T}} \mathbf{x} \qquad \forall\, i = 1, ..., k \end{aligned} \tag{4.8}$$

where p is the dimension of input space ($\mathbf{x} = x_1, ..., x_p$), w_{i0} are the biases (here meaning offsets and simply denoted as b_i) and $\mathbf{w}_i$ are the (unknown) model coefficients. Using the logistic function

$$l(y) = \frac{1}{1 + e^{-y}}$$

and Equation 4.8, the logistic model is defined as

$$P(c_i|\mathbf{x}) = \frac{1}{1 + e^{-(b_i + \mathbf{w}_i^{\mathrm{T}} \mathbf{x})}} \qquad \forall\, i = 1, ..., k.$$

A transformation function called *logit transformation*

$$\text{logit}\; z = \ln \frac{z}{1 - z}$$

is applied to ensure conditions 4.7 and yields

$$\ln \frac{P(c_i|\mathbf{x})}{1 - P(c_i|\mathbf{x})} = b_i + \mathbf{w}_i^{\mathrm{T}} \mathbf{x} \qquad \forall\, i = 1, ..., k.$$

The left side of the term is also called *log odds ratio*. This logistic regression model is generally fit by maximum likelihood (see Section 4.3.1) estimating the unknown coefficients b_i and $\mathbf{w}_i$. ML maximises the probability of getting the observed results (from the training samples) given the fitted regression coefficients $\mathbf{w}_i$. As the ML estimates need neither to exist nor to be unique, often regularisation methods are applied such as ridge regression. For the general multi-class case, the mathematical details become somewhat substantial and their full tract is not the intention of this work. The interested reader is pointed to detailed papers such as Zhu & Hastie (2004) or Fort & Lambert-Lacroix (2005) and references therein.

Hastie et al. (2001) state that the models of logistic regression and linear discriminant analysis are the same and only their estimations of the linear coefficients are different. They remark that logistic regression tends to be more robust than LDA as it relies on fewer assumptions (such as the underlying probability density distribution and identical covariance matrices). Even if LDA is used in cases violating those assumptions, the two methods are reported to yield similar results.

4.3.4 Parzen Windows

In order to estimate the density at a given point $\mathbf{x}$, a region $\mathcal{R}$ around $\mathbf{x}$ is introduced. n samples $\mathbf{x}_1, ..., \mathbf{x}_n$ are i.i.d. drawn according to the density function $p(\mathbf{x})$. A sequence of regions $\mathcal{R}_1, ..., \mathcal{R}_n$ is created with $\mathcal{R}_1$ containing one sample, $\mathcal{R}_2$ containing two samples, and so on, until finally all n samples fall into $\mathcal{R}_n$. It can be shown that the nth estimate for the density function $p(\mathbf{x})$ is given by

$$\hat{p}_n(\mathbf{x}) = \frac{k_n}{nV_n} \tag{4.9}$$

where k_n are the number of samples falling into $\mathcal{R}_n$ and V_n is the volume of $\mathcal{R}_n$. The following three conditions have to be fulfilled in order to ensure that $\hat{p}_n(\mathbf{x})$ converges to $p(\mathbf{x})$:

$$\begin{aligned} &\lim_{n\to\infty} V_n = 0 \\ &\lim_{n\to\infty} k_n = \infty && (4.10) \\ &\lim_{n\to\infty} k_n/n = 0. && (4.11) \end{aligned}$$

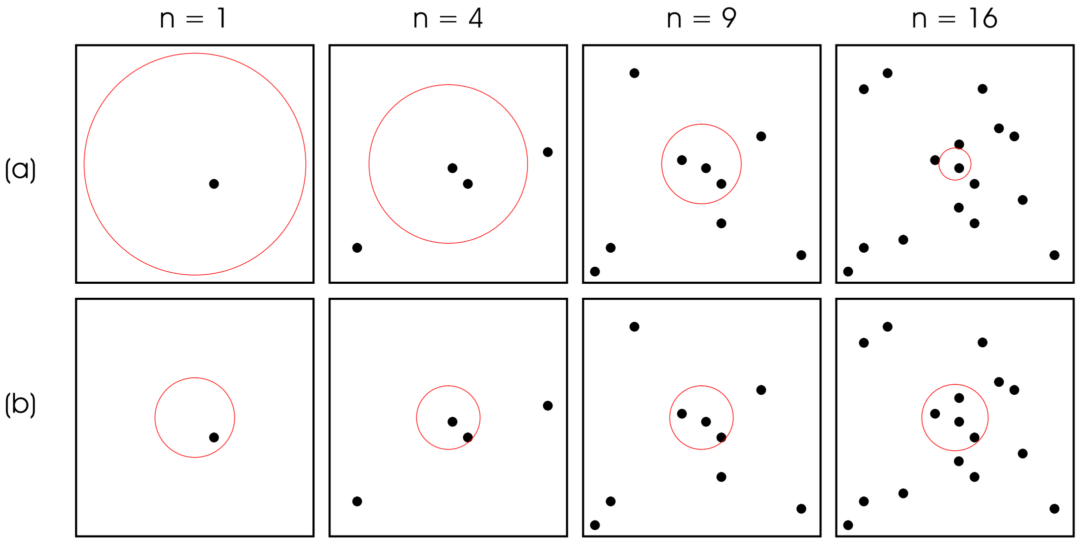


Figure 4.4: Parzen windows density estimation and kNN. (a) Parzen window region $\mathcal{R}_n$ of volume V_n is shrank as a function of n (here: $V_n = 1/\sqrt{n}$). (b) k-nearest neighbour adjusts the volume data-dependent to encircle a given number k_n of samples (here: $k_n = \sqrt{n}$). Figure after Duda et al. (2001).

To obtain a sequence of regions that satisfy these conditions, the Parzen window method shrinks an initial region by declaring the volume V_n as a function of the number of samples n (Figure 4.4). This is done by introducing a window function or *kernel* $K_\lambda(\mathbf{x}, \mathbf{x}_i)$ with window width λ such that

$$\hat{p}_n(\mathbf{x}) = \frac{1}{n\lambda} \sum_{i=1}^{n} K_\lambda(\mathbf{x}, \mathbf{x}_i).$$

A popular choice for K_λ is the Gaussian kernel

$$K_\lambda(\mathbf{x}, \mathbf{x}_i) = \varphi\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{\lambda}\right) \tag{4.12}$$

that can be extended to the Gaussian product kernel with which $\hat{p}_n(\mathbf{x})$ in $\mathbb{R}^d$ is then

$$\hat{p}_n(\mathbf{x}) = \frac{1}{n(2\pi\lambda^2)^{d/2}} \sum_{i=1}^{n} e^{-\frac{1}{2}\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|}{\lambda}\right)^2}.$$

The Parzen density estimate acts as the smoothed equivalent of the local average. The window function is basically an interpolation, with each sample $\mathbf{x}_i$ contributing to the estimate $\hat{p}_n(\mathbf{x})$ dependent on its distance from $\mathbf{x}$.

The decision regions of a Parzen windows classification algorithm depend on two parameters: on the choice of the window (kernel) function and on the window width λ. Recalling the nature of non-parametric techniques, there is no information available about the underlying distribution. In absence of any such information, the use of a generic window shape such as the Gaussian kernel function is plausible. However, no window width λ is superior to another. Therefore, the classification process must be validated in order to establish an appropriate value for λ. The disadvantage of non-parametric methods to require a large number of samples worsens exponentially with the dimensionality in input space. In order to trade space complexity for time complexity, artificial neural networks may be used to implement this kind of classifiers. The neural network realisation of the Parzen windows method is the probabilistic neural network (PNN). It will be discussed in Section 4.3.6.

The densities for each class are estimated separately and the proper class labels are assigned according to the maximum posterior probability. Adapting Equation 4.9 for the class-conditional case yields

$$\hat{p}_n(\mathbf{x}, c_i) = \frac{k_i/n}{V}$$

with a cell of volume V placed around $\mathbf{x}$ that encloses k samples of which k_i samples belong to class c_i. Following Bayes decision rule (Equation 4.1), the estimate for $P(c_i|\mathbf{x})$ is

$$\hat{P}_n(c_i|\mathbf{x}) = \frac{\hat{p}_n(\mathbf{x}, c_i)}{\sum_{j=1}^{c} \hat{p}_n(\mathbf{x}, c_i)} = \frac{k_i}{k} \tag{4.13}$$

which is simply the fraction of those samples in that cell that belong to class c_i. The performance of this classification algorithm approaches the best possible given that there are enough samples and that the cell is sufficiently small (Duda et al., 2001). Regarding the size of the cell, its volume V_n is chosen as function of n. As n approaches infinity, an infinite number k of samples will fall into an infinitely small cell of volume V_n.

Another approach would be to expand V_n until it encircles some specified number of samples k. This is exactly the way the k-nearest neighbour rule works, which follows next.

4.3.5 k-Nearest Neighbour Estimation

To overcome the problem of determining the (unknown) best window function K_λ of the Parzen window approach (section 4.3.4), the cell volume V can be made a function of a subset of training data samples rather than of the overall number of samples n. Hence, to estimate $p(\mathbf{x})$ from n samples, a cell of volume V is centred around $\mathbf{x}$ and grown until it captures k samples, k being a function of n (Figure 4.4).

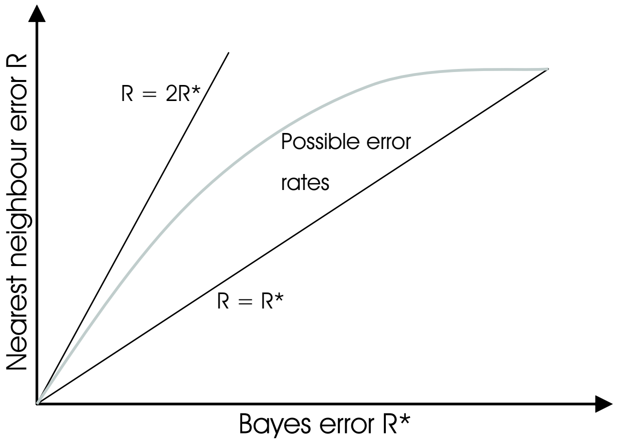


Figure 4.5: Bounds on nearest neighbour error rate R. The error is never better than the best possible Bayes error R^* but also never worse than twice that error ($2R^*$). Figure after Duda et al. (2001).

This subset consists then of the k-nearest neighbours of $\mathbf{x}$. Classification is done by majority vote among the k neighbours, ties are broken randomly. Typically, the distance measure is chosen to be the Euclidean distance in input space $d_i = \|\mathbf{x} - \mathbf{x}_i\|$ after variables being standardised to mean 0 and variance 1 in the training sample set.

A special case exists if $k = 1$. This so-called nearest neighbour rule simply assigns the class label associated with $\mathbf{x}_{nn}$ to the test point $\mathbf{x}$, with $\mathbf{x}_{nn}$ being the sample point nearest to $\mathbf{x}$. Cover & Hart (1967) showed that the error rate of the nearest neighbour rule converges asymptotically to the Bayes risk R^* (see Section 4.2) and is never more than twice R^* (Figure 4.5). In other words, with an infinite data set and an arbitrarily complex classification rule, at least half of the classification information resides in the nearest neighbour. However, the data set at hand is generally finite. In this case, research only showed that asymptotic convergence may be slow and that the error rate may not even decrease monotonically with increasing n (Duda et al., 2001). However, despite the lack of positive statements on convergence behaviour in the finite sample case, the (k)NN method often yields good results.

Unfortunately, there is no general decision rule whether to use the nearest neighbour rule or the k-nearest neighbour rule (and to decide on the size of k) for a given classification task. In theory, k should be large to obtain a reliable estimate $\hat{P}_n(c_i|\mathbf{x})$. At the same time, all of the k nearest neighbours should be very close to $\mathbf{x}$. So the choice of k is a compromise in order to make k large *and* a small fraction of all n samples at the same time (see conditions 4.10 and 4.11). In practise, a given k may yield a good training performance but bad generalisation. Therefore, only classification validation by means of test data leads to a best value of k for a given problem.

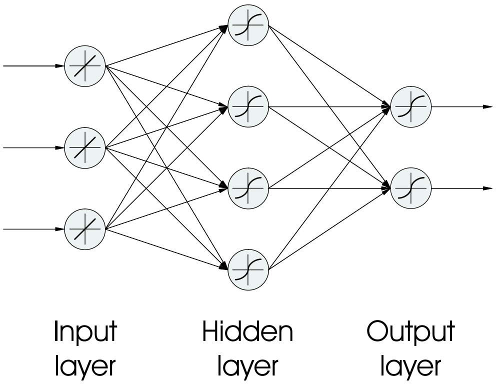


Figure 4.6: Example of a neural network with a fully connected 3-layer 3-4-2 topology: the hidden layer with 4 neurons is in between the input layer with 3 neurons (e.g. 3 log curves) and the output layer with 2 neurons (e.g. 2 lithological units to be classified). Neurons of hidden and output layers have sigmoidal activation functions, the input neurons have linear functions.

4.3.6 Artificial Neural Networks

Originally, the development of artificial neural networks (ANNs) was motivated by imitating the human brain and modelling networks of real neurons. The basic elements of the brain are (from a very simplistic point of view) a large number ($\approx 10^{11}$) of neurons, each consisting of a cell body and connecting strands with synapses at their ends. Each neuron is connected with a few thousands of other neurons by such synapses.

Historically, what is nowadays known as an artificial neural network in the field of pattern recognition had been developed as a *perceptron* by Rosenblatt (1959). It is understood as a computational model consisting of parallel processing units (called neurons) and their modifiable connections (called weights that act as factors). The neurons are generally arranged in layers, with neurons in one layer not being connected with each other. The first layer is always the input layer consisting of p neurons, p being the size of an input vector $\mathbf{x}$ (e.g. number of log curves). The last layer is always the output layer consisting of k neurons, k being the number of possible classes. In between there may be one or several so-called *hidden layers* with any given number of neurons. In a fully connected network, each neuron of layer 1 is connected with each neuron of layer 2, and so on (see Figure 4.6). The structure or *topology* of the neural net is completely described by the number of hidden layers and their number of neurons. Caudhill (1991) reports that one hidden layer is adequate for most classification problems. Kolmogorov's theorem proves that any function can be implemented by a 3-layer network given a sufficient number of hidden neurons (Duda et al., 2001). As for the number of hidden neurons, Lippmann (1987) suggests three times more hidden than input neurons. Bhatt & Helle (2002b) recommend as few as possible and warn that too many hidden neurons will lead to memorisation of the network, resulting in poor generalisation performance. Duda et al. (2001) give $n_w = n/10$ as a rule of thumb, n_w being the number of all weights and n the number of training samples.

Each neuron computes a weighted sum of its inputs and outputs a single value as a nonlinear function of this sum. When the transfer of outputs to the next layer is in one direction only, the networks are termed *feed-forward* nets. Formally, outputs are

$$z_j = f\left(\sum_{i=1}^{d} x_i w_{ji} + b_j\right) = f\left(\mathbf{w}_j^{\mathrm{T}}\mathbf{x}\right) \tag{4.14}$$

where $f(\cdot)$ is a nonlinear activation function, j indexes the units of a hidden or output layer and i indexes the d units in the previous layer. w_{ji} denote the weights between these two layers. A bias b_j is appended to the weight vector $\mathbf{w}$ for mathematical reasons. The activation functions are typically the same for all neurons in a network (though this is not required) and have to be continuous and differentiable. For this work, the sigmoidal function $f(x) = \tanh x$ was selected since it is widely used in the literature. The final outputs z_k from the output layer are used as discriminant functions for the classification task.

Once the network topology has been defined, the network needs to be trained. That is, the output signals z_k generated by the net from input training vector $\mathbf{x}$ are compared with a target vector $\mathbf{t}$ (e.g. $\mathbf{t} = \{0; 1; 0\}$ when classifying into class 2 out of 3 classes). The difference (called *training error*) is used to adjust the weights throughout the neural network. There are several learning algorithms, the most known and widely used one being the backpropagation algorithm.

ANNs have been the focus of considerable research after Rumelhart & McClelland (1986) revived ideas of Werbos (1974) and LeCun (1985) and made the backpropagation algorithm widely known. Numerous papers and books have been published since. There are dozens of papers dealing with neural networks and log interpretation alone (see Chapter 1 for an overview). Many refinements and modifications have been proposed to overcome the shortcomings of neural networks. As they do not change the inherent properties of ANNs, this study will only consider the basic yet most popular case of a 3-layer backpropagation neural network.

Backpropagation Neural Networks

Starting with an untrained ANN and random initial weights, a training pattern is fed to the net, the signal is passed through the hidden layer(s) and the output is determined. The difference between output and target vector is the training error

$$E(\mathbf{w}) = \frac{1}{2}||\mathbf{t} - \mathbf{z}||^2$$

where $\mathbf{w}$ describes all weights in the net, $\mathbf{t}$ is the target and $\mathbf{z}$ the output vector. $E(\mathbf{w})$ is minimum when the output matches best with the training input. The weights are adjusted by means of gradient descent:

$$\Delta\mathbf{w} = -\eta\frac{\partial E}{\partial\mathbf{w}}$$

where η is the learning rate, defining the size of changes in $\mathbf{w}$. For the above discussed case of a 3-layer network, both input-to-hidden and hidden-to-output weights have to be recalculated. The learning rule of the former contains information about the latter. The error $E(\mathbf{w})$ computed at the output layer is propagated back to the hidden layer in order to adjust the hidden-to-input and input-to-hidden weights, hence the algorithm's name.

For mathematical reasons, the initial weights of a backpropagation artificial neural network (BPANN) cannot be zero. Thus, the training process is generally started with random initial weights. Each presentation of a training pattern $\mathbf{x}_i$ to the network is termed *epoch*. To minimise the error, gradient descent is applied. With each epoch the training error decreases monotonically. The backpropagation process requires a stopping criterion, i.e. a threshold error value that, when reached, terminates the iteration. Because the global minimum is not necessarily a desirable goal (as it would most likely lead to an overfit solution), the stopping criterion resembles a kind of regularisation (Hastie et al., 2001). Unfortunately, the network with the lowest training error does not guarantee the best performance on other data sets (poor generalisation). The optimum weights $\mathbf{w}$ of a BPANN can therefore only be evaluated by means of a test data set. As every network starts with different random weights, several ANNs have to be started, trained and tested. The best performing network is then selected by trial and error. Despite being time-consuming, even this process does not strictly guarantee the best possible result (Bhatt & Helle, 2002a).

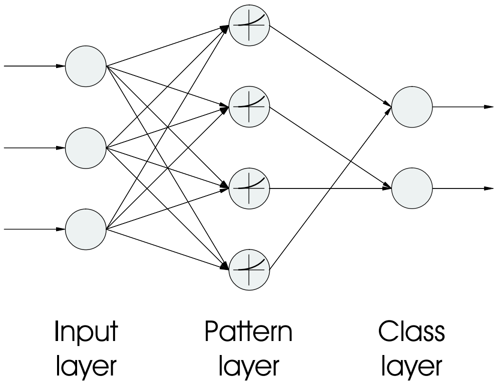


Figure 4.7: Example of a probabilistic neural network with 3-4-2 topology for the same classification task as the ANN in Figure 4.6: the pattern layer with 4 neurons is in between the input layer with 3 neurons (e.g. 3 log curves) and the class layer with 2 neurons (e.g. 2 lithological units to be classified). Neurons of the pattern layer have exponential activation functions. This network is designed for 4 training samples.

Probabilistic Neural Networks

As already mentioned in Section 4.3.4, Parzen window density estimation may be implemented by means of a probabilistic neural network (PNN). Much like a 3-layer ANN presented above, it is composed of neurons arranged in layers and connecting weights. An input layer with p neurons (p being the number of variables per input vector) is fully connected with a pattern layer with n neurons (n being the number of training samples). Each neuron of the pattern layer is connected to exactly one of k neurons of the class layer, k being the number of possible classes (see Figure 4.7). Similar to Equation 4.14, each neuron of the pattern layer outputs a function of the inner product of all input weights and the normalised input vector

$$z_j = f\left(\mathbf{w}_j^{\mathrm{T}}\mathbf{x}\right).$$

Using the exponential function

$$f(x) = e^{(x-1)/\sigma^2}$$

ensures the proper implementation of the Gaussian window function (Equation 4.12). σ is the width of the effective (Gaussian) Parzen window. Training is performed by normalising all $\mathbf{x}_i \ \forall \ i = 1, ..., n$ to have unit length. Then, for the first training sample, the input-to-pattern weights of the first pattern layer neuron are set such that

$\mathbf{w}_i = \mathbf{x}_i$, $i = 1$. The ith (here: the first) neuron of the pattern layer is then connected to that class layer neuron corresponding to the known class. This process is repeated for $i = 2, ..., n$ until all pattern layer neurons are connected to one class layer neuron. The trained network can then be applied to the test data in the usual way.

PNNs are very fast learning networks due to their simple learning rule $\mathbf{w}_i = \mathbf{x}_i$. The drawback is that with many training samples n, the topology and hence the size of vector $\mathbf{w}$ may be very large, requiring a lot of computer memory.

4.3.7 Support Vector Machine

Classifying a data set into one of finite classes based on prior observation (training data) can be viewed as estimating an unknown functional dependency (between input data and class labels). Vapnik, Chervonenkis and others developed what was named *statistical learning theory* in order to

- describe the best approximation to this dependency,
- formulate the general principles for finding the best estimation and
- develop algorithms implementing these principles.

Starting in the 1960s, different principles and concepts were developed (Vapnik & Chervonenkis, 1968; Vapnik, 1979) until the analysis of empirical risk minimisation inductive inference was completed (Vapnik & Chervonenkis, 1989). This allowed then the implementation of a classification algorithm called *support vector machine* (SVM; Vapnik, 1998). The following is a very brief summary of the several principles SVMs are based on. For details, the reader is referred to the previously mentioned literature as well as to the work by Vapnik (2000) and Kecman (2001).

In Equation 4.2 the expected loss (or *risk*) was expressed as a dependency of a given action but it can also be written as a function of the (true) underlying probability density:

$$R_{exp} \sim P(\mathbf{x}, c).$$

With only a finite set of training data available, the average over the probability density function is replaced with the average over these training samples, yielding the *empirical risk*. The induction principle of empirical risk minimisation (ERM; Vapnik & Chervonenkis, 1989) puts this risk as

$$R_{emp} \sim f(\mathbf{x}, \mathbf{w})$$

where $f(\mathbf{x}, \mathbf{w})$ is a parameterised function depending on the input data vector $\mathbf{x}$ and a weight vector $\mathbf{w}$. Regarding the algorithms presented in this thesis, $\mathbf{w}$ would be the means and covariances of LDA (Equation 4.5), the coefficients of logistic regression (Equation 4.8) or the weights of a neural network (Equation 4.14). Linking empirical and expected risk, the law of large numbers ensures that the empirical risk converges to the expected risk as the number of training data points n increases towards infinity:

$$\lim_{n \to \infty} \|R_{exp} - R_{emp}\| = 0.$$

Unfortunately, this does not guarantee that $f(\mathbf{x}, \mathbf{w})$ (which minimises R_{emp}) minimises R_{exp} as well. Vapnik & Chervonenkis (1989) introduced their *learning theorem for bounded loss functions* proving a uniform convergence of R_{emp} to R_{exp}. This ensures that a weight vector $\mathbf{w}_{emp}$ (obtained by minimising R_{emp} using training data samples only) will also minimise the true risk R_{exp} as the training data size increases $(n \to \infty)$. This is why SVM is the only algorithm that actually focusses on the generalisation performance and not on training performance (expressed by R_{emp}).

Finding the minimum empirical risk is an ill-posed problem (Kecman, 2001) for there is an infinite number of possible solutions to the ERM problem. Within the statistical learning theory, the solution to this problem is the restriction of the hypothesis space H of approximating functions (that map $\mathbf{x}$ to classes c) by means of a nested structure

$$H_1 \subset H_2 \subset \cdots \subset H_m \subset \cdots \subset H$$

where m is the model complexity (see Section 4.4.2). This restriction of model complexity is the basis of the structural risk minimisation (SRM) inductive principle (Vapnik, 1998). It ensures that R_{emp} is minimised in H_m as opposed to R_{exp} being minimised in H. In other words, there is a unique solution to the ERM problem if H is restricted in terms of model complexity. This will also be discussed in section 4.4.2.

With these prerequisites, Vapnik (2000) showed that the following risk bound holds:

$$R_{exp}(\mathbf{w}) \leq R_{emp}(\mathbf{w}) + \Omega(n, h, \mu) \tag{4.15}$$

with probability $1-\mu$. In other words, the generalisation performance given a weight vector $\mathbf{w}$ cannot be worse than the bound $\Omega(\cdot)$ called Vapnik-Chervonenkis (*VC*) *confidence*. h is a measure of model complexity and is called *VC dimension*. It is $m+1$ for linear in parameter models. Support vector machines represent such models, meaning that they are linear with respect to the model parameters (α_i, as will be shown in Equation 4.21). $1 - \mu$ is the level of confidence. The bound is valid for any learning machine, expressed as a function of size of training set n and the VC dimension h of that learning machine.

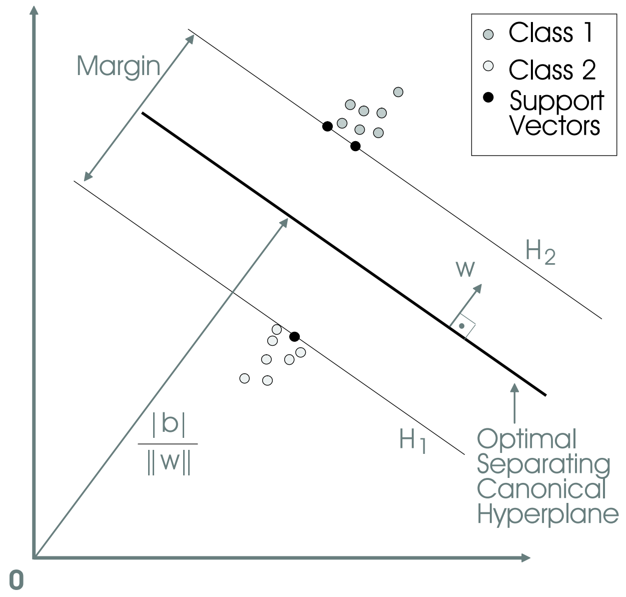


Figure 4.8: Principle of classification by a support vector machine (linearly separable case for two classes in $\mathbb{R}^2$). Two classes 1 and 2 are separated by that hyperplane that maximises the margin, i.e. the distance to the nearest data points. The latter are the only data points relevant for the classification task and are called support vectors (lying on margin hyperplanes H_1 and H_2).

The novelty of the statistical learning theory lies in the fact that it focusses on the minimisation of the generalisation error rather than the training error. All other classification techniques discussed so far are set up with a given structure (e.g. neural net topology, see Section 4.3.6), and minimise the training error (i.e. the empirical risk). The statistical learning theory keeps the training error fixed (at some acceptable level) and minimises the confidence interval and the generalisation error. It does so by creating a model with minimised VC dimension which in turn ensures an optimal generalisation performance.

The support vector machine is the implementation of this theory. It is a linear maximum margin classifier and has been developed for the 2-class case. As shown in Figure 4.8 for $\mathbb{R}^2$, the linearly separable data are divided by a line (in $\mathbb{R}^d$: a hyperplane). The margin is the distance perpendicular to the hyperplane between that plane and the nearest data point. Of all possible hyperplanes, the best is the one that maximises this margin.

During the learning process, the SVM finds the weight vector $\mathbf{w}$ and a bias value b of a discriminant (or decision) function

$$f(\mathbf{x}, \mathbf{w}, b) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b.$$

All points on the separating hyperplane in Figure 4.8 follow

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0.$$

Thus, the decision rules for classification are:

$$\begin{aligned} &\text{Class 1 if} \quad \mathbf{w}^{\mathrm{T}}\mathbf{x} + b > 0 \\ &\text{Class 2 if} \quad \mathbf{w}^{\mathrm{T}}\mathbf{x} + b < 0. \end{aligned}$$

The hyperplane is defined as canonical, i.e.

$$\min \|\mathbf{w}^{\mathrm{T}}\mathbf{x} + b\| = 1.$$

Hence, the points on the margin hyperplanes H_1 and H_2 satisfy

$$\begin{aligned} \mathbf{w}^{\mathrm{T}}\mathbf{x} + b &= +1 \quad \text{for } \mathrm{H}_1 & (4.16) \\ \mathbf{w}^{\mathrm{T}}\mathbf{x} + b &= -1 \quad \text{for } \mathrm{H}_2. & (4.17) \end{aligned}$$

These points are the only ones actually needed for finding the optimal separating hyperplane and are called *support vectors*. The decision rules are now the constraints

$$\begin{aligned} &\text{Class 1 if} \quad \mathbf{w}^{\mathrm{T}}\mathbf{x} + b \geq +1 & (4.18) \\ &\text{Class 2 if} \quad \mathbf{w}^{\mathrm{T}}\mathbf{x} + b \leq -1. & (4.19) \end{aligned}$$

Using Equations 4.16 and 4.17 it can be shown that the margin M is

$$M = \frac{2}{\|\mathbf{w}\|}.$$

The goal is to maximise the margin, i.e. minimise $\|\mathbf{w}\|$ (or actually $\|\mathbf{w}\|^2$) subject to constraints 4.18 and 4.19. This is a non-linear optimisation problem with inequality constraints and can be solved by maximising the Lagrange function in dual formulation (Vapnik, 2000)

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j c_i c_j \mathbf{x}_i \mathbf{x}_j \qquad (4.20)$$

subject to constraints

$$\alpha_i \geq 0$$

$$\sum_{i=1}^{n} \alpha_i c_i = 0$$

where α are the Lagrange multipliers found during the search for an optimal saddle point of the function and c_i are the class labels ($+1$ or -1). L_D only depends on the Lagrange multipliers α. All points with $\alpha > 0$ are support vectors. All other terms with $\alpha = 0$ disappear (meaning all points other than support vectors do not influence the classification). The solution to 4.20 for the linearly separable case is

$$\mathbf{w} = \sum_{i=1}^{n_s} \alpha_i c_i \mathbf{x}_i \tag{4.21}$$

subject to constraints

$$\begin{aligned} \alpha_i &\geq 0 \\ \sum_{i=1}^{n} \alpha_i c_i &= 0 \end{aligned} \tag{4.22}$$

where n_s is the number of support vectors.

For the non-separable case where data classes are overlapping (e.g. due to noisy data), the solution 4.21 is the same, but the constraint 4.22 is different:

$$C \geq \alpha_i \geq 0$$

where the upper bound C is a penalty parameter chosen by the user. A high C assigns a high penalty to classification errors, and *vice versa*.

The SVM algorithm presented so far is a linear classifier. In order to discriminate non-linear classification problems, input vectors $\mathbf{x}$ of input space $\mathbb{R}^d$ are mapped into vectors $\mathbf{z}$ of a higher dimensional feature space $\mathbb{H}$:

$$\Phi : \mathbb{R}^d \rightarrow \mathbb{H}.$$

The mapping function Φ is implemented by means of a kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{z}_i^{\mathrm{T}} \mathbf{z}_j = \mathbf{\Phi}_{\mathbf{x}_i}^{\mathrm{T}} \mathbf{\Phi}_{\mathbf{x}_j}.$$

The Lagrangian (Equation 4.20) is rewritten as

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j c_i c_j K(\mathbf{x}_i, \mathbf{x}_j).$$

Thus, the non-linear problem is mapped into a higher dimensional space where it can be solved linearly. It is an elegant property of SVMs that although the feature space $\mathbb{H}$ may be extremely high dimensional, the kernel function can be directly computed in the (lower dimensional) input space. Examples for kernel functions are:

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= ((\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j) + 1)^p && \text{Polynomial of degree } p \\ K(\mathbf{x}_i, \mathbf{x}_j) &= e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2} && \text{Gaussian RBF} \\ K(\mathbf{x}_i, \mathbf{x}_j) &= \tanh((\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j) + b) && \text{Multilayer perceptron.} \end{aligned} \tag{4.23}$$

In summary, the only parameters a user can choose are the upper bound C to penalise classification errors and the kernel function K.

One problem persists: the SVM algorithm is a binary classifier. There are recent efforts to expand the theory foundations to multi-class SVM. Other methods apply binary classifiers to multi-class problems such as one-versus-all, all-pairs, error-correcting-output-code, one-versus-one or directed acyclic graph approaches (Dietterich & Bakiri, 1995; Allwein et al., 2000; Hsu & Lin, 2001). Several options were tested on the PROMESS-1 data set and the one-versus-all method was chosen for its simplicity and fast performance. In the case of k classes, class c_1 is classified versus all other classes c_2 to c_k. Then class c_2 is classified versus c_3 to c_k and so on. Finally, after $k-1$ binary classifications, all class labels are determined.

4.4 Other Considerations

4.4.1 Bagging

Having a complete data set such as the ones produced by the PROMESS-1 project allows the data to be divided into test and training data sets of varying sizes. This for the evaluation of classifiers favourable situation is ideal to enhance a classifier's performance by means of bagging (derived from bootstrap aggregation). It uses multiple subsets of a given training set that are created by selecting n_s samples from the training data set $\mathcal{S}$ containing n samples, $n_s < n$. Each subset is used to train a so-called *component* classifier. The combined classification is achieved by majority vote of all component classifiers (Figure 4.9).

4.4.2 Bias and Variance

As already mentioned in the introduction (see also Figure 1.2), training a given classifier on a training data set as good as possible is not the ultimate goal in pattern

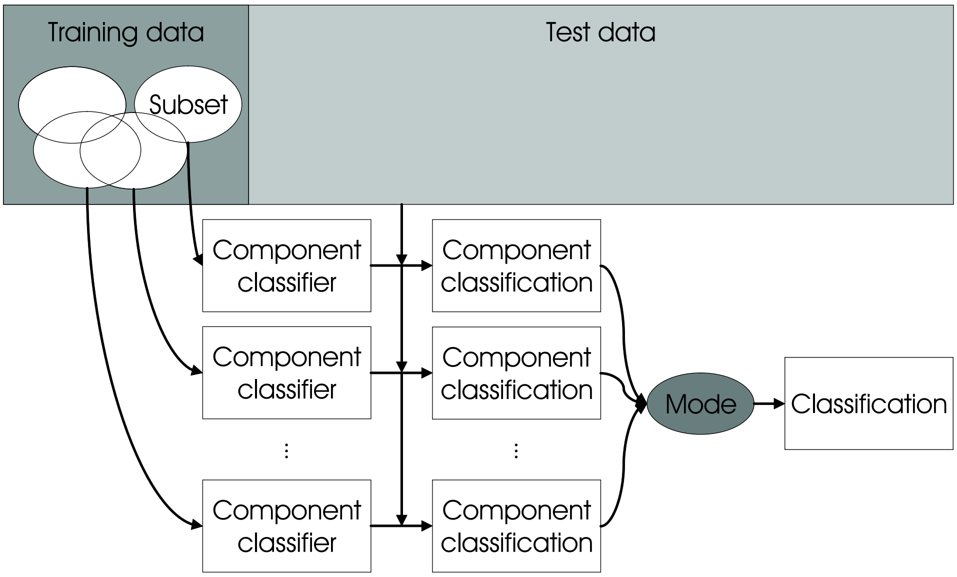


Figure 4.9: Schematic principle of bagging. Component classifiers are trained with subsets of the training data. Subsets may or may not include same data points. The final classification is achieved by majority vote (the *mode*) of all component classifications.

recognition. It is the performance of a trained classifier on new data (generalisation) that is more important. Unfortunately, a low training error does not guarantee a low generalisation error, i.e. the classification error of the test data set. In fact, both training and generalisation error are a function of model complexity m (see Figure 4.10).

Model complexity is always a tradeoff between bias and variance. Bias is defined as the accuracy of the classification match, i.e. a measure of how well the classification fits the data. Variance is the precision of this match, i.e. the refinement and/or complexity of the trained model. High bias means a poor match, likewise high variance means a weak match (because of bad generalisation). Hence, low bias and low variance are desired, but one can only be reduced at the cost of the other. A high model complexity will lead to very detailed decision boundaries (high variance) that will have small differences between true and expected values (low bias), and *vice versa*. In the domain of classification it is exactly this bias-variance tradeoff that determines the generalisation performance of a classifier: high model complexity will reduce the error on the training set to minimum, but the thus trained classifier will almost certainly perform poor on new (test) data. Regarding the classification algorithms discussed so far, the following list shows the parameters that determine their model complexity (Duda et al., 2001):

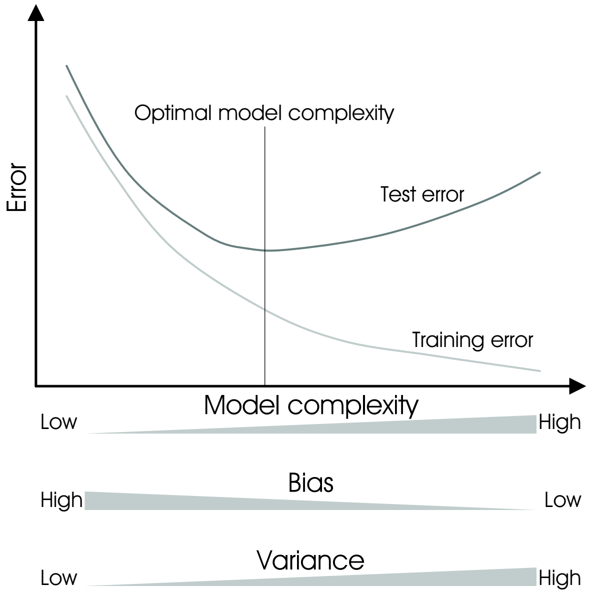


Figure 4.10: Bias-variance tradeoff and test and training error as functions of model complexity.

- LDA: model complexity cannot be adjusted;
- QDA: model complexity cannot be adjusted;
- SVM: self-adjusting by means of SRM;
- kNN: number of neighbours;
- LOGREG: model complexity cannot be adjusted;
- ANN: number of hidden neurons;
- PNN: Parzen window width.

For every classification task, the most adequate model complexity has to be determined to achieve the best generalisation performance (low test error) as shown in Figure 4.10. This was done empirically in the case of PROMESS-1 boreholes by choosing the most appropriate parameter(s) for each algorithm as shown in the next chapter.

4.4.3 Computational Performance and Software Used

All of the above described computations were performed on a standard personal desktop computer with 3.6 GB memory space and a 2.4 GHz Intel Pentium 4 processor.

It was running on a Gentoo Linux operation system, all calculations were performed by The MathWorks' software package MATLAB (version 7.0.4, release 14, service pack 2). For the different algorithms, the following toolboxes and scripts were used: statistic, neural network and optimisation toolboxes by The MathWorks, The Spider toolbox by Weston et al. (2005), discriminant analysis toolbox by Kiefte (1999) and the logistic regression toolbox by Fort (2005).

The amount of input data was small compared to typical classification tasks found in the literature. For 18 log curves recorded over an interval of 147 m at a sampling interval of 5 cm (as for hole PRGL-1, figures for PRAD-1 are even smaller), the total number of values were 52,920. 5 % of them being training data correspond to 2,650 values, leaving 50,274 data points as test set. Classification in medical or speech recognition applications is often dealing with dozens to hundreds of input dimensions and many hundred-thousands of data values. The classification of log curves is not computationally challenging. In extreme cases, such as e.g. logging a 5,000 m well recording 25 log curves at 10 cm intervals, it would lead to 1.25×10^6 data points, which still do not require special main frame computers. Standard PCs with ample memory are capable of performing these tasks, which then may become time-consuming to compute though.

The parameter predominantly affecting model complexity and thus dominating computing time is the dimension of input space (here: number of log curves). For

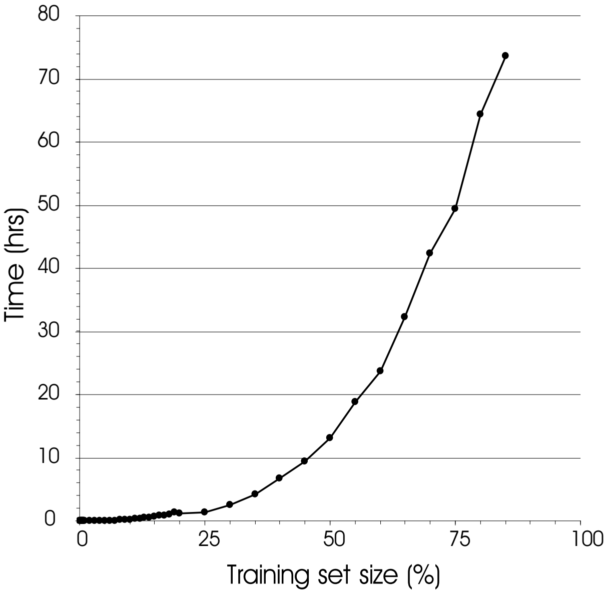


Figure 4.11: Computing times for logistic regression. The training data set size is increased as a percentage of all available data (52,920 values).

a given classification performance, the number of training samples required grows exponentially with that dimension. This fact is known as the curse of dimensionality (see also Section 4.1.3) and its reason is that high dimensional functions are extremely more complicated than low dimensional ones (Bellmann, 1961). Again, 17 and 18 log curves for holes PRAD-1 and PRGL-1, respectively, can both be considered as low dimensional classification tasks.

Typical computing times in the discussed case of PROMESS-1 boreholes were less than one minute for training the classifier with 2,650 values (= 5 % of all data) and testing it on the remainder. However, logistic regression and backpropagation neural networks showed exponentially increasing computing times when the training data set was increased considerably, as shown in Figure 4.11 for logistic regression. Linear discriminant analysis, support vector machines, k-nearest neighbour and probabilistic neural networks showed no such increasing computation times with growing training data sets.

5 Stratigraphical Classification

In this chapter, logging data acquired during the PROMESS-1 project at drill sites PRAD-1 and PRGL-1 are used to computationally classify the logged intervals into stratigraphic sequences. First, the data sets are tested for Gaussian distribution. Then, dimensionality reduction in terms of factor analysis is attempted. For each of the previously discussed algorithms the best parameter(s) are determined. The performances of the methods are compared and the impact of bagging on the data is examined. Also, the effect of training data size on the classification match is discussed. A thorough analysis of performance when including or excluding selected log curves follows.

5.1 Gulf of Lion

5.1.1 Conventional Interpretation

Based on single and multi-channel seismic data and preliminary palaeontological results from PRGL-1 cores, the drilled interval 0–300 mbsf was divided into 5 stratigraphical sequences (Figure 2.4). Also, XRF log data provided information in sequence boundaries. It is now widely agreed that the main present day sediment bodies were deposited during glacials (at sea-level low stand when the areas were closer to the shoreline than today). They are separated by major erosional surfaces that mainly originate from continental erosion (Berné et al., 2004). The classification task in the case of PRGL-1 was to divide the logged interval into these 5 sequences and thus determining the erosional surfaces.

Visual core descriptions report a majority (>98 %) of silty clay throughout the entire interval with minor occurrences of clay, silt and very fine sand (Dennielou, pers. comm.). Other interpreted core data (e.g. mineralogical or petrological logs) are not available at the time of writing. For seismic velocity data were not as accurate as necessary, sequence boundaries were established by using the Ca/Fe ratio recorded from XRF-scanned cores (Figure 5.1a). High Ca count rates are commonly associated with pelagic regimes, whereas high Fe content stands for detrital regimes (Rothwell et al., 2005). These curves can therefore be interpreted as an indicator of distance to the shore line (and thus, sea-level). Sequence boundaries were then set to 71, 121,

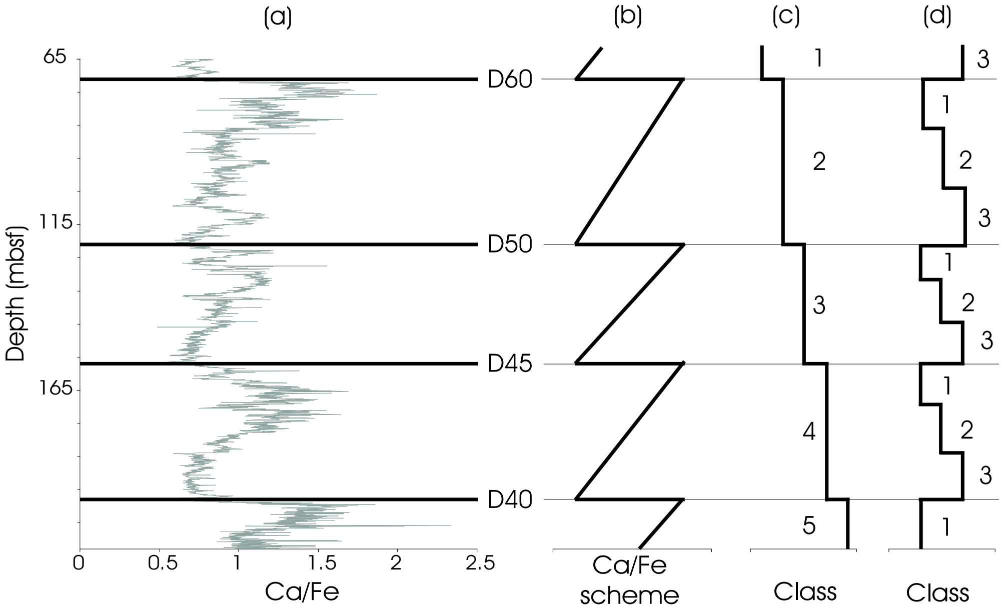


Figure 5.1: Ca/Fe plot and sequence boundaries D60 to D40 (PRGL-1). (a) Ca/Fe data. (b) saw-tooth pattern of Ca/Fe data. (c) class labels scheme as applied to PROMESS-1 data. (d) Alternative class labels scheme following sea-level high-stand/mid-stand/low-stand groups.

157 and 198 mbsf.

Conventional log analysis by visual human interpretation of the logs (Appendix A) proofs to be extremely difficult if not impossible. Discontinuity D40 (Figure 2.4) can be detected on most logs with ease but other boundaries are hardly detectable (with the exception of using Ca and Fe, of course, and the Potassium curve). Because the lithological changes within this interval of silty clay are so subtle, most geophysical and geochemical parameters do not vary much. Only looking at the curves, an interpreter could likely miss sequence boundaries which, however, are aimed to be detected by automated classification. Clearly, the stratigraphical units are detectable only by means of a combination of logging curves (in other words, in $\mathbb{R}^d$ space, d being the number of curves) and not by examining a few selected log curves separately. The combination of all available log data in such (higher dimensional) input space is the major advantage of automated pattern recognition over conventional log analysis.

Because the log curves show no major changes at the thus established sequence boundaries, the classification task was set-up such as to characterise the geophysical properties of each unit rather than detecting those boundaries. By finding a distinct variable set for each unit (class), their discrimination leads inevitable to the determination of boundaries as well. Moreover, this approach is superior when it comes to the detection of stratigraphical units that occur multiple times within an interval (such as thin layered beds for instance). PROMESS-1 data are simple in terms of their class labels, which are an ordered succession of numbers (see Figure 5.1c). Here, no unit block occurs more than once, but for the general case this option is certainly desirable to have.

A remark should be made regarding the conventional interpretation of stratigraphical log data on one side and the classification into units on the other. As shown in Figure 5.1b, most geochemical data (like the Ca/Fe ratio) exhibit some sort of saw-tooth pattern that reflects sea-level changes and distance to the shoreline at the respective depth. This pattern repeats itself within each unit block. However, the approach of this work is not to classify each pattern sequence into several subgroups (e.g. sea-level high-stand, mid-stand, low-stand) but to look for an underlying pattern to be classified that distinguishes each stratigraphical unit from the others as such. This difference is depicted in Figure 5.1c and d. The training class labels used for PROMESS-1 data follow that of Figure 5.1c. The goal is to detect and distinguish between those underlying parameters that are different for each glacial period and not those that change within each period in the same way (such as shown in Figure 5.1d).

5.1.2 Data Used

For the classification of logging data, the following 18 log curves were selected:

- Density (Rho) and sonic velocity (V_P) from MSCL measurements;
- Calcium (Ca), Cobalt (Co), Chromium (Cr), Copper (Cu), Iron (Fe), Potassium (K), Manganese (Mn), Nickel (Ni), Strontium (Sr), Titanium (Ti) and Vanadium (V) from XRF measurements;
- Thorium (Th), Uranium (U), mean electrical micro-resistivity (RD), Hydrogen (H) and magnetic susceptibility (SU) from downhole logging measurements.

Zinc and Lead XRF measurements as well as Carbon, Oxygen and Silicon downhole logging measurements were not used for they were regarded as being too noisy. Data pre-processing was applied as described in Section 4.1. For borehole PRGL-1, the logged interval was from 65.7 mbsf to 213.0 mbsf at 5 cm vertical resolution, thus comprising 2946 measure points (equaling $2946 \times 18 = 53028$ data points). Although core recovery was excellent throughout the entire borehole (thus providing complete XRF and MSCL measurements between 0 and 300 mbsf), wireline logging was not possible below 213 mbsf due to hole collapse and above 65 mbsf due to hole stability

Factor	Eigenvalue	Variance (%)	Cumulative Variance (%)
1	**6.09**	33.82	33.8
2	**2.51**	13.95	47.8
3	**1.70**	9.43	57.2
4	**1.22**	6.77	64.0
5	**1.09**	6.08	70.0
6	0.95	5.26	75.3
7	0.84	4.66	80.0
8	0.82	4.53	84.5
9	0.71	3.93	88.4
10	0.57	3.19	91.6
11	0.37	2.05	93.7
12	0.33	1.86	95.5
13	0.27	1.49	97.0
14	0.21	1.16	98.2
15	0.15	0.83	99.0
16	0.08	0.42	99.4
17	0.06	0.31	99.7
18	0.05	0.26	100.0

Table 5.1: Factor analysis results from log curves of hole PRGL-1: Eigenvalues, variance and cumulative variance. 70 % of the data set's variance can be represented by 5 factors.

issues. Fortunately, all 4 sequence boundaries to be detected lie within the logged interval.

The depth information (z) was not included in the data for two reasons: first, class labels were assigned in ascending order, thus being a linear function of z. This undesired dependency would have overlain other classification rules and made the classification dependent on the depth location of the training samples. Second, in the more general case of spatial mixed and interchanging class labels this approach would only worsen the classification result. For more details, see the discussion (Chapter 6).

5.1.3 Factor Analysis

In order to reduce the input data dimension and remove redundant information within the logging data, a factor analysis as outlined in Section 4.1.3 was attempted. The results are given in Table 5.1. There are 5 factors with eigenvalues >1. They represent only 70 % of the data set's variance, a value that is rather low. Usually, 3 factors are sufficient to explain $\approx 75\,\%$ of the data (Bücker, pers. comm.). Using these 5 factors instead of the 18 log curves in subsequent classification operations would mean discarding 30 % of the information contained in all data. It was there-

Log Curve	χ^2	Kolmogorov-Smirnov
Ca	564	0.069
Co	70	**0.018**
Cr	103	**0.015**
Cu	2×10^8	0.066
Fe	161	0.034
H	773	**0.024**
K	1121	0.074
Mn	1694	0.056
Ni	184	0.029
ln Mean Micro-resistivity	7306	0.041
Sr	71	**0.012**
ln Susceptibility	304	**0.017**
Th	111	**0.011**
Ti	295	0.054
U	96	**0.019**
V	112	0.033
Density	238	0.037
V_p	505	0.046
Critical value	69	0.025

Table 5.2: Tests of distribution on PRGL-1 log curves. Left: χ^2-test (which no curve passes). Right: Kolmogorov-Smirnov test. Bold numbers pass this test (i.e. the null hypothesis that the values come from a standard normal distribution cannot be rejected). Both tests assumed a probability of error $\alpha = 0.05$.

fore decided to perform the classifications with all 18 logging curves rather than with the 5 factor logs.

5.1.4 Data Distribution

For all selected log curves, normal probability plots and histograms were created (see Appendix D). χ^2 (chi-square) and Kolmogorov-Smirnov tests were performed. The χ^2-test indicates that no log curve is normally distributed. However, the Kolmogorov-Smirnov test shows a normal distribution for Co, Cr, H, Sr, ln of susceptibility, Th and U curves (see Table 5.2). This discrepancy can be explained by deviations at the distribution margins that are attenuated by the Kolmogorov-Smirnov test (Trauth, 2003). Having thus established the fact that not all input data curves are Gaussian distributed, the multivariate distribution of all 18 log curves cannot be Gaussian either. The assumption of normally distributed data will hence be violated in the subsequent application of learning algorithms. It has to be stated though, that this has little effect on the real-life use of these methods, an observation previously made by other authors (e.g. Hastie et al., 2001).

Class	Data Points	Training Points	Fraction
1	111	6	5 %
2	1002	50	5 %
3	716	36	5 %
4	826	41	5 %
5	291	15	5 %
Total	2946	148	5 %

Table 5.3: Distribution of training data for each class.

5.1.5 Supervised Learning

In addition to a test data set, supervised learning requires a training data set with input vectors $\mathbf{w}$ and class labels c. Subdividing the total available data into training and test data (also called off-training data) can be accomplished be several means. The training data set can be either a randomly drawn subset of the total data or samples can be drawn at a fixed sample rate (say, one sample every metre). Furthermore, all training data may be either

- a fixed fraction of the total data,
- a fixed fraction of each class (thus dependent on the number of sample points per class),
- a fixed number of samples (of the total data) or
- a fixed number of samples per class.

Unless otherwise stated, the subsequent classifications were performed with a training data set of 5 % of the total data per class. The training data were randomly drawn from the logged interval of PRGL-1 with 2946 data points (=100 %) as shown in Table 5.3 . The test data are the remainder of the total data (95 %). As to what extend this nonetheless arbitrarily designed procedure resembles a real-life situation of having a core recovery of 5 % is debatable. It shall be pointed out though that a training-to-total data ratio of 1/20 can safely be regarded as a tough condition for any classifier (even more so considering a mere 6 training samples for class 1).

Performance was measured by comparing the test data classification results with true class labels established from conventional analysis (see section 5.1.1). The number of correctly assigned class labels is output as a percentage. These data do not include the training data set as the latter is correct by definition.

With both training and test data sets generated, the afore discussed classification algorithms were applied as described in the next sections.

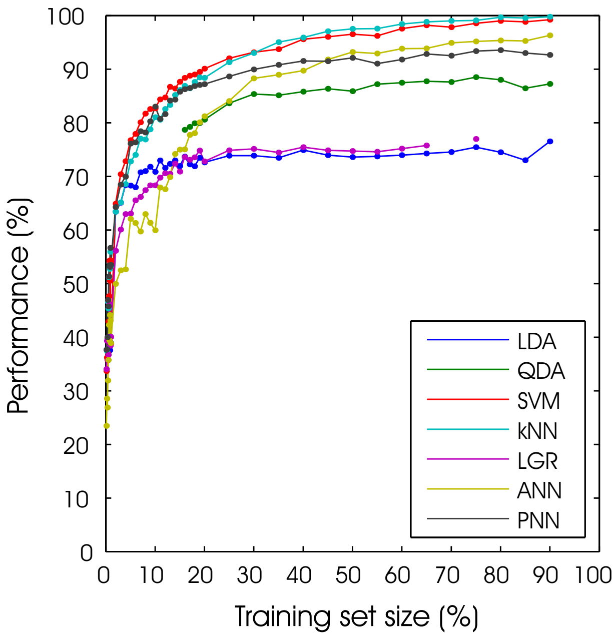


Figure 5.2: Performance with varying training data set size (in percentage of total available data, PRGL-1). The mean performance value from 3 classification runs with different randomly selected training sets is shown. Logistic regression (LGR) was aborted with input data >60 %. Quadratic discriminant analysis (QDA) only works with more than 16 % of the data.

Discriminant Analysis

Linear and quadratic discriminant analyses were performed. The latter algorithm requires a minimum of training data. Each class has to be represented by a minimum of p samples where p is the dimension of input space (here: number of logging curves). This restricts the application of quadratic discriminant analysis to training set sizes >16 % (see Figure 5.2). This limit originates from the sampling limit imposed from the class with the least data samples, which is class 1 with 111 samples (16 % of 111 equals 19 samples, see also Table 5.3).

Thus, for the predominately discussed case with 5 % training data, only LDA will be presented and QDA will be omitted.

k-Nearest Neighbour Estimation

For the best classification using the k-nearest neighbour algorithm, the most appropriate value of k needed to be determined. From Figure 5.3 it is evident that classification results improve when the number of neighbours decreases. Obviously, k was chosen to be 1, effectively implementing the nearest neighbour rule.

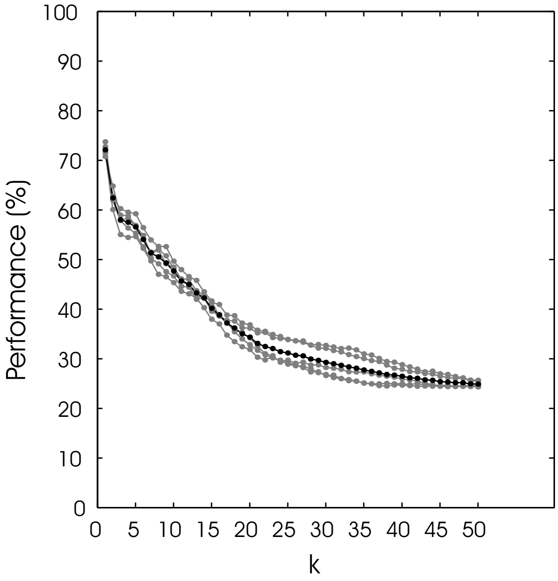


Figure 5.3: Performance of k-nearest neighbour algorithm with changing parameter k (PRGL-1). Of all available data, 5 % were assigned to the training set, 95 % to the test set. 5 data sets were computed (grey curves), the black curve shows their mean. Best performance is at k = 1.

Logistic Regression

Logistic regression was performed using a penalised logistic regression implementation with ridge regression as described by Zhu & Hastie (2004). Figure 5.2 shows only results with less than 80 % training data, because the algorithm turned out to be too time consuming when trained with more data samples (see also Figure 4.11).

Support Vector Machine

For the application of SVM to the PRGL-1 data set, a radial basis function kernel (Equation 4.23) was used. The best parameters for the given data set were established by training the classifiers with 5 % of total data and testing it on the remainder 95 %. The parameters yielding best test performance were then chosen. These are for the upper bound C=7 and for the RBF spread σ=0.4 (see Figure 5.4).

Backpropagation Neural Network

Contrary to other classification techniques, backpropagation neural networks (ANN) give different results when repeatedly trained on the same training data set. This is due to the randomness of initial weights $\mathbf{w}$ as described in Section 4.3.6. Deter-

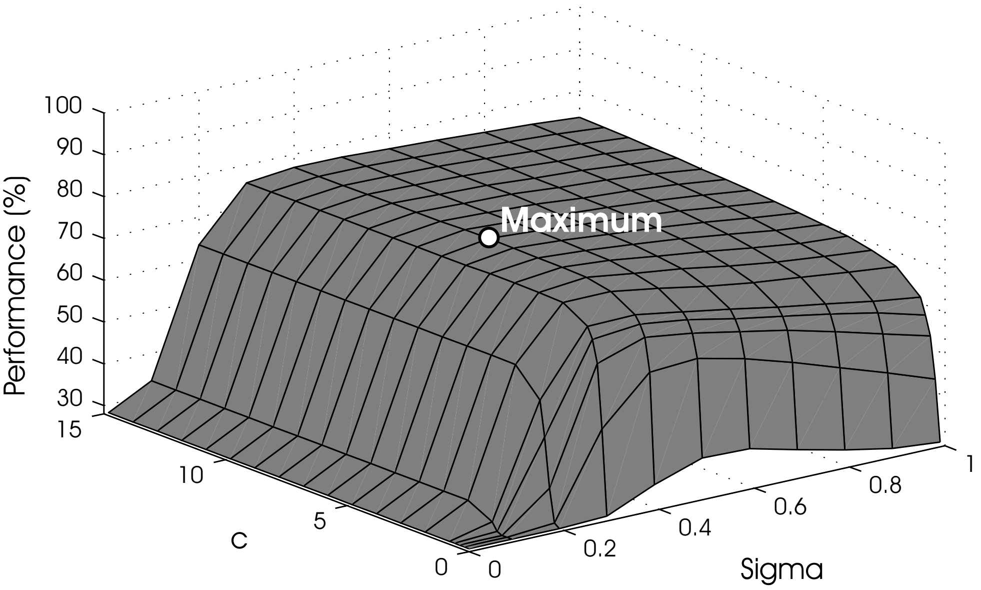


Figure 5.4: Performance of RBF-kernel support vector machines with changing upper bound C and RBF-kernel spread σ (PRGL-1). Of all available data, 5% were assigned to the training set, 95% to the test set. A mean performance value was calculated from 5 data sets. Performance drops considerably for values of C < 2 and σ < 0.3 . Best performance is at C = 7 and σ = 0.4 .

mining the best combination of number of hidden neurons and stop condition of the error minimisation process proved to be rather fruitless as there are apparently no preferred parameters that would significantly improve the classification result (Figure 5.5). Rather, the initial weights condition the subsequent result, a property of neural networks that is undesirable but intrinsical. The parameters arbitrarily chosen were 24 hidden neurons and a stop condition of $e_{\text{stop}} = 5 \cdot 10^{-4}$. With 18 log curves as input data and a classification goal of 5 stratigraphical sequences, the network topology was hence a 18-24-5 neural net.

Probabilistic Neural Network

According to the theory of PNNs (see Section 4.3.6), the network topology of the classification of PRGL-1 logging data was 18-148-5 (18 log curves, 148 training data points, 5 class labels). As this topology is entirely determined by the problem at hand, the only parameter to be determined by the user is the size of the Parzen window σ. Several runs were combined to a mean parameter curve revealing that a value of σ=0.2 would work best (Figure 5.6).

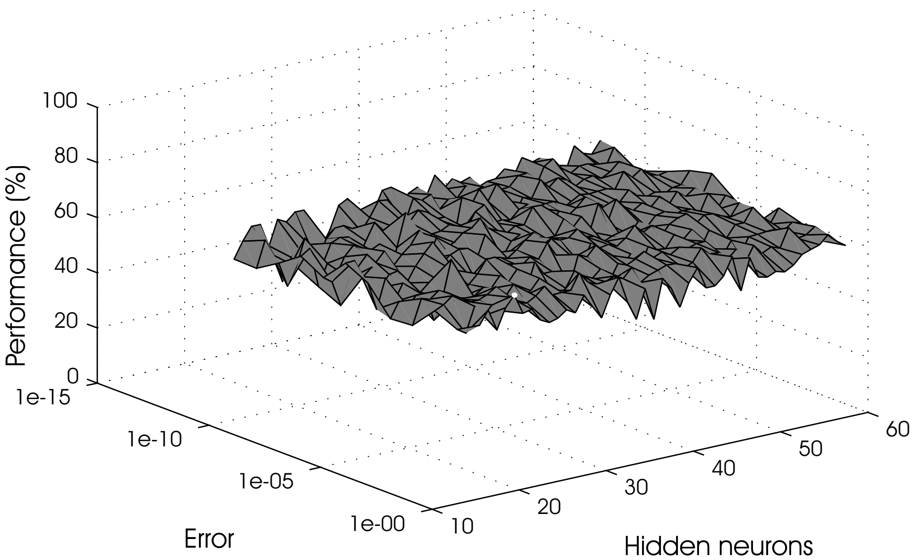


Figure 5.5: Performance of backpropagation neural networks with changing training error bound and number of hidden neurons (PRGL-1). Of all available data, 5 % were assigned to the training set, 95 % to the test set. Performance varies homogeneously over the tested parameter interval, no parameter combination is superior to another.

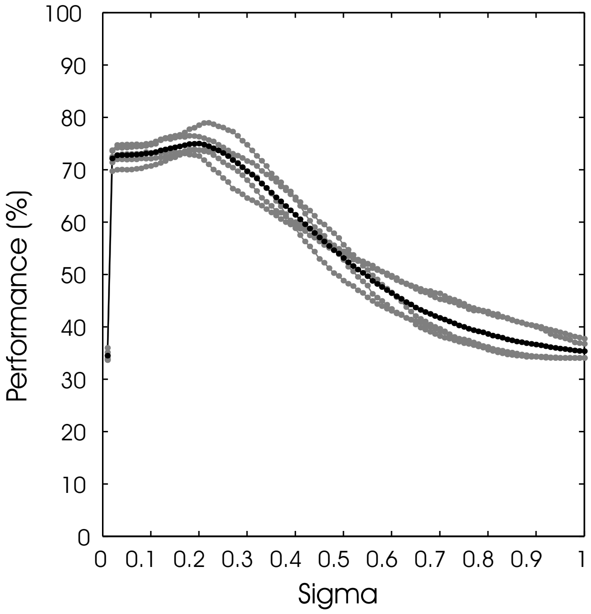


Figure 5.6: Performance of probabilistic neural networks with changing RBF spread σ (PRGL-1). Of all available data, 5 % were assigned to the training set, 95 % to the test set. 5 data sets were computed (grey curves), the black curve shows their mean. Best performance is at $\sigma = 0.2$.

Comparison

As previously described, the total available data of borehole PRGL-1 was divided into a training and a test data set (5 % and 95 %, respectively). With these data sets the 6 above discussed classification techniques were performed. The results are plotted in Figure 5.7. This figure is central to this work as it shows the basic results of all classifications on one page. The five colours represent the 5 classes to be identified. The leftmost column shows that randomly chosen training data points (5 % per class), the one next to it the test data. Then follow the results of classifications. Also given are the match values for each classifier. ANN is worst with 61 %, followed by LGR and LDA (66 % and 71 %, respectively). kNN (72 %), PNN (74 %) and SVM (78 %) perform best. Evidently, the automatically classified stratigraphy suffers from many very thin layers within larger blocks of the (mostly properly classified) sequences. In this special case where the goal is to establish large volume sequences and their boundaries, the occurrence of such thin layers is highly unlikely and may be rejected for geological —not mathematical!— reasons. This consideration justifies a smoothing operation, where each data point is assigned to the class that most often occurs in a 3 m-interval around that point (i.e. the *mode* of this interval). The smoothed result is shown in Figure 5.8. Generally, classification improves by around 12 % with this operation, yielding a very good match of 91 % in case of the SVM classifier.

Naturally, some sequence boundaries are more difficult to detect than others. Sequences 4 (orange) and 5 (grey) are well discriminated by all classifiers. However, a quick look at the logging curves would have established that boundary at 198 mbsf as well. More subtle are the boundaries on top of and below sequence 3 (green) at 157 and 121 mbsf, respectively. Here, the better performing algorithms (SVM, kNN, PNN) show some advantages over the remainder, even more so when looking at the smoothed columns (Figure 5.8). Discriminating sequence 1 (red) and 2 (blue) again is an easy task for SVM, kNN. LDA establishes this boundary (at 71 mbsf) equally well, PNN could do better and LGR completely fails in recognising sequence 1, which may be attributed to the little training data available (see Table 5.3). Another view on this subject offers the *performance cross matrix* plot which was specifically designed to highlight misclassification subject to each class label. Figure 5.9 shows the result for classifications of PRGL-1. It quantifies all misclassification for each class. Most striking are the generally difficult classification of class 1 as well as issues with class 4. Class 5 poses no major difficulties to all algorithms but ANN.

An important aspect to keep in mind is that the performances just presented are not generally valid results. They reflect the performance for only one specific training set used (and in the case of neural networks, the initial random weights). In order to

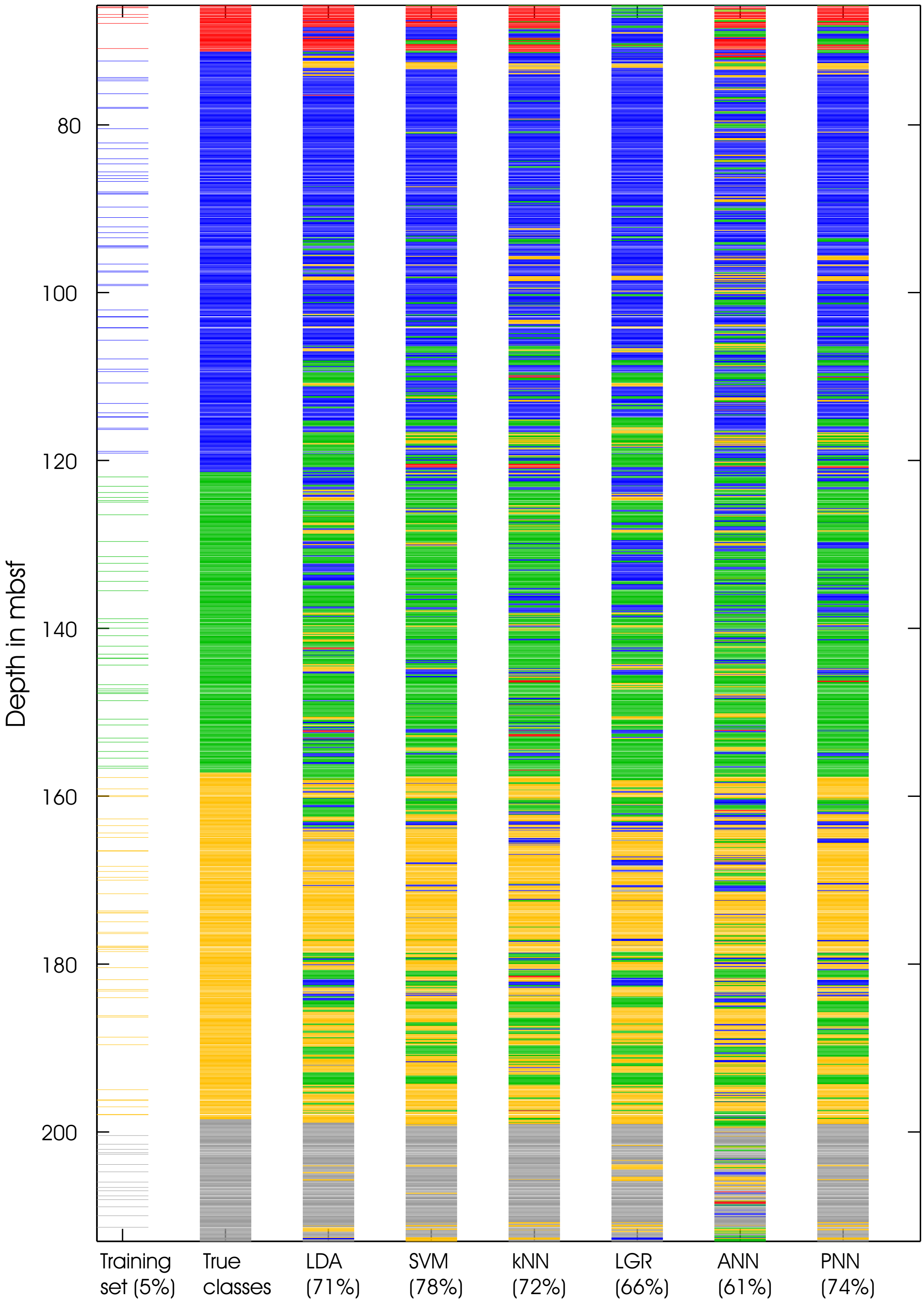


Figure 5.7: Classification of hole PRGL-1. From left to right: Training data set (5% of all available data per class); true class labels (sequence 1 = red, 2 = blue, 3 = green, 4 = orange, 5 = grey); linear discriminant analysis; support vector machine; k-nearest neighbour; logistic regression; artificial neural network; probabilistic neural network.

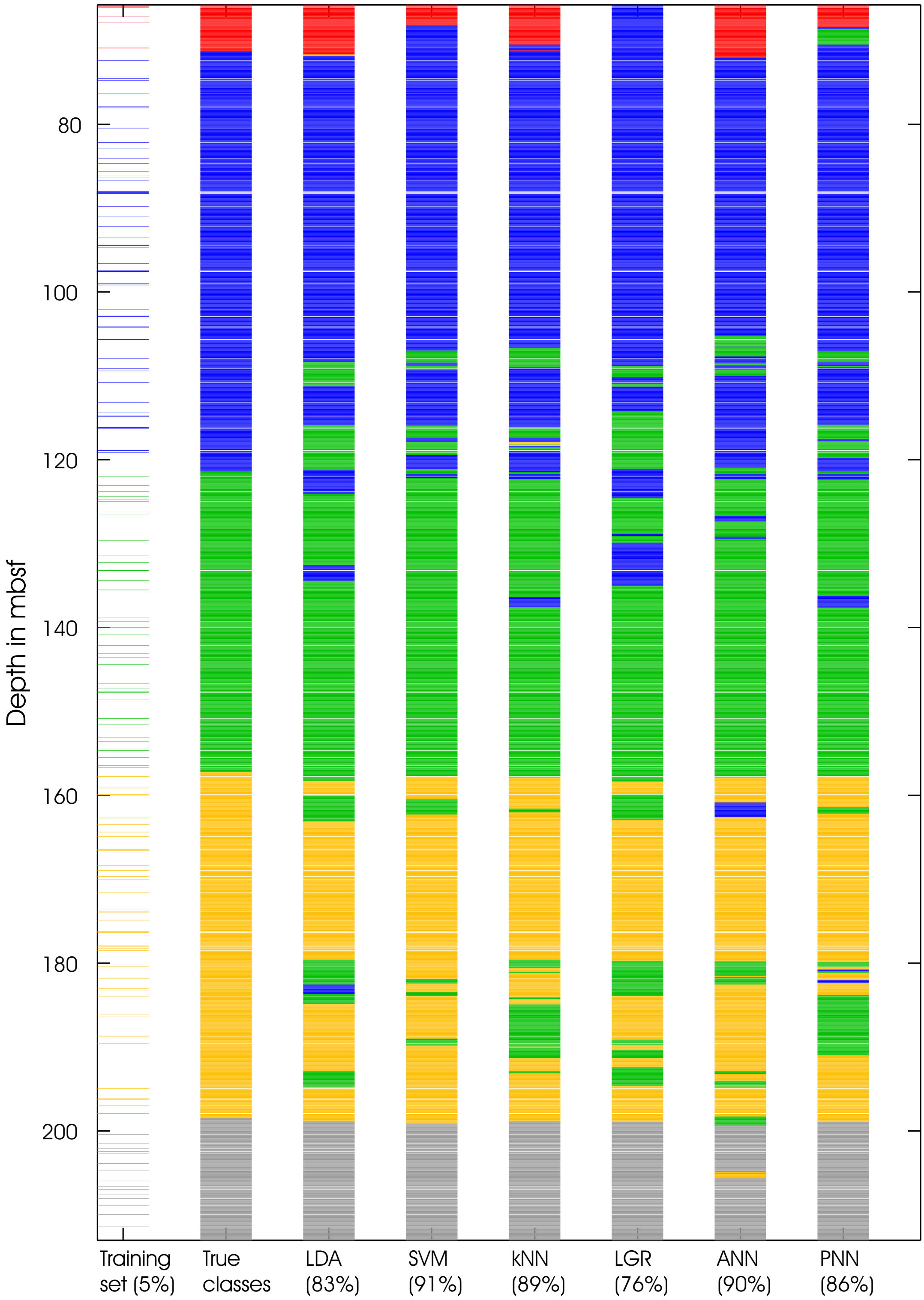


Figure 5.8: Smoothed classification of hole PRGL-1. Each value represents the most frequent class within a window of ±1.5 m around the sample point. From left to right: Training data set (5 % of all available data per class); true class labels (sequence 1 = red, 2 = blue, 3 = green, 4 = orange, 5 = grey); linear discriminant analysis; support vector machine; k-nearest neighbour; logistic regression; artificial neural network; probabilistic neural network.

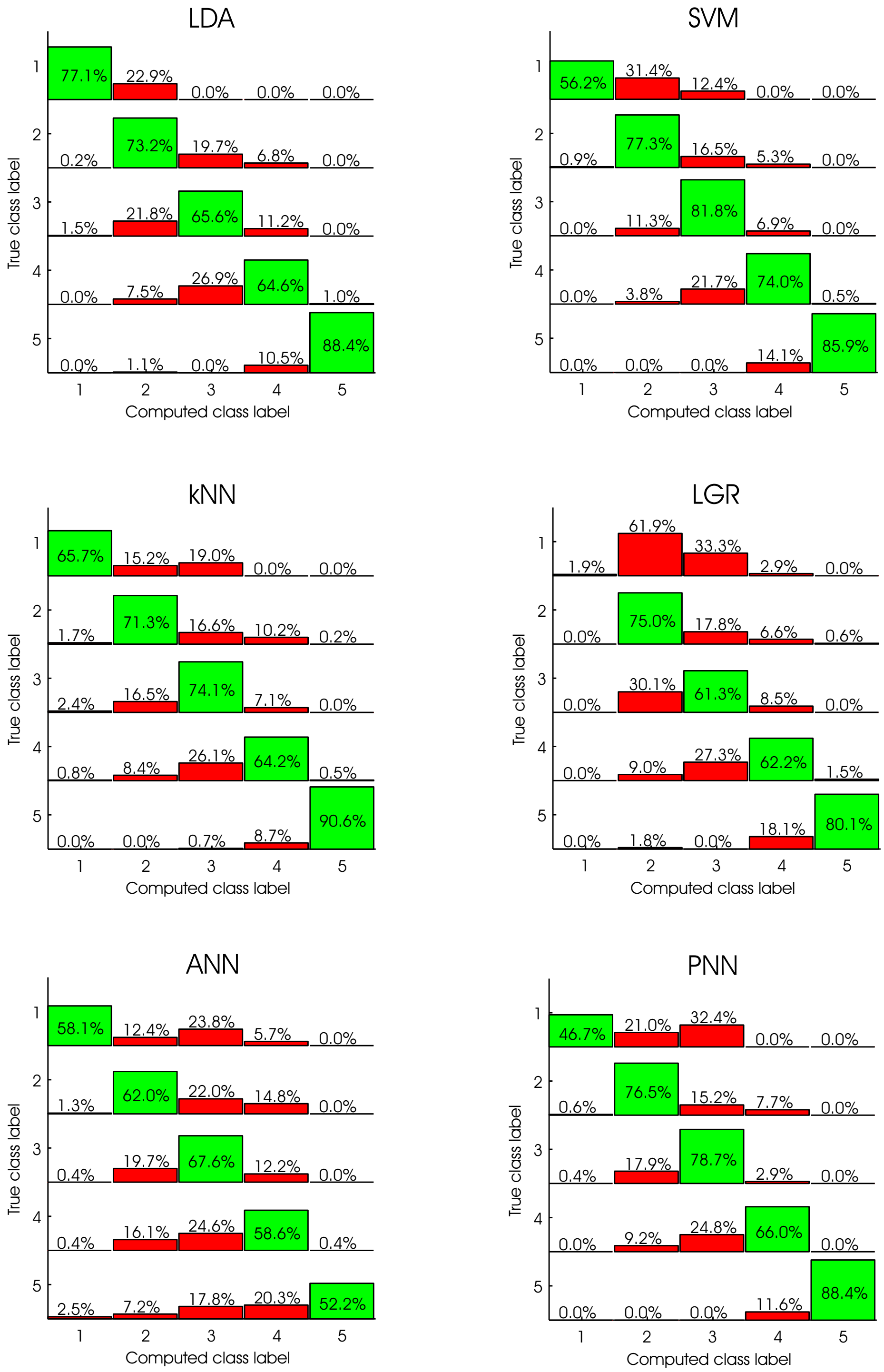


Figure 5.9: Performance cross matrix plots of 6 algorithms (PRGL-1). Plotted are computed class labels vs. true class labels. The main diagonal represents accurate classifications, all other elements denote misclassifications.

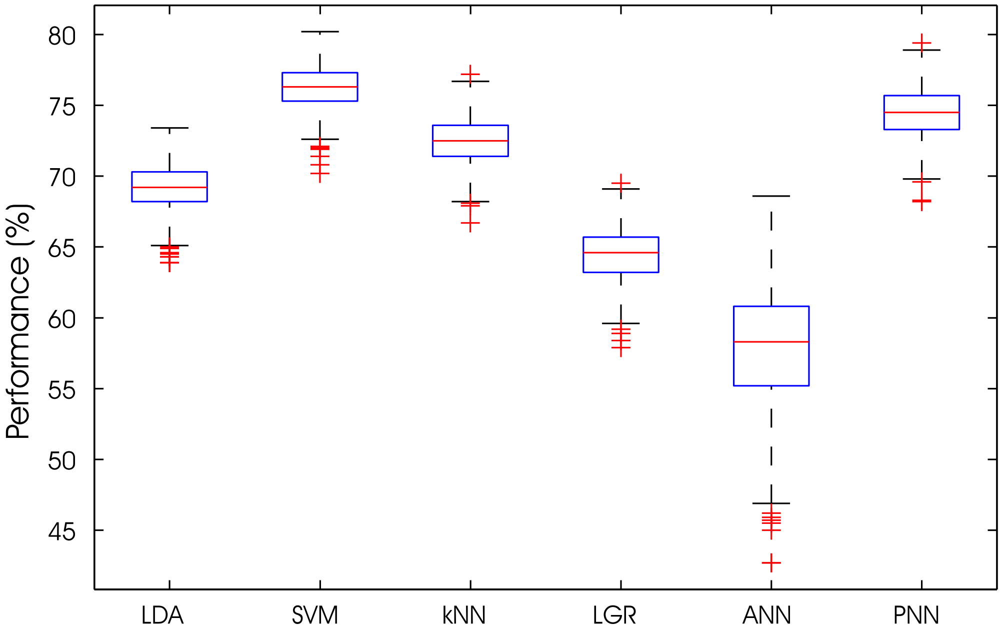


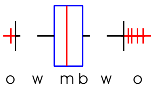
Figure 5.10: Box-and-whisker-plot of performance variances using 600 different training sets (PRGL-1). Of all available data, 5 % were assigned to the training set, 95 % to the test set. Best median is achieved by SVM, followed by PNN and kNN. ANN shows large variations. Whisker length is the interquartile range ×1.5 . Crosses denote outliers.

identify the *general* performance of each classifier, 600 training and test data set combinations were generated and classified. The box-and-whisker plot[1] of Figure 5.10 shows the medians, interquartile ranges and outliers for all algorithms. Clearly, SVM performs best, closely followed by PNN and kNN algorithms. At times, LDA may show equally good results. LGR has a generally poorer performance but with a variance much the same as the better performing algorithms. ANN not only performs worst but also varies considerably in terms of its performance output. It seems that this variation of more than 25 % is mainly attributed to different initial weights.

The analysis of the performance dependency on the size of the training data set reveals that classifications improve with more training data (Figure 5.2). The results worsen considerably with less than 5 % of data (equivalent to 148 data points for PRGL-1) and reach values of above 95 % of accurate classification when the algorithms are trained on SVM and kNN with half of all data (50 %). A classification match of 90 % can be achieved with one third of all training data. Especially ANNs benefit from more training data and outperform many other algorithms when trained

---

[1] The box-and-whisker plot illustrates a data distribution in the following way: centre line (m) is the median; box (b) is the interquartile range (containing 50 % of the data); whisker length (w) is 1.5× interquartile range (b); crosses (o) are outliers.



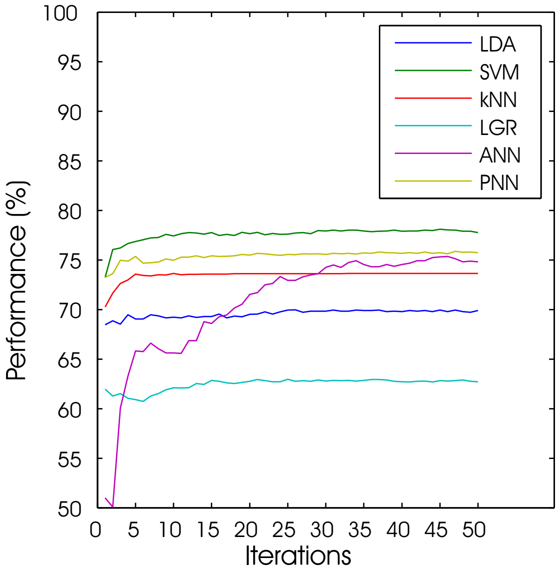


Figure 5.11: Classification improvement with repeated bagging (50 times, PRGL-1). From a training data set (with 5 % of all input data), subsets were created that contained 80 % of these training data (i.e. 4 % of all data). For each classifier, performance is plotted for a majority classification vote of n subsets, $n = 1, ..., 50$.

with 50 % training data. LDA is quite the opposite in that it does hardly improve with more training data at all. LGR shows some improvement though at a generally low level.

Bagging

Bootstrap aggregation was applied to all algorithms in order to evaluate its feasibility in terms of improvement and robustness. As shown in Figure 5.11 most classifiers do not improve much, but bagging ensures that the best result for a given training data set will be achieved. The mediocre performance of logistic regression (LGR) does not improve much though. Remarkably, ANN shows dramatic improvement when bagged. It then even outperforms the kNN algorithm, getting boosted by almost 25 %. It seems that bagging is partly able to remove the negative effects of the neural network's feature of large performance variance (Figure 5.10).

5.1.6 Input Data Selection

So far, all classifications of borehole logging data from PRGL-1 have been fed with 18 logging curves. The focus in this section is on reducing this amount of curves and gain some insight as to which curves do contribute the most for the class dis-

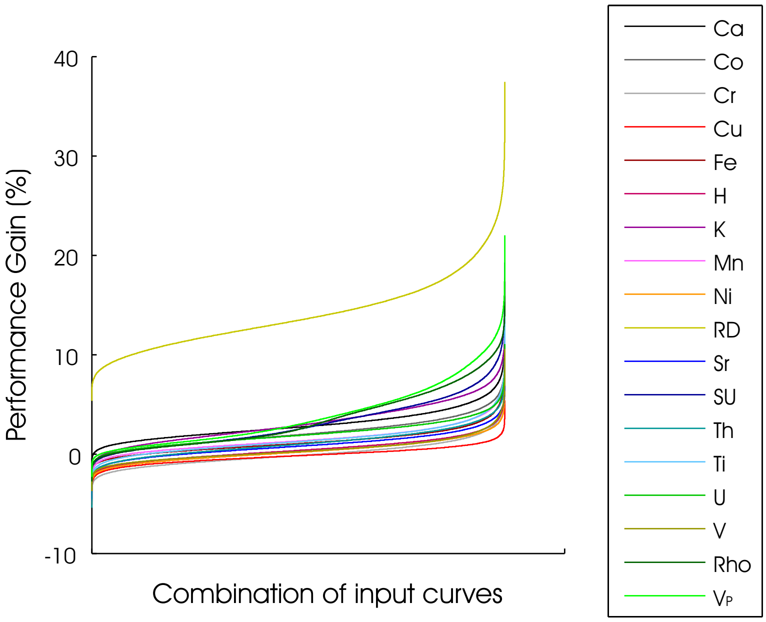


Figure 5.12: Performance gain when excluding and including each logging curve (sorted from left to right, PRGL-1). Classification performance increases considerably when including mean micro-resistivity (RD). Including Cu, Cr, Ni, V or H to the input training data has little or negative effect on the performance.

crimination and which ones do less so. For this purpose, the SVM algorithm (using parameters C=7 and σ=0.4) was run with every combination of input curves, which amounts to $2^{18}-1$ calculations. For each logging curve, the combination pairs (including and excluding that specific curve) for each input pattern were extracted and their difference calculated, giving the gain (or loss) of performance due to that respective curve being included in the classification process. The results are plotted in Figure 5.12. Most striking is the performance improvement when including the mean micro-resistivity. *Vice versa*, excluding resistivity from the training process would make the performance drop by 10–20 % and in extreme cases by 37 %. This fact comes as a surprise since the resistivity curve shows no obvious signs of its excellent discriminating characteristic (see Appendix A). Other curves that appear to be most useful for the classification algorithm are sonic velocity (V_P), density (Rho), susceptibility (SU), Potassium (K) and Calcium (Ca).

On the other side, the elements Cu, Cr, Ni, V and H either do not improve the result significantly or even worsen it. To demonstrate the difference, the (sorted) performance values for in- and excluding resistivity and Copper are plotted in Figure 5.13. The better the curve selection performs the smaller is the gain. The largest effect occurs when performance is generally poor (on the left hand side). From the box-and-whisker plot (Figure 5.14) of the same data it can be deducted that literally all

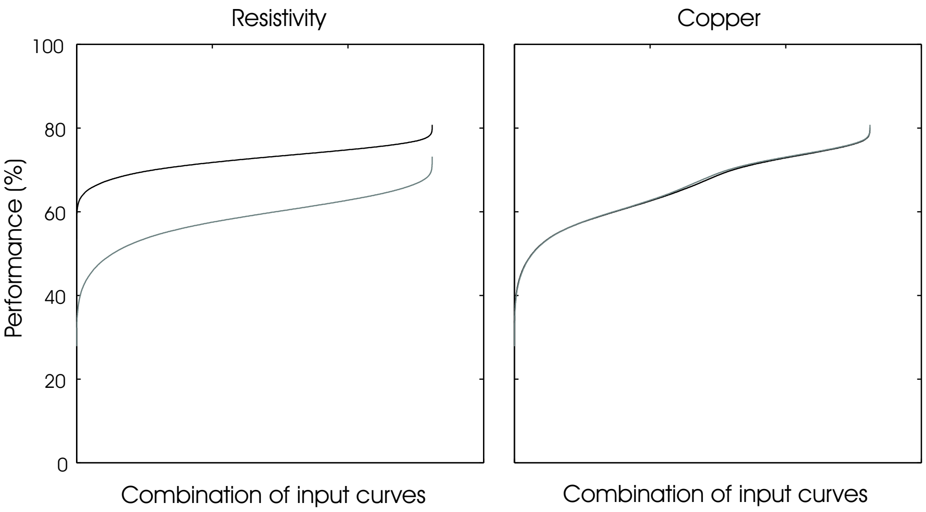


Figure 5.13: Classification performances when in- and excluding RD (left) and Cu (right). Black curves show performance when including the respective data curve, light grey curves show performance when excluding that curve. Data from hole PRGL-1

curves have many outliers, especially above the median. Again, the superiority of the resistivity is striking. It should be noted that the best performance was not achieved with all 18 log curves. Instead, only 10 curves were necessary for this task, as shown in Table 5.4. Using all curves is a little less successful. Remarkably, using only the best 6 curves derived from Figure 5.14, a very good result can also be achieved. Using the same curves, but not resistivity, makes the result drop by almost 15 % though.

Also tested was the scenario in which only curves were used that are acquirable by downhole logging tools. This would simulate a situation in which no core data would be available (say, due to poor core recovery). Enjoyably, this combination proofed to be well performing, just a little less successful than if all available curves were used (see also Table 5.4).

Curves	Classification match
All 18 curves	77.6 %
Best result (Ca, Co, H, K, Mn, RD, SU, Ti, Rho, V_P)	80.8 %
Best result without RD (Ca, Co, H, K, Mn, SU, Ti, Rho, V_P)	67.5 %
Best 6 discriminators (RD, V, Rho, SU, K, Ca)	74.1 %
Best 5 discriminators without RD (V_P, Rho, SU, K, Ca)	59.7 %
Wireline curves only (Ca, Fe, H, K, RD, SU, Th, U, Rho, V_P)	77.0 %

Table 5.4: Some performance results of PRGL-1 with varying curve combinations.

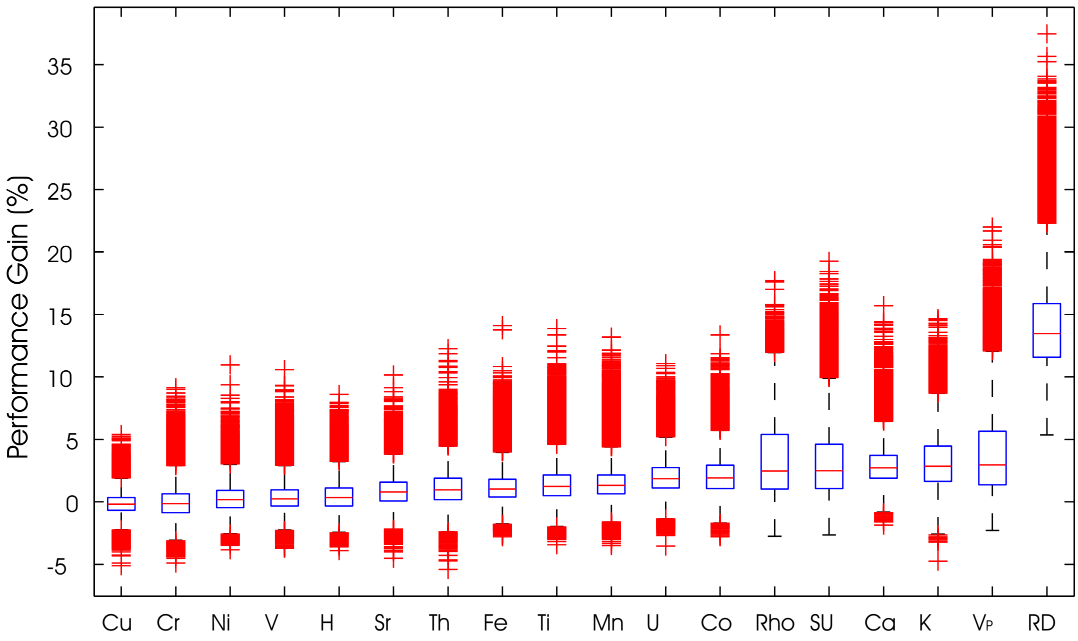


Figure 5.14: Box-and-whisker plot of performance gain when excluding and including each logging curve (medians sorted from left to right, PRGL-1). Classification performance increases considerably when including resistivity (RD) and also sonic (V_P), density (Rho), susceptibility (SU), Potassium (K) and Calcium (Ca). Including Cu, Cr, Ni, V or H to the input training data has little or negative effect on the performance.

5.2 Adriatic Sea

5.2.1 Conventional Interpretation

Existing seismic data from various surveys prior to PROMESS-1 suggest the presence of 4 erosional surfaces ES1 - ES4 on top of 4 progradational units (see Figure 2.5). These units were deposited mainly during glacials (Trincardi & Correggiari, 2000). As previously done for hole PRGL-1 in the Gulf of Lion, the XRF-derived Ca/Fe ratio was utilised as an indicator of detrital *vs.* pelagic regimes and distance to the shore line. The ratio (Figure 5.15) is in line with the erosional surfaces but further subdivisions can be made accounting for intervals with rather slow decrease of Ca/Fe accounting for intervals where erosion has not removed all of interglacial accumulated sediments. This would add an extra boundary in addition to those established by Trincardi & Correggiari (2000) which is denoted as ES3+. Subsequently, the logged interval (45-71 mbsf) was divided into 4 sequences as depicted in Figure 5.15. The logging interval had been restricted to those 26 metres because wireline logging was not possible above 44 mbsf.

Visual core descriptions show mainly silty clay throughout the entire interval with

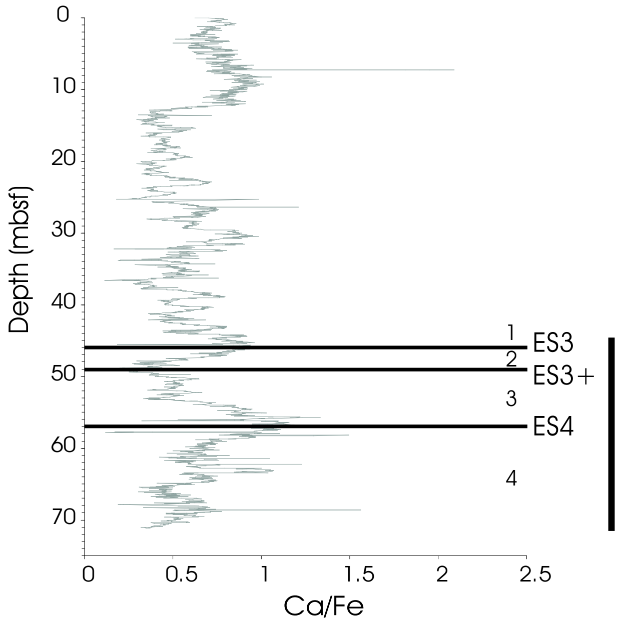


Figure 5.15: Ca/Fe plot and sequence boundaries ES3 and ES4 (PRAD-1). Class boundaries are numbered from 1 to 4. Logging interval is indicated by the black line to the right.

other occurrences being clay and silt. Again, other interpreted core data are not available yet. Attempting to deduct the sequence boundaries from conventional log interpretation is difficult (see Appendix B). Though there are some good indicators (Ca, Fe, Co, density, sonic), other curves do not show changes at the erosional surfaces ES3 or ES4.

5.2.2 Data Used

From available XRF, MSCL and wireline logging data, the following 17 curves were selected for further application in learning algorithms:

- Density (Rho) and sonic velocity (V_P) from MSCL measurements;
- Calcium (Ca), Cobalt (Co), Chromium (Cr), Copper (Cu), Iron (Fe), Potassium (K), Manganese (Mn), Nickel (Ni), Strontium (Sr), Titanium (Ti) and Vanadium (V) from XRF measurements;
- Thorium (Th), Uranium (U), mean electrical micro-resistivity (RD) and magnetic susceptibility (SU) from downhole logging measurements.

For technical reasons, no wireline geochemical data were acquired at PRAD-1. From XRF logs, Zinc and Lead log curves were rejected because of noise. Sample rate for

Factor	Eigenvalue	Variance (%)	Cumulative Variance (%)
1	**5.13**	30.18	30.2
2	**2.48**	14.59	44.8
3	**1.86**	10.92	55.7
4	**1.27**	7.48	63.2
5	**1.12**	6.58	69.8
6	**1.01**	5.96	75.7
7	0.83	4.87	80.6
8	0.66	3.90	84.5
9	0.63	3.70	88.2
10	0.50	2.96	91.1
11	0.39	2.32	93.5
12	0.35	2.04	95.5
13	0.26	1.54	97.1
14	0.18	1.04	98.1
15	0.17	1.01	99.1
16	0.08	0.48	99.6
17	0.07	0.41	100.0

Table 5.5: Factor analysis results from log curves of hole PRAD-1: Eigenvalues, variance and cumulative variance. 76 % of the data set's variance can be represented by 6 factors.

the interval 45-71 mbsf was 5 cm. The total data set size was therefore $519 \times 17 = 8823$ values.

Again, the depth information z was not used for reasons outlined in the discussion (Chapter 6).

5.2.3 Factor Analysis

In order to reduce the input data dimensionality, a factor analysis was performed with the selected 17 log curves as described in Section 4.1.3. The results are given in Table 5.5. Six factors score an eigenvalue above 1 with a cumulative variance of 76 %. Replacing the 17 log curves by those 6 factors (thus reducing the dimension by about two thirds) would discard almost a quarter of the data's variance. This measure was regarded as being too costly and therefore rejected. The classification tasks were performed with the original (standardised) 17 log curves.

5.2.4 Data Distribution

Normal probability plots and histograms were created for the selected 17 log curves (Appendices D and E). Whereas all curves fail the χ^2 test on normal distribution, 11 out of 17 curves pass the Kolmogorov-Smirnov test (see Table 5.6). As previously

Log Curve	χ^2	Kolmogorov-Smirnov
Ca	1889	**0.039**
Co	83	**0.026**
Cr	1×10^5	**0.050**
Cu	2×10^5	0.078
Density	372	**0.059**
Fe	213	**0.030**
K	5×10^5	**0.058**
Mn	1336	**0.055**
$\ln$ Susceptibility	158	**0.034**
Ni	90	**0.029**
$\ln$ Mean micro-resistivity	178	0.081
Sr	8×10^{29}	0.237
Th	7152	0.147
Ti	7875	**0.038**
U	1×10^5	0.254
V	5×10^{20}	0.088
V_p	30797	**0.034**
Critical value	69	0.059

Table 5.6: Tests of distribution on PRAD-1 log curves. Left: χ^2-test (which no curve passes). Right: Kolmogorov-Smirnov-Test. Bold numbers (11 out of 17 curves) pass this test (i.e. the null hypothesis that the values come from a standard normal distribution cannot be rejected). Both tests assume a probability of error $\alpha = 0.05$.

discussed in Section 5.1.4, the data set cannot be Gaussian distributed because not all curves are Gaussian. Again, the assumption of normally distributed data will be violated in the subsequent application of learning algorithms.

5.2.5 Supervised Learning

The logging data at hand needs to be subdivided into a training and a test data set prior to be used with a classification algorithm. From the possibilities to generate a training data set (presented in Section 5.1.5), the same approach as used with hole PRGL-1 was chosen, i.e. a fixed fraction of each class was randomly drawn from the data set. Because the number of total data points is rather low (519 values, see Table 5.7), 10 % per class were chosen as training data set, leaving the remaining 90 % for the test data set. This split ratio is used for the entire processing unless otherwise stated. Even though the training data set is twice as big as for PRGL-1, the absolute figures are still very low. For class 1, merely 2 data points are present in the training data set (6 data points for class 2). Nevertheless, this section on classification of PRAD-1 data serves as a case study of shallow boreholes with a rather small data set. Having very little training data at hand may be a demanding task for

Class	Data Points	Training Points	Fraction
1	20	2	10 %
2	56	6	10 %
3	154	15	10 %
4	289	29	10 %
Total	519	52	10 %

Table 5.7: Distribution of training data for each class.

the classifiers but is not unlikely be encountered during geo-scientific operations.

Discriminant Analysis

Linear discriminant analysis was performed and quadratic discriminant analysis was attempted. As shown in Figure 5.16, QDA needs more than 85 % of data, equalling 17 training samples out of 20 available samples of class 1 (see Table 5.7). For the classification with 10 % training and 90 % test data, only LDA was used.

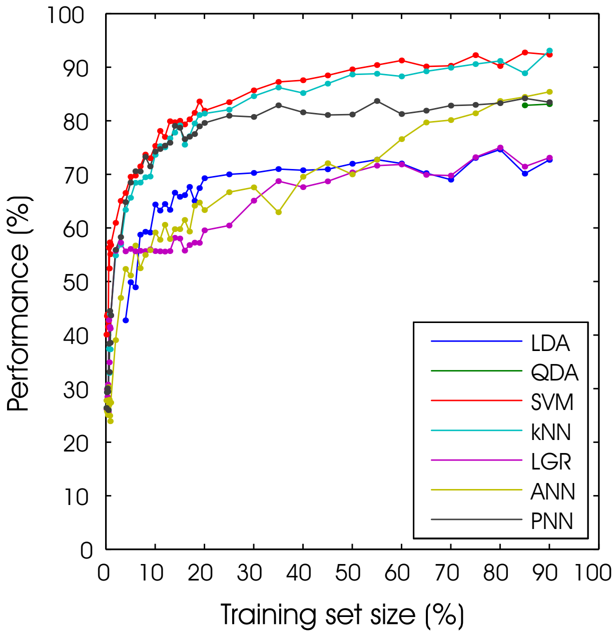


Figure 5.16: Performance with varying training data set size (in percentage of total available data, PRAD-1). The mean performance value from 5 classification runs with different randomly selected training sets is shown. Quadratic discriminant analysis (QDA) only works with more than 85 % of the data.

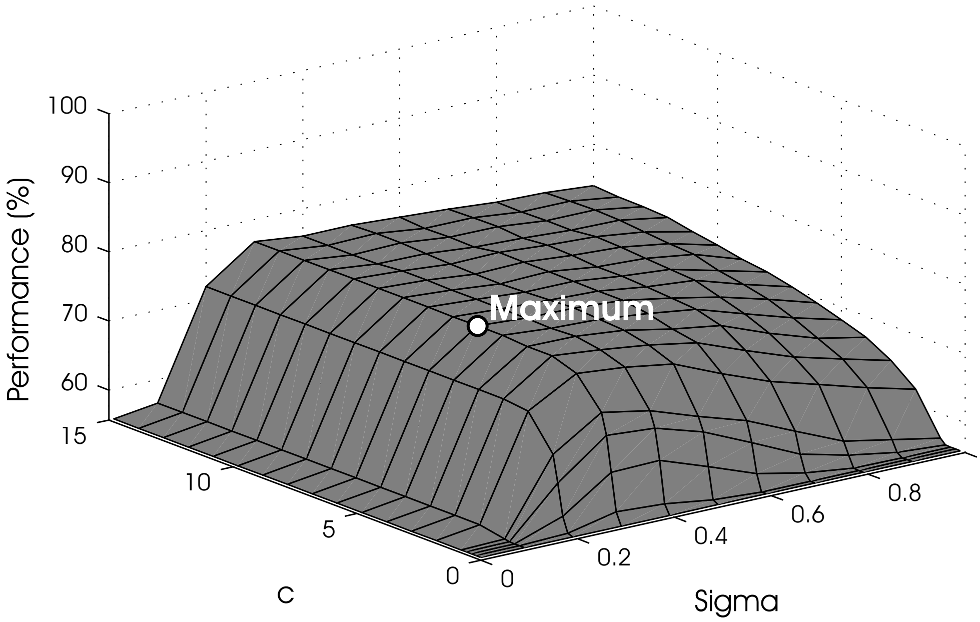


Figure 5.17: Performance of RBF-kernel support vector machines with changing upper bound C and RBF-kernel spread σ (PRAD-1). Of all available data, 10 % were assigned to the training set, 90 % to the test set. A mean performance value was calculated from 5 data sets. Performance drops considerably for values of C < 2 and $\sigma < 0.2$. Best performance is at C = 6 and $\sigma = 0.3$.

k-Nearest Neighbour Estimation

The k-nearest neighbour estimation was run with varying k. Like the results from PRGL-1, here again the best k value is 1, thus implementing the nearest neighbour rule.

Logistic Regression

Logistic regression (LGR) was performed with 10 % training data. As the absolute number of data points of PRAD-1 is considerably less than with PRGL-1, LGR was successfully tested on other training/testing data set ratios as shown in Figure 5.16. Computing times increased with more input data but were still justifiable.

Support Vector Machine

Similar to the parameter findings in Section 5.1.5, the SVM was implemented with a RBF kernel and an one-*vs.*-all (OVA) multiclass transformation (see Section 4.3.7). As shown in Figure 5.17, the best upper bound C is 6 and the best RBF spread σ is 0.3 .

Backpropagation Neural Network

Similar to results presented from hole PRGL-1, backpropagation neural networks (ANNs) are robust with respect to net topology and stop error condition. Parameter ranges that were tested are 17 to 60 hidden neurons and 1×10^{-1} to 1×10^{-12} for the stop error condition, respectively. Either parameter combination is similar to any other for the given training set size of 10 %. Eventually, a 17-24-4 net topology was chosen (17 log curves, 24 hidden neurons and 4 stratigraphic sequences). The stop condition was set to $e_{\text{stop}} = 5 \cdot 10^{-4}$.

Probabilistic Neural Network

PNN was implemented with a 17-52-4 net topology (17 log curves, 52 training data points, 4 class labels). The topology is entirely determined by the problem at hand, and the only parameter to be determined by the user is the size of the Parzen window σ. Similar to results from borehole PRGL-1, a value of $\sigma = 0.2$ yields the best performance.

Comparison

Similar to the classification results of hole PRGL-1, the above described algorithms (LDA, SVM, kNN, LGR, ANN and PNN) were used with 17 log curves from hole PRAD-1. A typical result is shown in Figure 5.18. A smoothed version with a 3 m sliding window like proposed at borehole PRGL-1 was not attempted due to the short logging interval (26 m). Clearly, the identification of class 1 (red) is not accomplished by either algorithm. Evidently, the very low number of training samples (just 2, see Table 5.7) prohibits a successful classification in this extreme situation. The discrimination between classes 2 (blue) and 3 (green) appears to be equally challenging. The neural network gives the best result here, although this may be due to accidentally advantageous initial weights. Again, the proper detection of class 2 is severely affected by the low number of training samples (6). Classes 3 and 4 (orange) are properly separated by all classifiers except LGR, which completely fails to detect any classes other than class 4. The match results given below the figures have to be considered with care, as the majority of (correct) classifications takes place within class 4. The detection of class boundaries is in fact independent of a good classification match. For instance, SVM performs best in total but does not accurately detect classes 2 or 3.

Looking at the performance cross matrix of hole PRAD-1 (Figure 5.19), the results reflect the findings from the classification columns just discussed. Classification performance is dependent on the number of available training samples. Especially

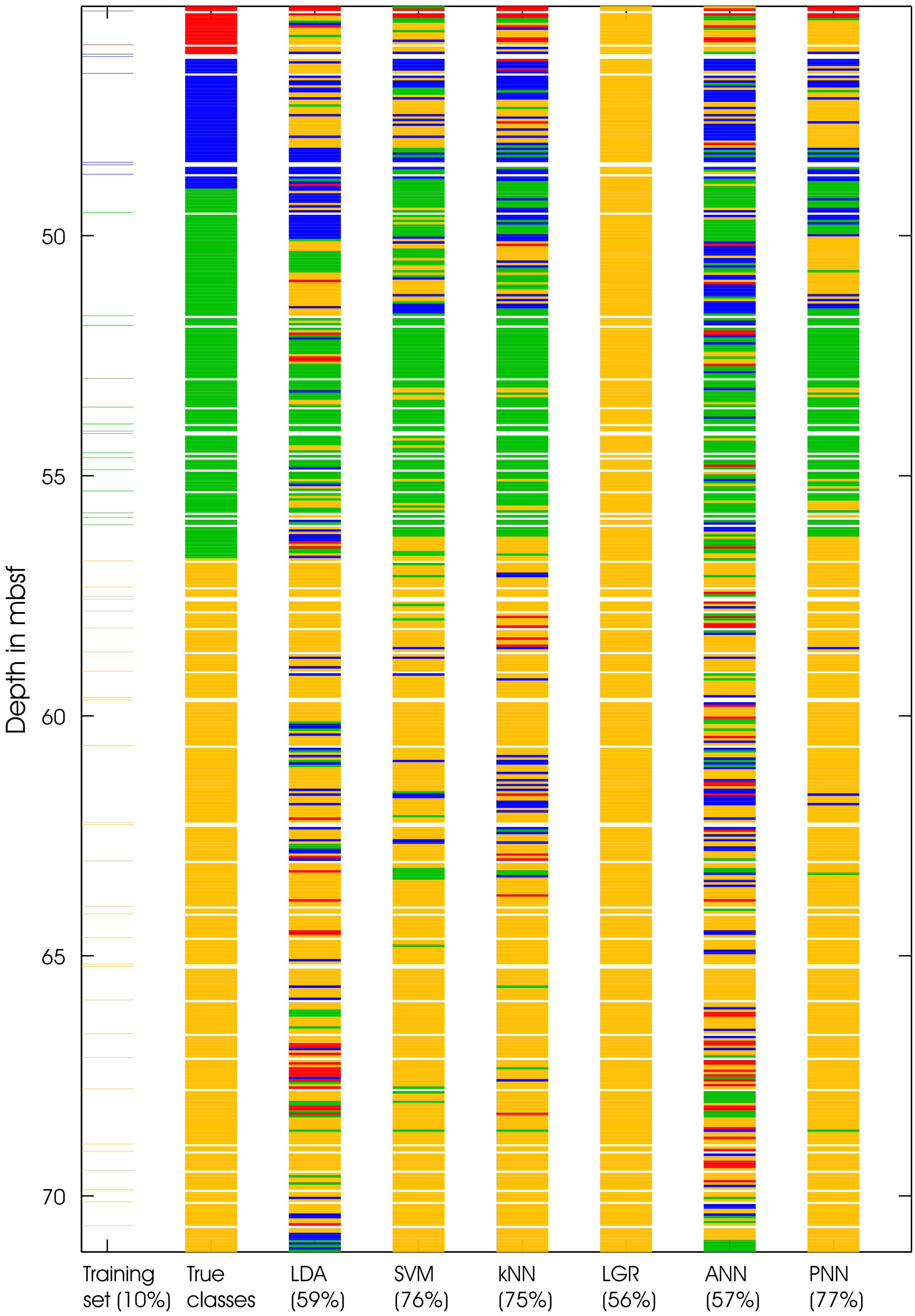


Figure 5.18: Classification of hole PRAD-1. From left to right: Training data set (10 % of all available data per class); true class labels (sequence 1 = red, 2 = blue, 3 = green, 4 = orange); linear discriminant analysis; support vector machine; k-nearest neighbour; logistic regression; artificial neural network; probabilistic neural network. For parameters and processing details refer to text.

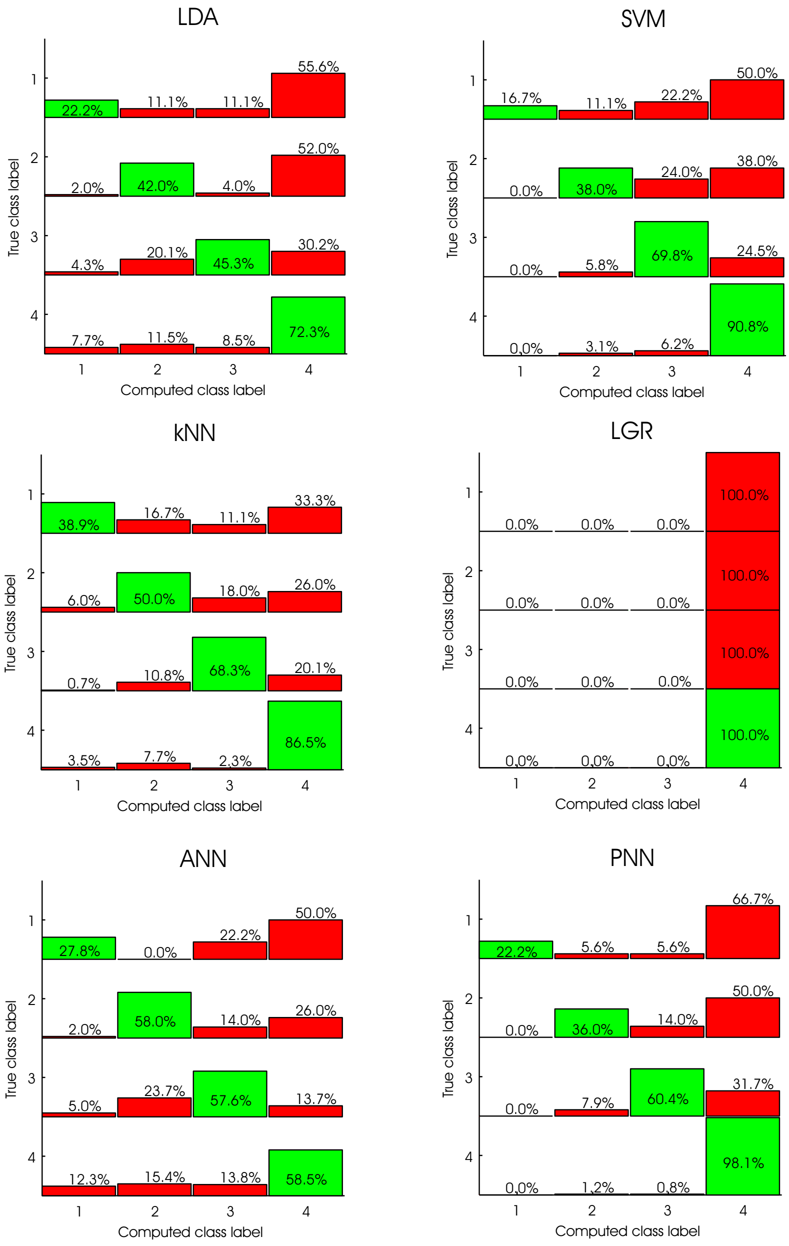


Figure 5.19: Performance cross matrix plots of 6 algorithms (PRAD-1). Plotted are computed class labels vs. true class labels. The main diagonal represents accurate classifications, all other elements denote misclassifications.

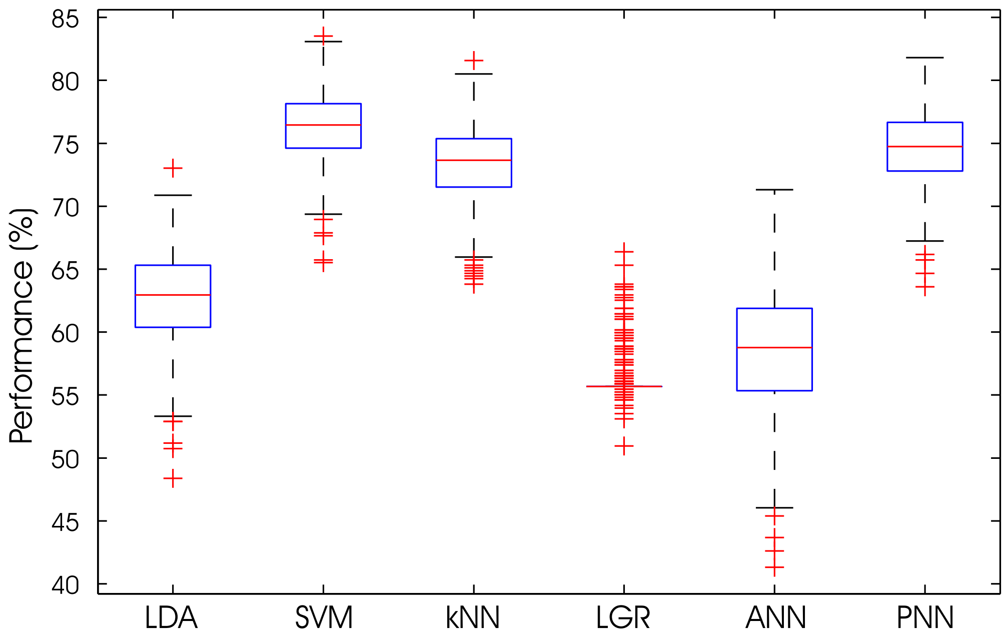


Figure 5.20: Box-and-whisker-plot of performance variances using 600 different training sets (PRAD-1). Of all available data, 10 % were assigned to the training set, 90 % to the test set. Whisker length is the interquartile range × 1.5 . Crosses denote outliers. Best median is achieved by the SVM, followed by PNN and kNN. LDA and ANN show larger variations. LGR fails completely.

classes 2 and 3 are often misclassified (and associated with class 4). Only ANN shows a satisfying pattern. LGR does not work altogether because of too few training data.

Again, the so far discussed results represent only one out of many cases because of the randomly selected 10 % of training samples. In order to assess the algorithms' performance in a more general way, 600 classifications with different training data were computed. The results are plotted in Figure 5.20. Overall, SVM, kNN and PNN seem to perform best, while LDA and ANN are less successful. LGR is "locked" at a 56 % match, which accounts for a uniform classification into class 4 only (as in Figure 5.18). The figure may have a limited significance only, as the identification of classes 1-3 may not be reflected by a good overall result.

Clearly, classification in case of hole PRAD-1 logging data with 10 % training and 90 % test data is an extreme case at each classifiers limit. Increasing the number of training samples will therefore significantly improve the performance as shown in Figure 5.16. kNN and SVM perform equally well, reaching a 90 % match when trained with 60 % or more data. ANN benefits as well from more training data though on a generally lower level. LDA and LGR do not perform better than 70% regardless of the number of training samples.

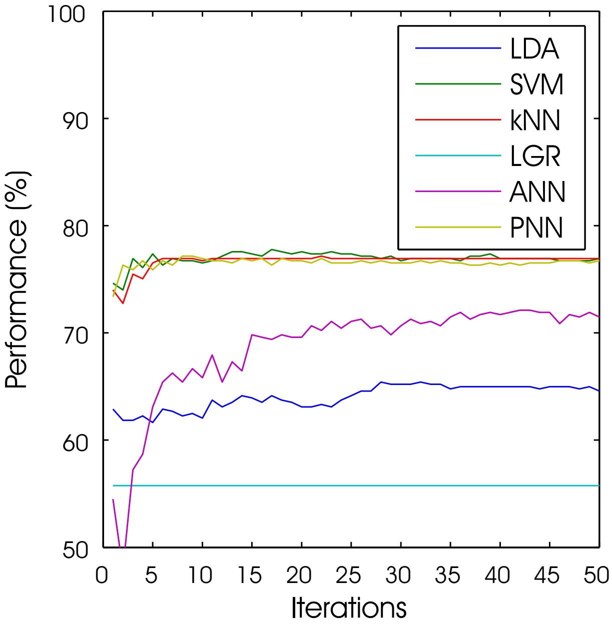


Figure 5.21: Classification improvement with repeated bagging (50 times, PRAD-1). From a training data set (with 10 % of all input data), subsets were created that contained 80 % of these training data (i.e. 8 % of all data). For each classifier, performance is plotted for a majority classification vote of n subsets, $n = 1, ..., 50$. Only ANN benefits considerably from bagging.

Bagging

Bagging was performed with 10 % of all data and subsets that comprised of 80 % of these samples (equals 8 % of all data). 50 bagging operations were performed and the result was plotted after each bagging (Figure 5.21). None of the algorithms benefits much from bagging with the exception of ANN. The latter can be improved by more than 15 % by means of bagging, a result that clearly justifies this additional processing step.

5.2.6 Input Data Selection

Similar to Section 5.1.6, a SVM algorithm with parameters C=6 and σ=0.3 was run with all $(2^{17} - 1)$ possible logging curves combinations. Again, the performance for each combination with and without a given curve was compared, sorted and plotted (Figure 5.22). Though the feature is not as striking as with hole PRGL-1, the resistivity curve (RD) has once more the largest impact in terms of classification match for borehole PRAD-1 (an increase between 5 % and 15 %). Other log data that also discriminate the proposed classes properly are Manganese, Thorium, density and Titanium (see also Figure 5.23). Including Ni, Cr, Cu or Ca does not improve the

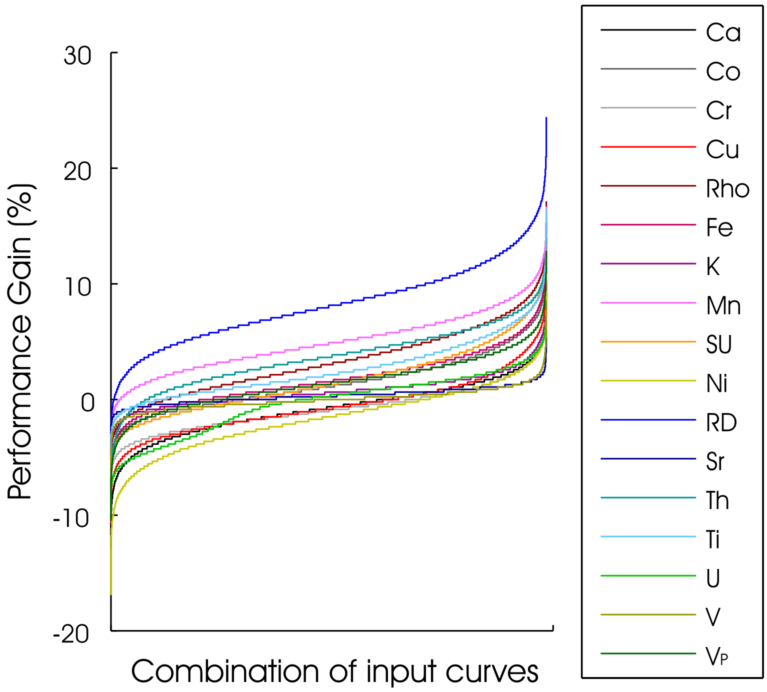


Figure 5.22: Performance gain when excluding and including each logging curve (sorted from left to right, PRAD-1). Classification performance increases considerably when including mean micro-resistivity (RD). Including Ni, Cr, Cu or Ca in the input training data has little or negative effect on the performance.

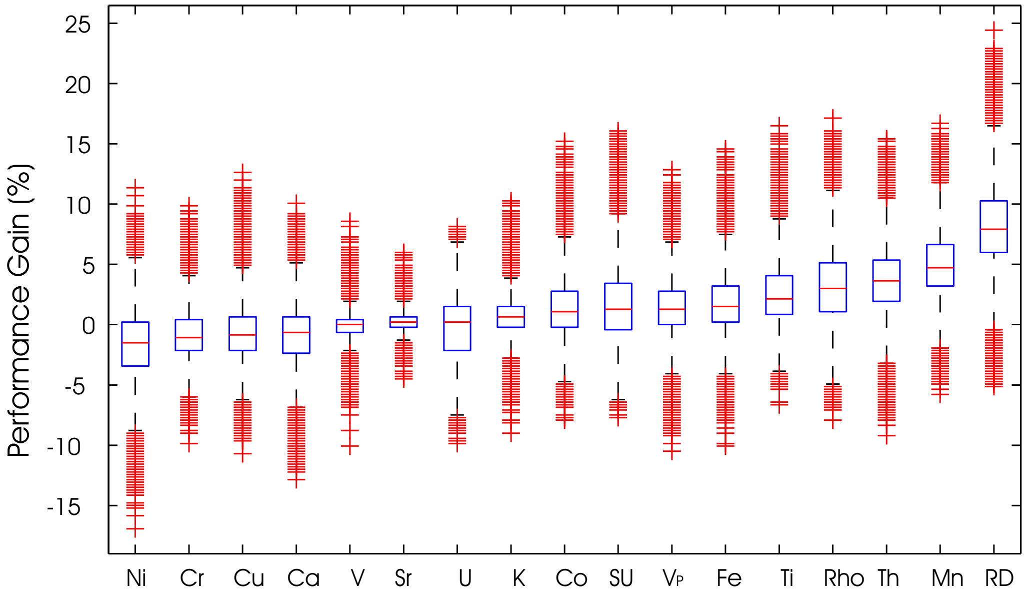


Figure 5.23: Box-and-whisker plot of performance gain when excluding and including each logging curve (medians sorted from left to right, PRAD-1). Classification performance increases considerably when including resistivity (RD) and also with Manganese (Mn), Thorium (Th), density (Rho) and Titanium (Ti). Including Ni, Cr, Cu or Ca in the input training data has little or negative effect on the performance.

Curves	Classification match
All 17 curves	75.6 %
Best result (Co, Fe, K, Mn, RD, Sr, SU, Th, Ti, U, V)	84.2 %
Best result without RD (Co, Fe, K, Mn, Sr, SU, Th, Ti, U, V)	74.7 %
Best 5 discriminators (RD, Mn, Th, Rho, Ti)	77.3 %
Best 4 discriminators without Rt (Mn, Th, Rho, Ti)	68.3 %
Wireline curves only (Ca, Fe, K, RD, Rho, SU, Th, U, Vp)	76.9 %

Table 5.8: Some performance results from PRAD-1 with varying curve combinations.

classification match but may even worsen it. While the negative effect of Ni, Cr and Cu is in line with findings from hole PRGL-1 (see Section 5.1.6), the bad performance when using the Ca curve is the opposite to the case in the Gulf of Lion where Ca was found to be the 4th best discriminator (see Figure 5.14). These results are somewhat different to those of borehole PRGL-1 (Figure 5.14). Apparently, the marine clays are of slightly other mineral composition than those at the Gulf of Lions.

For a randomly chosen training set of 10 % of all data, Table 5.8 shows that the best result with only 11 out of 17 curves has an almost 10 % higher classification match than when using all curves. Excluding the resistivity from the classification task will lead to a performance drop of about 10 %. Using wireline data only would yield a similar result as if using all XRF, MSCL and wireline log data.

6 Discussion & Conclusions

Having outlined the acquisition of (wireline, MSCL and XRF) logging data, the PROMESS-1 project and several classification techniques, the previous chapter presented the results of their application to PROMESS-1 data. These results will now be discussed with emphasis on the comparison of algorithms and the findings regarding the importance of log curves to the successful classification process. A summary and outlook conclude this chapter.

Data Input

Reducing the number of log curves and thus input dimensions by means of factor analysis proved to be unfeasible. At both locations PRAD-1 and PRGL-1, the reduction of log data to factor logs would have meant discarding a too large amount of data variance (24 % and 30 %, respectively). Evidently, PROMESS-1 data show too little variance to be utilised for a dimensional reduction. Subsequently, no attempt was made to give these factors any geological or geophysical meaning, as the classification processing was performed with the actual (normalised) log curves. There are examples in the literature where factor analysis has been successfully performed on log data. Bücker et al. (2000) report the dimensional reduction down to 3 factor logs accounting for more than 80 % of the data's variance. Regarding PROMESS-1 data, the majority of log curves deal with chemical elements derived from the XRF scanner. Although they reflect changes of composition of the marine clay, their variance may not be as high as with physical parameters such as magnetic susceptibility and density. In conclusion, the selected log curves cannot be reduced to any sufficiently low number of factors that would explain a decent amount of data variance.

Training Data Generation

A major application of supervised learning to log data in real-life situations and a main motivation of this work is possible poor core recovery (or, like with PROMESS-1 data, the assumption of it). In cases where there are little or no cores available, wireline logging data are the only *in situ* measurements at the scientists' disposal. Both MSCL and XRF scanners need successful sampled and properly preserved cores. If core recovery is less than 100 %, there is little chance to gain information regarding the missing intervals. Reducing the (known) total data to a subset of it (the training data set) simulates a less-than-complete core recovery with missing intervals. The investigation of classification performance in such (simulated) situations aims to examine if this tool is still useful for the geoscientist even if or especially when there are no core data available.

As repeatedly mentioned before, of all available (wireline, MSCL and XRF) data a certain fraction of it was used to train the classification algorithms. These trained classifiers were then used to properly label the remaining (test) data according to their respective classes. The actual ratio of training-to-test data was chosen arbitrarily. However, the figures were intended to be challenging for the algorithms and at the same time still produce reasonable results. It was demonstrated that in the case of quadratic discriminant analysis and logistic regression, limits are tighter than with the other algorithms analysed. QDA cannot be performed with less than p samples per class, p being the number of log curves. On the other hand, presenting >1800 data points to the LGR algorithm made the classification task increasingly time consuming but not better in terms of performance. Having a core recovery of only 5 % during a scientific campaign would be a very disappointing result. In fact, much higher recovery rates are almost always encountered and as Figures 5.2 and 5.16 show, these will lead to very satisfying classification matches (e.g. >85 % with one third of training data).

With less than 100 % core recovery, the missing intervals are likely to be linked to regions with less compaction, less adhesive matrix structures, changes in grain sizes or just with technical incidents during the drilling and/or coring process. Eventually, the position of these intervals are inherently unpredictable. This fact is represented by drawing the training data points randomly from the total data pool, as opposed to using a fixed drawing interval (e.g. make every 10th data sample a training data sample). In both cases PRAD-1 and PRGL-1, the shares of each class of the entire interval are not evenly distributed. This is especially true for class 1 in both boreholes that only represents less than 4 % of the to-be-classified logging interval. To avoid any preference in favour of classes that make up for large portions of the data set (e.g. class 4 at PRAD-1), the sampling of training data points was adjusted such that the subset to be drawn was sampled from each class individually. This ensures a proper representation of each class within the training data set. A typical core with missing intervals would consist of larger blocks with or without sediments than it is the case with this simulated core recovery and its quite selective spots of training sample occurrences. Nonetheless the presence of each class within the training data is essential for a proper classification task and adjusting the training data sampling procedure in this respect only seems reasonable.

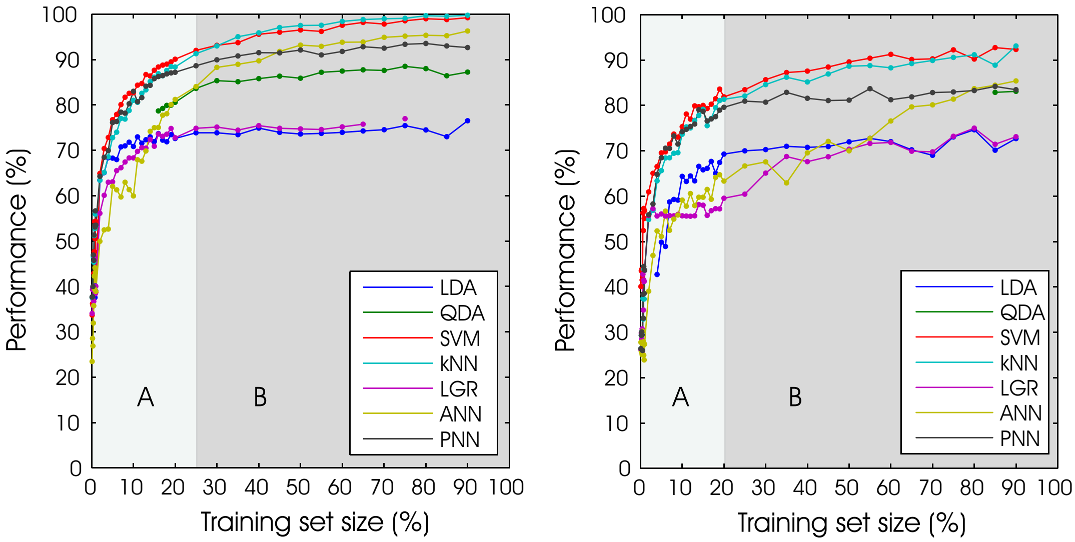


Figure 6.1: Regions of training-to-test data ratio performance. A: Performance is a function of training data size. B: Performance is stable. Data from PRGL-1 (left) and PRAD-1 (right).

Training Data Size

When increasing the ratio of training-to-test data, there are two main regions distinguishable: a range where performance increases drastically when the training set size is raised little (0–25 % and 0–20 %, see Figure 6.1, region A) and a region where performances are more stable even if the size is increased considerably, i.e. >25 % (PRGL-1) and >20 % (PRAD-1). Region A clearly represents a situation where the performance is dependent on the size of the training data set. With 1 % or less training data, the classification match is not any better than mere chance. When the size of the training data set reaches a certain level (region B), the performance only changes little. This is true for all investigated algorithms but ANN. Here, more training data also means better performance, especially when the absolute number of training samples is generally low (as with PRAD-1, right hand side of Figure 6.1). Region A represents the demanding case where core (meaning training) data are sparse and which is most difficult for any classification algorithm. Region B on the other hand is a more ideal situation with ample training data samples available. The typical core recovery rates to be expected during a scientific campaign fall in this region.

The previously presented training set sizes of 5 % (PRGL-1) and 10 % (PRAD-1) fall in region A. Admittedly being a rather demanding task for the learning algorithms, such ratio represents the lower limit of what is still feasible (see Figure 6.1). Presenting even less training data to the algorithms would both be unrealistic (a 5 % core recovery rate of a 20 m core is equivalent to having only sampled 1 m of sediments) and pointless in terms of a scientific statement. Summarising the results presented in Chapter 5, for PRGL-1, SVM, kNN and PNN perform best with this little training, followed by LDA and LGR. ANN shows poor results which is due to too few training samples (Figure 5.10). QDA needs a minimum amount of training data which is dependent on the input space dimension.

Region B represents the range in which performance levels only slightly rise with more training data. While SVM and kNN show minor improvements (5–10 %), PNN, QDA, LDA and LGR remain at almost constant performance levels. The major exception is the ANN algorithm that benefits drastically from being fed with more training data. This fact is also illustrated by the difference between the two boreholes. While at PRGL-1, ANN yields match values of above 90 % with only 50 % of the training data, at PRAD-1 this value is never reached even when the network is trained with 90 % of all data. The absolute numbers of training samples (see Tables 5.3 and 5.7) need to be sufficiently large for ANN to perform well.

Comparison of Classifiers

Apparently, SVM and both the Parzen windows density estimation (implemented by PNN) and the nearest neighbour algorithm manage to approximate best the classification rule for boreholes PRGL-1 and PRAD-1. All three methods are discriminative and non-parametric classifiers (Figure 4.2). As far as SVMs are concerned, the good results are less surprising because the algorithm minimises the generalisation error rather than the training error. This approach is certainly superior to algorithms that minimise the training error only as discussed in Chapter 4. With k set to 1, the nearest neighbour rule classifies new data points exclusively on the basis of the nearest sample point. Seeing the very good results, the remaining error appears to be close to the (best achievable) Bayes error R^* shown in Figure 4.5. Presuming that the nearest neighbour of any given sample point indeed matches with the correct class label in most cases would militate in favour of

- rather simple class boundaries or
- contiguous regions of same class labels within the data space.

This is supported by the fact that the Parzen windows approach yields similar good results. Although its window is a function of the number of training samples n rather than the number of neighbours, the algorithm places emphasis on the surrounding sample points alike. Recalling the ordinary performance of LDA, simple (=linear) class boundaries are less likely. Notwithstanding the fact that SVM is a linear classifier as well, this algorithm is able to perform the linear classification task in a higher dimensional space thanks to its kernel function. Therefore, the class labels of logging data of PROMESS-1 holes PRGL-1 and PRAD-1 are likely to be arranged in contiguous regions with little overlapping. The value of the penalty parameter C of the SVM gives another hint in this direction. A high value $C \to \infty$ means hard class boundaries (i.e. outliers are costly penalised) if σ^2 is held constant. Figures 5.4 and 5.17 show that C is rather small and (more importantly) does not influence the performance when made bigger. In other words, the SVM classifier does not improve much when the boundaries are made more strict which again suggests classes with little overlapping. When comparing the results of LDA and QDA, the latter generally performs more than 15 % better than LDA. This again points towards a generally better aptitude of non-linear classifiers with respect to this kind of log data.

The similar performance of LDA and LGR is explained by the fact that their underlying models are basically the same. They differ only in the way they estimate the linear coefficients. The LGR model is more general as it makes less assumptions regarding the data distribution (LDA requires the data to be Gaussian). Although proven that the log data used within this work are not normally distributed, LDA performs not worse than LGR. This robustness of LDA towards data distribution has been remarked by several authors before (e.g. Hastie et al., 2001). With very little training data, linear class boundaries seem to be an appropriate approximation. When there are more training samples fed to these two algorithms however, this advantage diminishes. This can be seen in regions B of Figure 6.1 where the performance remains almost at the initial (mediocre) level while other algorithms improve their performance drastically.

When comparing different sets of training and test data, the variance of results are very similar for each of the classification algorithms but ANN (Figures 5.10 and —notwithstanding the poor LDA performance— 5.20). The interquartile range (including 50 % of the results) is ≈5 % or less. Solely the backpropagation neural network shows larger variances due to its changing initial weights. These cause varying starting points on the error surface of the gradient descent which then leads to the algorithm's arrival in different local minima.

This underlying negative property of the neural net can be compensated for by bagging as pictured in Figures 5.11 and 5.21. Clearly, using multiple ANN classifiers and effectively stacking them such that the majority vote determines the final class label removes possibly harming isolated cases of classification. However, bagging not only moves the classification result towards a mean value (or median as in Figures 5.10 and 5.20). It boosts the result beyond that and achieves outputs that equal those of the top performing SVM, kNN and PNN algorithms. The reason for this is that resampling techniques such as bagging use the *final* accuracy (after being applied to the test data) of a classifier. This can also be interpreted as finding the best model complexity by improving the variance (accuracy).

Regarding the failure of the logistic regression at borehole PRAD-1, 10 % of training data are clearly too few to successfully yield any reasonable results. The algorithm simply assigns all samples to the majority class of the training data set (which in this case is class 4). This behaviour is called *underfitting*. The LDA performance level is reached only when the LGR algorithm is trained with more than 35 % of data (Figure 5.16).

With classification matches of around 75 % (after training with 5 % and 10 % of the data, respectively) the questions remains: where do the misclassifications occur? Here, the performance cross matrix plots (Figures 5.9 and 5.19) are suitable to quickly detect the main difficulties of an algorithm. In case of borehole PRGL-1, major mismatches occur in class 1, especially with the LGR algorithm. The least errors are encountered in class 5, which somewhat weakens the reason of too little training data. However, the amount of training data is influencing the classification performance but it is not the only factor. Where the input data are easily separable into different classes, good class boundaries can be established (here: the sequence boundary between classes 4 and 5, see Figure 5.7). If the classes are more difficult to separate, little training data make the labelling process a challenging task (as with class 1 in this case). Sequences 2–4 are generally well recognised, though the boundary between class 2 and 3 is thought to be shallower ($\approx$3 m) by the algorithms. This is a hint for a possibly inappropriate training data set. With more training samples, the sequence boundary is met more accurately (see Appendix F).

Within each sequence, misclassifications happen randomly but may be clustered when there are larger intervals without training data (e.g. at 161 mbsf or at 190–195 mbsf, see Figure 5.7). This phenomenon can be repeated with different training data sets (after all, Figure 5.7 is just an example of one out of many possible results with varying training data sets). Apparently, there are local properties that can only be accounted for when the training data set is covering the entire interval, although this coverage does not have to be complete, of course. Having only core samples of very few selected spots (e.g. the upper first 50 metres) would very likely lead to disappointing results. Having training data from the entire to-be-classified interval (even with larger gaps) is a requirement for successful pattern recognition. This also shows that the true class labels and hence stratigraphical model may be somewhat too simple for the algorithms. Apparently there is more information hidden in the data. Its detection is prevented by the rather simplistic training data set (4 or 5 classes).

The classification of borehole PRAD-1 with 10 % training data serves as an extreme example because the number of training samples is altogether extremely low (see Table 5.7). In this situation, performance is mainly dependent on training data size: the class with most samples (class 4) is matched best, the one with the fewest samples (class 1) is matched worst. When the training-to-test-data ratio is increased, performance is excellent (see Appendix G). This example points to a possibly delicate case when sequences (i.e. classes) are disproportionate in terms of their presence within the training data set. If some classes are weakly represented by the training data, underfitting occurs. Class 1 of hole PRAD-1 is an example of this.

The major drawback of automated classifiers is that most of them work as a "black box", i.e. the interpreting person does not gain any information about how confident the algorithm is regarding class labels. Some algorithms allow such sort of extended information to be extracted (e.g. ANN) but most do not and therefore this information was not presented as this work deals with comparing the algorithms. The problem remains though and it has to be clearly stated that the algorithms presented are not suitable to extract such additional information. The class labels are rather an "all-or-nothing" result that conceals other potentially interesting findings such as how (un)ambiguous the class decision was. Likewise, none of the algorithms allows for an insight in terms of the actual differences between classes. No explicit statement can be made here for single curves, rather all input data are discriminated as a whole.

Coming back to the initial goal of this research on classification as formulated in Chapter 1, the question remains which supervised classification algorithm is best suited for the discrimination of stratigraphical sequences in shallow marine sediments in the Mediterranean Sea. This work shows that for the given problem (using wireline, MSCL and XRF logging data from marine sediments to determine glacial sedimentological units) non-linear classifiers work best. kNN, PNN and SVM (though technically a linear classifier in feature space) perform excellent even with few training samples. ANN and QDA yield equally good results when trained with sufficient training data. LDA and LGR are less appropriate because class boundaries are presumed to be non-linear (see Table 6.1).

Best Discriminating Curves

Another focus of this research is to identify those logging curves that are important for the automated classification process and those that prevent a good match. As previously presented, the Dipmeter mean micro-resistivity proved to be of utmost importance to the identification of stratigraphical units for both boreholes PRGL-1 (Figure 5.12) and PRAD-1 (Figure 5.22). In order to determine this to be a property

Good performance	Good performance when trained with ample data	Poor performance
kNN PNN SVM^l	ANN QDA	LDA^l LGR^l

Table 6.1: Suitability of classifiers when applied to geophysical borehole data of shallow marine sediments. ^{l} denotes linear algorithms. These results are only valid for the classification of stratigraphical sequences.

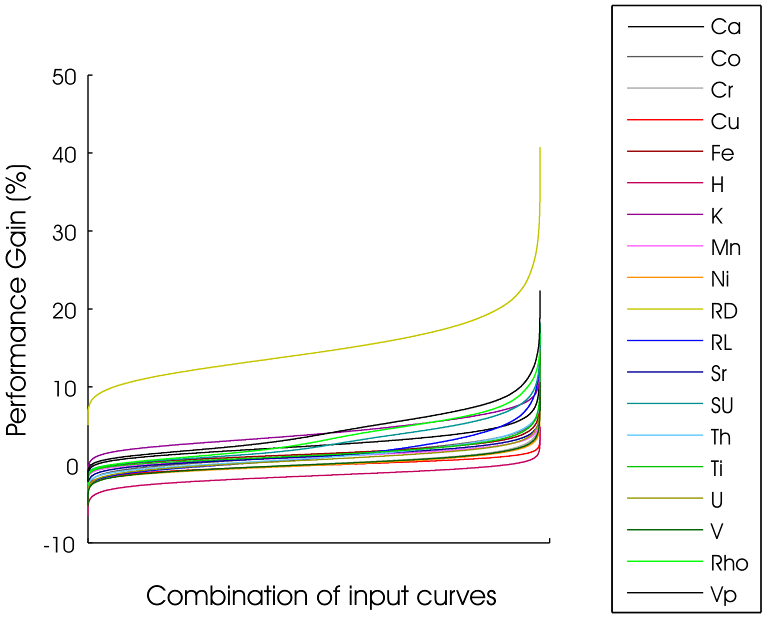


Figure 6.2: Performance gain when excluding and including each logging curve (sorted from left to right). Included are Laterolog resistivity (RL) and mean Dipmeter micro-resistivity (RD). Classification performance increases considerably when including the latter. Data from PRGL-1.

of resistivity or rather of micro-resistivity, another computation of all logging curve combinations at hole PRGL-1 was performed but this time adding the deep resistivity curve (RL) from the dual laterolog tool. The result is plotted in Figure 6.2 and shows that it is in fact only the micro-resistivity curve (RD) making such a difference. The excellent discrimination is not a property of resistivity itself. Rather, the deep resistivity is a good but not outperforming discriminator. The different behaviour can also be seen on a box-and-whisker plot of both curves divided by classes (Figure 6.3). While the micro-resistivities of the Dipmeter (DIP) show distinct medians that increase with depth (class 1 is on top of class 2 and so on), the dual laterolog's (DLL) deep resistivity medians and interquartile ranges are more overlapping, especially in classes 1–4. Formally, an one-way analyis of variance (one-way ANOVA) yields the same results

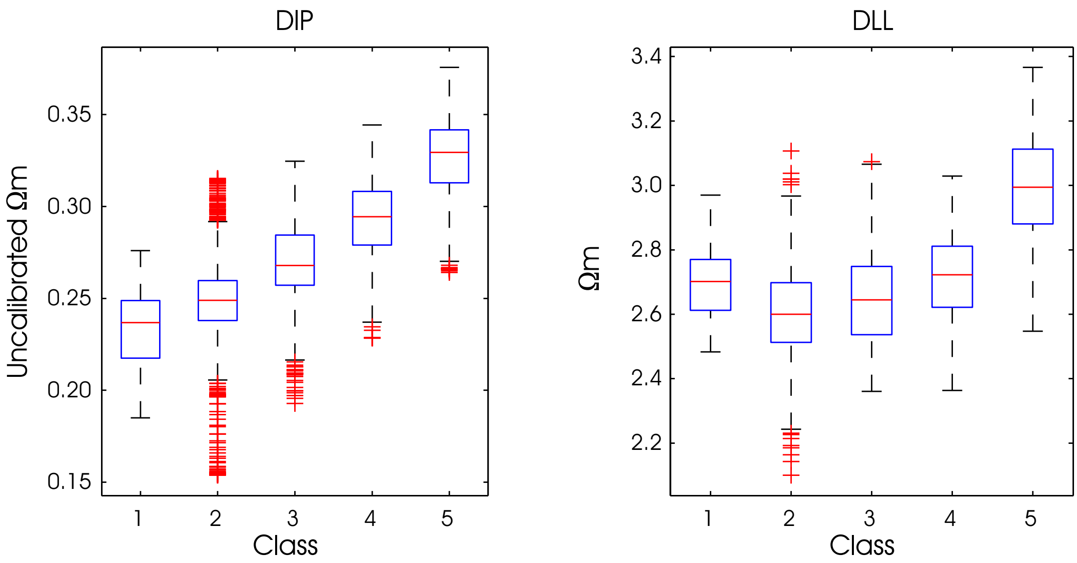


Figure 6.3: Distribution of resistivity logs by true classes. Mean Dipmeter micro-resistivities (left) show a different distribution than deep laterolog resistivities (right). Data from PRGL-1.

that are then statistically well-founded. The one-way ANOVA tests the data on the variability between groups and on variability within groups (Davis, 2002). It outputs a F statistic that is compared to a critical value F_c. If F is larger than this value, the result supports the alternate hypothesis that one or more of the samples are drawn from groups with different means. For borehole PRGL-1, F values are 809 (DIP) and 505 (DLL), respectively. F_c is 2.4 in both cases (assuming a 5 % level of significance), therefore both curves show significant different group means. Regarding borehole PRAD-1, no such investigation utilising the deep resistivity could be performed since this curve had not been recorded over the entire interval.

It seems that micro-resistivity is dependent on depth-related parameters such as compaction whereas resistivity is not. When looking at the original curves (Figure 6.4), micro-resistivity shows a wider range while the deep resistivity is more stable, especially at the top and the bottom part of the interval. The Dipmeter tool buttons that record the micro-resistivity at the sonde's four arms are pushed into the borehole wall hydraulically. As the name suggest, there is little depth of penetration. Only the bulk resistivity in the closest surrounding of the button is contributing to the output. Evidently, this lies within the invaded zone (see Figure 3.2) thus being exposed to drilling mud. Therefore the tool response is generally thought to be little if at all influenced by formation fluids (being replaced by the drilling mud) but reflects rock properties. This interpretation however has to be questioned in clay regimes like the ones at PROMESS-1 boreholes where effective porosity (from inter-

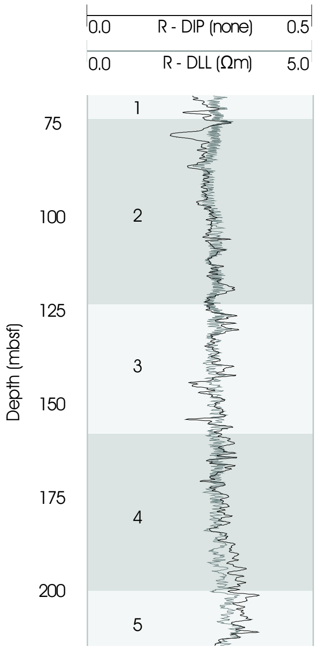


Figure 6.4: Resistivity logs of borehole PRGL-1. Mean Dipmeter micro-resistivities are plotted in black, deep laterolog resistivities are grey. Numbers (1–5) denote the stratigraphical units (classes).

connected pores) is extremely low. Here, varying resistivity values could be explained by changing fractions of pore fluids and hence pore space, i.e. by compaction. Although this is supported by increasing resistivities (less conductive pore fluids) with depth (more overburden pressure) it does not explain why the DLL deep resistivity does not respond in a similar way. Although the vertical resolution of DLL and Dipmeter resistivities are not the same (DLL is averaging over a few decimetres, Dipmeter records a value every half centimetre, see Appendix C), this discrepancy is considered negligible because the sequences are large scale sedimentary units. Concluding, the good discriminating properties of the micro-resistivity has to be due to some interaction of the (conductive) drilling mud with the formation that changes (distinctively) with each stratigraphic sequence. This may be indeed subtle changes in porosity (though at a generally low level) that are caused by compaction effects. Due to the generally low-permeable clays only fluids outside the equilibrated system (such as drilling mud) could make these changes visible, which would explain why these effects only occur within the close borehole vicinity (i.e. few millimetres). Each sequence was deposited during glacial times and covered with interglacial sediments. These were then exposed to rapid erosion (during which the compaction effect could not be completely removed) until the next glacial period supplied new sediments on top. Therefore the individual sequences could be distinguished by their different levels of compaction. Unfortunately, this scenario cannot be supported by porosity data either from downhole logs or cores. It should be mentioned though that the presented algorithms perform the classification task with good results only when multiple logging curves are used. Reducing this task to one of simply converting compaction effects into stratigraphical sequence discrimination would not seem to be adequate.

Because compaction has now been established as a major class discriminator for both boreholes PRAD-1 and PRGL-1, a remark should be made concerning the underlying parameter of compaction, which is depth (z). For all classification tasks, the spatial information of the data points (which is z) have been excluded. With the given situations discussed in this work, classes are a succession of units with depth. Mathematically, the class labels are a linearly ordered set with respect to z. Adding this parameter to the training data would yield excellent results but these would be solely based on z (see Figure 6.5a). This does not come as a surprise as the learning algorithms naturally detect this simple relationship and classify accordingly. It has to be stated though that the inclusion of this information is of no use to the interpreter as class boundaries are then solely dependent on the spatial location of training samples as one can clearly see in Figure 6.5b where the boundaries are set halfway between the nearest training samples. In the situation of PROMESS-1 data classes are dependent on their spatial position (and thus on z) because of their linearly ordered nature. Including this extra information to the training data set falsifies the classification process as it introduces a parameter that does not *always* represent a physical property. Here, z (or compaction) is a physical property *as well*, but because the two features (linearly ordered set *vs.* compaction) are indistinguishable, the depth information must be discarded. Also, in more general cases where classes are mixed (with respect to z) and less "sorted", the inclusion of depth information would not improve the classification result.

As for the other logging curves, they by and large cluster together. When comparing the two boreholes PRGL-1 and PRAD-1, the bulk density seems to be a reasonably good discriminator in both cases. This would support the compaction concept just described. Although this effect ought to be removed from the sonic logs (see Section 4.1), they still show good discrimination behaviour at least as far as hole PRGL-1 is concerned. On the low side, Copper downgrades the classification performance in both cases (PRGL-1 and PRAD-1). The reason for this can only be very speculative. It seems that the accumulation of this element within the clay material is either independent from the sedimentation process or even corresponds to an opposing pattern of unknown nature. Again, an ANOVA supports these findings. Table 6.2 shows significantly different distribution for each log curve and class combination for

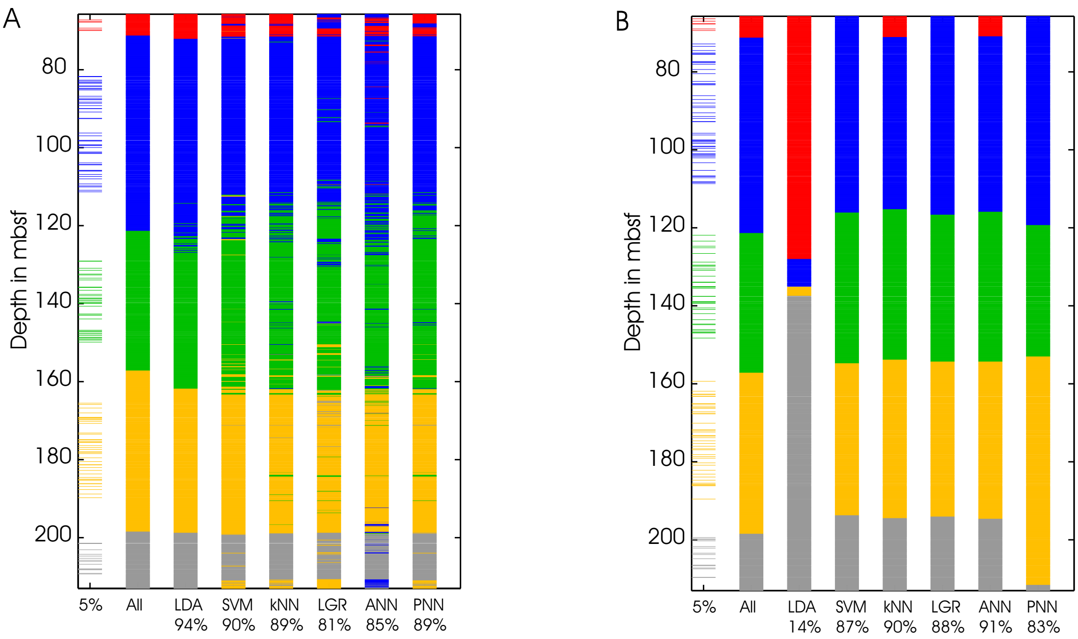


Figure 6.5: (A) Classification results when including depth information into the training data. Classification results dramatically improve as shown by the match figures. (B) Classification results when training data only consists of depth z. Class boundaries are dependent on the position of training data samples. LDA and PNN fail in this situation. All data from PRGL-1.

hole PRGL-1. While resistivity (RD) and Titanium (Ti) have different distributions for each class combination, Copper (Cu) shows the most similar distribution patterns (6 out of 10). As to why Titanium shows significantly different distributions in the ANOVA remains unsolved. Although the mean values are similar for classes 2 to 4, the variance is generally low. Thus, the ANOVA result suggests distinct distributions for each class boundaries but indeed such distinction only exists for classes 1 and 5 in this case.

The last questions that this work aims to answer is: can wireline measurements produce a sufficient data basis on which glacial-interglacial stratigraphical sequences can be distinguished? Evidently so, as both Tables 5.4 and 5.8 prove. Although the results presented there are only an example dependent on the respective training data set used, they show that core data are not strictly necessary to identify the stratigraphical units. This is a quite reassuring result because core data are often not available, be it due to technical or monetary difficulties. Remarkably, PROMESS-1 has been a reverse case. Data acquirable by wireline tools had to be replaced by core measurements because some of the wireline logging data were not recorded due to technical issues.

	Group										
Curve	1*vs.*2	1*vs.*3	1*vs.*4	1*vs.*5	2*vs.*3	2*vs.*4	2*vs.*5	3*vs.*4	3*vs.*5	4*vs.*5	-*s*
RD	+	+	+	+	+	+	+	+	+	+	0
Ti	+	+	+	+	+	+	+	+	+	+	
Ca	+	+	+	+	+	-	+	+	+	+	1
Co	+	+	+	+	-	+	+	+	+	+	
Fe	+	+	+	+	+	+	+	-	+	+	
H	+	+	+	+	-	+	+	+	+	+	
Rh	+	+	-	+	+	+	+	+	+	+	
Sr	+	+	+	+	+	+	+	-	+	+	
U	+	+	+	+	+	-	+	+	+	+	
V	+	+	+	+	+	-	+	+	+	+	
Vp	-	+	+	+	+	+	+	+	+	+	
Cr	+	+	+	-	+	+	+	-	+	+	2
Mn	+	+	+	+	+	-	+	-	+	+	
SU	+	+	+	+	+	+	+	-	-	-	3
K	-	-	-	+	+	+	+	-	+	+	4
Th	-	-	-	+	+	+	+	-	+	+	
Ni	-	+	-	+	+	-	-	+	+	-	5
Cu	-	+	-	+	-	-	+	-	-	+	6

Table 6.2: ANOVA results for all logging curves at PRGL-1. Each class is tested against each other (top row). + symbols denote significantly different distributions, - symbols denote the opposite (all with a 95 % level of confidence). The -s column sums all occurrences of non-different distributions for each logging curve.

Links to Existing Literature

Relating the findings of this work to previously published results by other authors is not a straightforward task. There are several new aspects that have not been published by others yet. Because there is no best classifier as such, the algorithms' performances can only be evaluated in the framework of shallow marine sediments, if not only shallow marine sediments *in the Mediterranean Sea*. In the existing literature, backpropagation neural networks are frequently applied to (wireline) logging data. Core scanning data has not been subjected to such learning algorithms at all. Also, there is little communication regarding the application of SVM or PNN to geophysical log data. "Simple" algorithms such as LDA or kNN seem not to be worthwhile to become published, which this work proves to be unjustified. Applying several classifiers to PROMESS-1 data is indeed a case study, but it is more than that. Existing and widely known algorithms as well as new methods have been compared to each other to reveal structural advantages of some algorithms over others —for the specific case of this data set. Side results include experience and recommendations regarding training set size, sample techniques, computing times, variance and parameters.

Conclusions

The comparison of several supervised learning algorithms applied to classify logging data from shallow marine sediments of the PROMESS-1 project yields the following results:

Among the tested algorithms, those that are both non-parametric and discriminating (SVM, kNN and PNN) are best suited to learn and classify stratigraphical units based on core and wireline log data. The three mentioned algorithms are superior in terms of absolute match levels (when comparing the classification output with true class labels), variance of results when using different training data sets and number of training samples required. ANN and QDA perform equally well if sufficient training data are available. The linear classifiers LDA and LGR cannot be recommended for this kind of learning task, as their performance levels are considerably lower. Furthermore, computing times of the LGR algorithm become prohibitively large when dealing with larger training data sets.

Supervised learning requires a training data set with known (and correct) class labels. The following properties of the training data set are found to be beneficial to the classification match in case of logging data:

All classes to be classified have to be adequately represented by the training data set. If the relative number of training samples of a class is very low compared to that of other classes, underfitting occurs and the result is corrupted. Also, the entire interval should be covered by the training data if possible. In large areas without training samples misclassifications are more likely to occur. In general, the more training data are available the better the classification match will be. The number of training samples is by no means the only parameter controlling the final performance, but may severely affect the result in extreme cases (i.e. when the number is very low).

Aside from that, the distribution of logging data is not important to the algorithms. Even in cases where a Gaussian distribution is formally required algorithms perform fairly well.

Bagging is an easy yet effective method to improve backpropagation neural networks that suffer from too few training data. Already good performing classifiers are not boosted much by this technique.

The mean Dipmeter micro-resistivity proves to be the best discriminating log curve. Results are boosted when this curve is part of the training data although the performance is not solely dependent on its inclusion in the data set. The reason for this are thought to be compaction effects involving very small porosity changes within the invaded zone around the borehole. In general, log curves acquirable by wireline tools perform nearly as good as a combination of all wireline, MSCL and XRF data.

Thus, in case where no core data are available, stratigraphic sequences can still be classified by exclusive means of wireline data.

Outlook

This work represents a thorough comparison of different classification algorithms applied to data acquired during the PROMESS-1 project. For this kind of logging data and classification goal, the best algorithms have now been determined. The question remains how specific the results are. The findings of this thesis are valid for wireline and core logging data recorded in the shallow marine regime of the two presented sedimentary systems in the Mediterranean Sea. Whether the classification of logging data from other geological settings and with different classification goals would yield similar results may be the focus of further research. Such goals could be permeability / porosity estimation, lithofacies discrimination or fracture detection.

As far as similar settings and goals are concerned, this work shows that in the absence of cores, stratigraphical sequences can still be accurately established by means of wireline data. Seismic data are generally capable to provide enough information to roughly identify sedimentary units and thus generate a training data set. Using the above mentioned non-linear discriminative algorithms, wireline log data can then be used to detect sequence boundaries with the high resolution typically provided by those measurements. The feasibility of such application has been proven in this work.

PROMESS-1 data were simple data sets in terms of their linearly ordered class labels with no interchanging class units. This makes the data appear depth-dependent although all presented algorithms are capable of classification without such strong dependency. Thus, the presented data do not properly show the comprehensive potential of these algorithms. Future work may demonstrate their feasibility when applied to more complex classification tasks.

APPENDIX

A. Logging Data (Gulf of Lion)

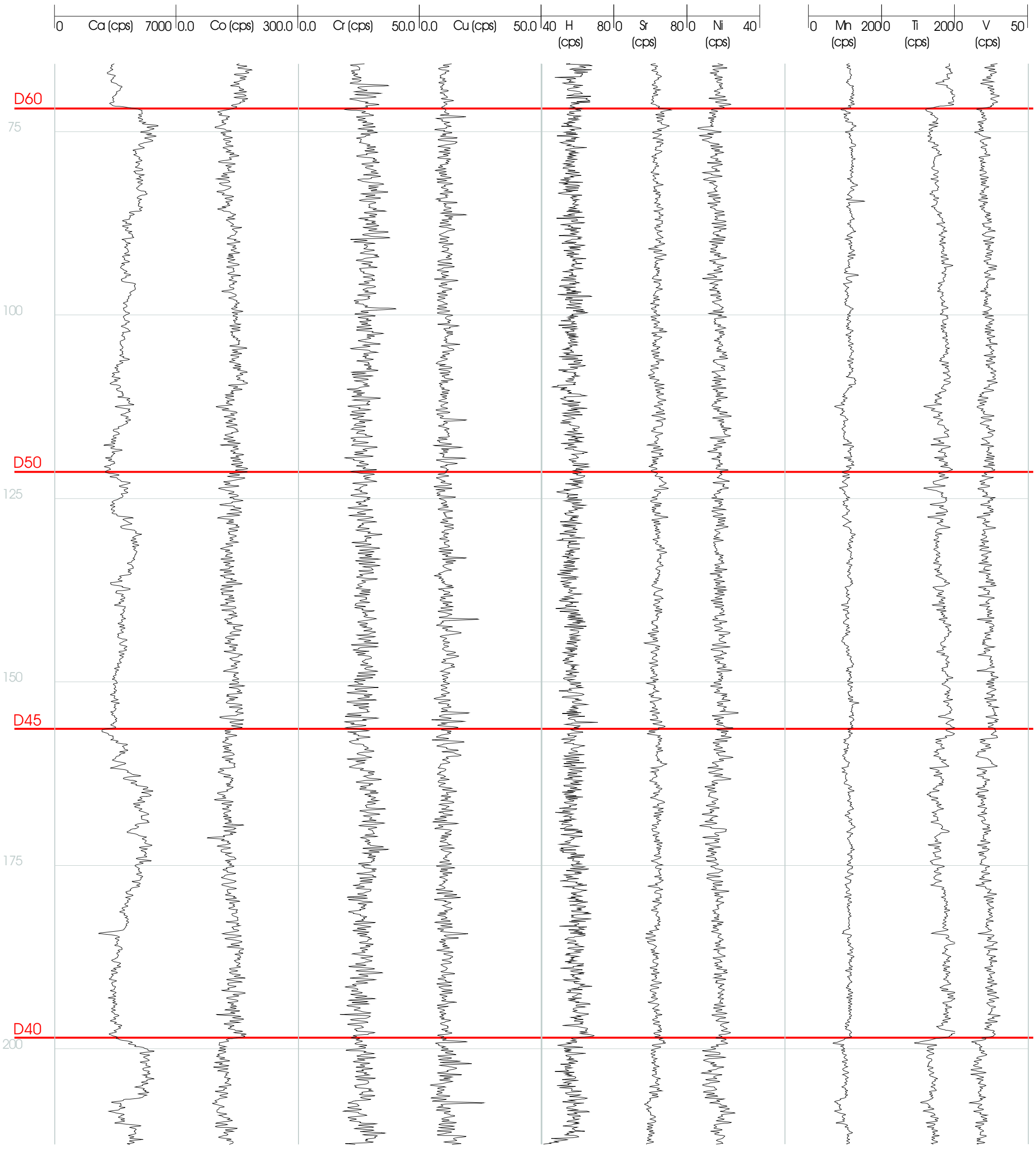


Figure A.1: Logging data of PRGL-1: Ca, Co, Cr, Cu, H, Sr, Ni, Mn, Ti and V. Sequence boundaries D40 to D60 are shown as used for classification. Depth in mbsf.

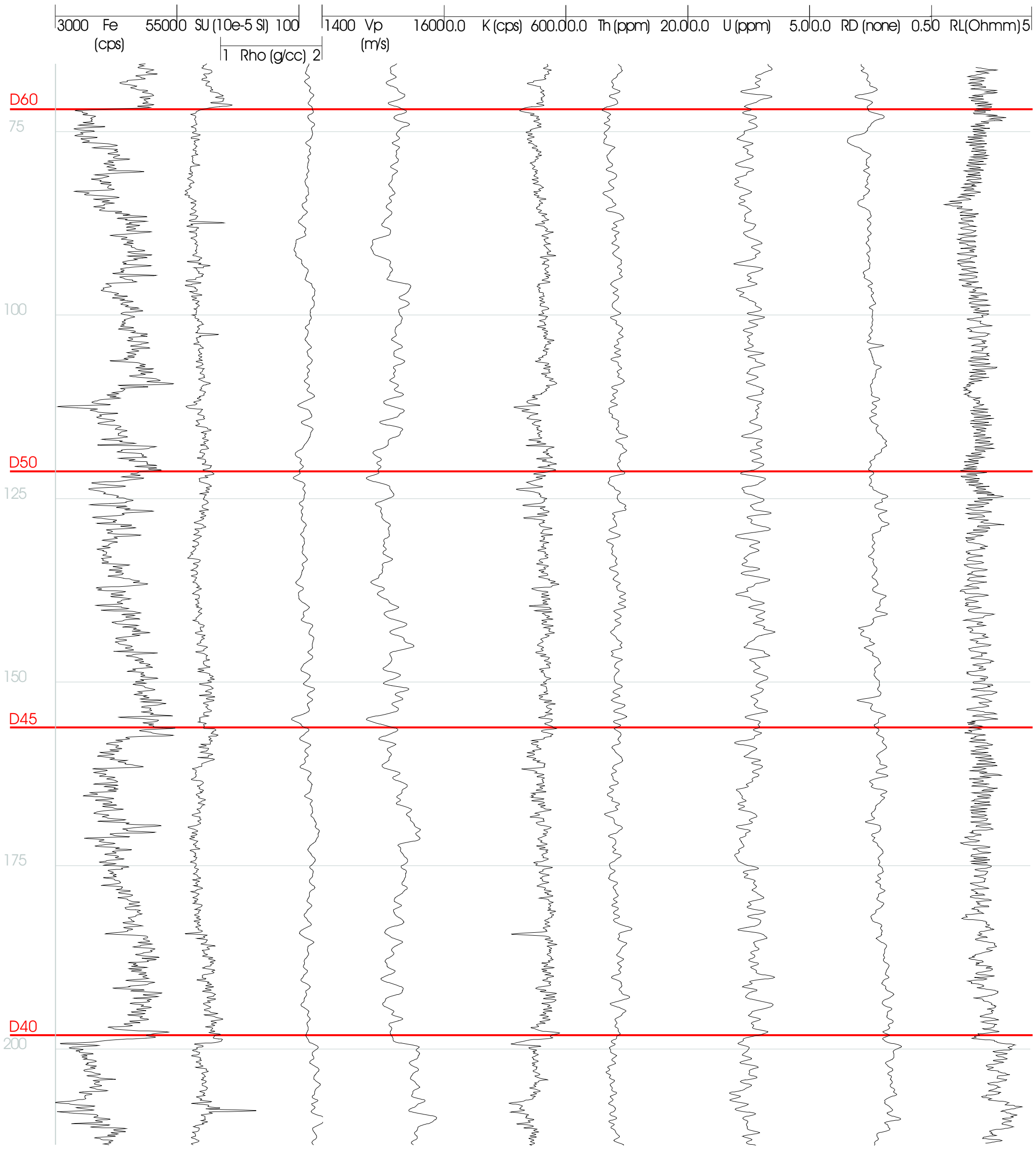


Figure A.2: Logging data of PRGL-1: Fe, susceptibility, density, sonic, K, Th, U and resistivity. Sequence boundaries D40 to D60 are shown as used for classification. Depth in mbsf.

B. Logging Data (Adriatic Sea)

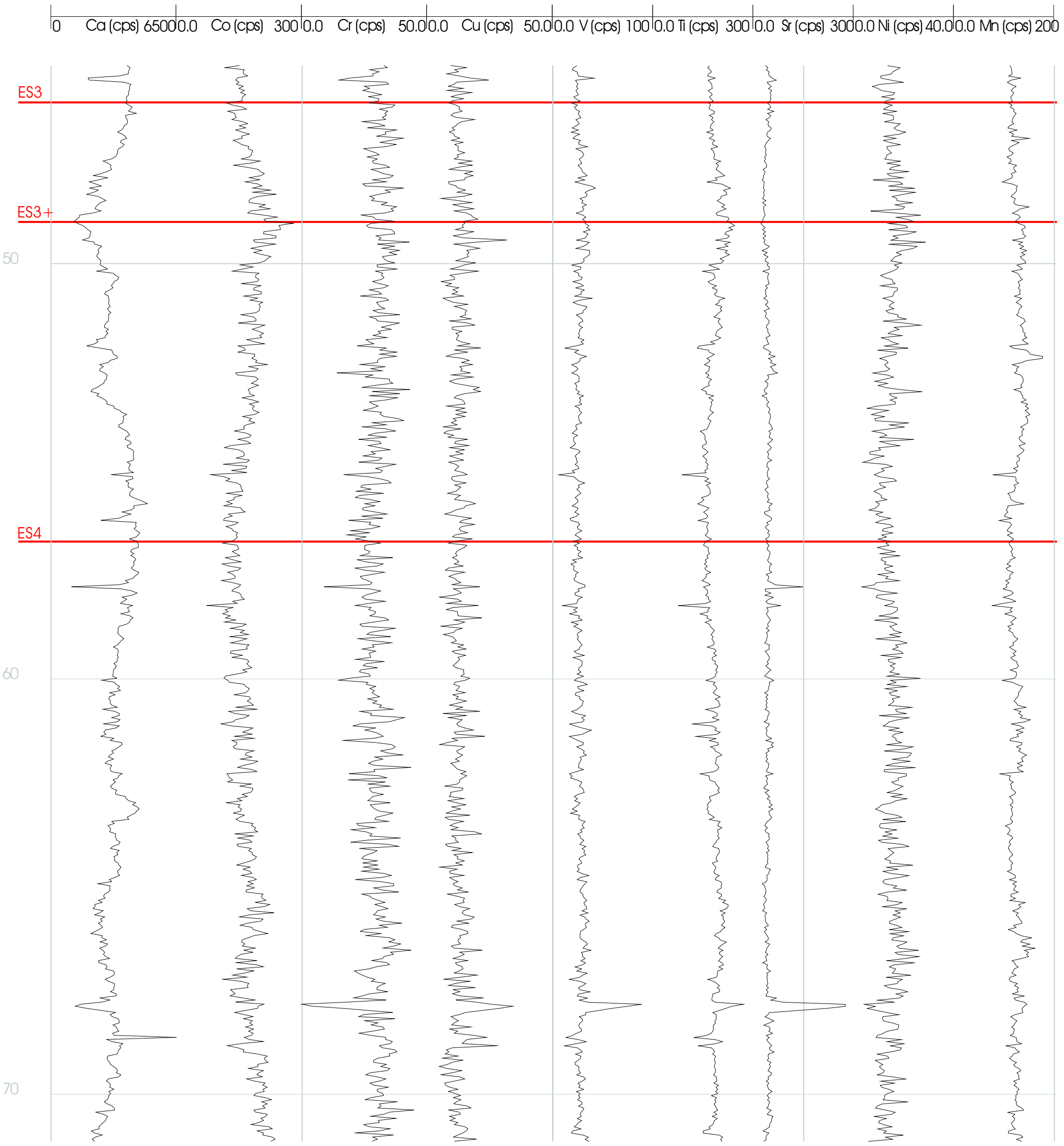


Figure B.1: Logging data of PRAD-1: Ca, Co, Cr, Cu, V, Ti, Sr, Ni and Mn. Class boundaries ES3 to ES4 are shown as used for classification. Depth in mbsf.

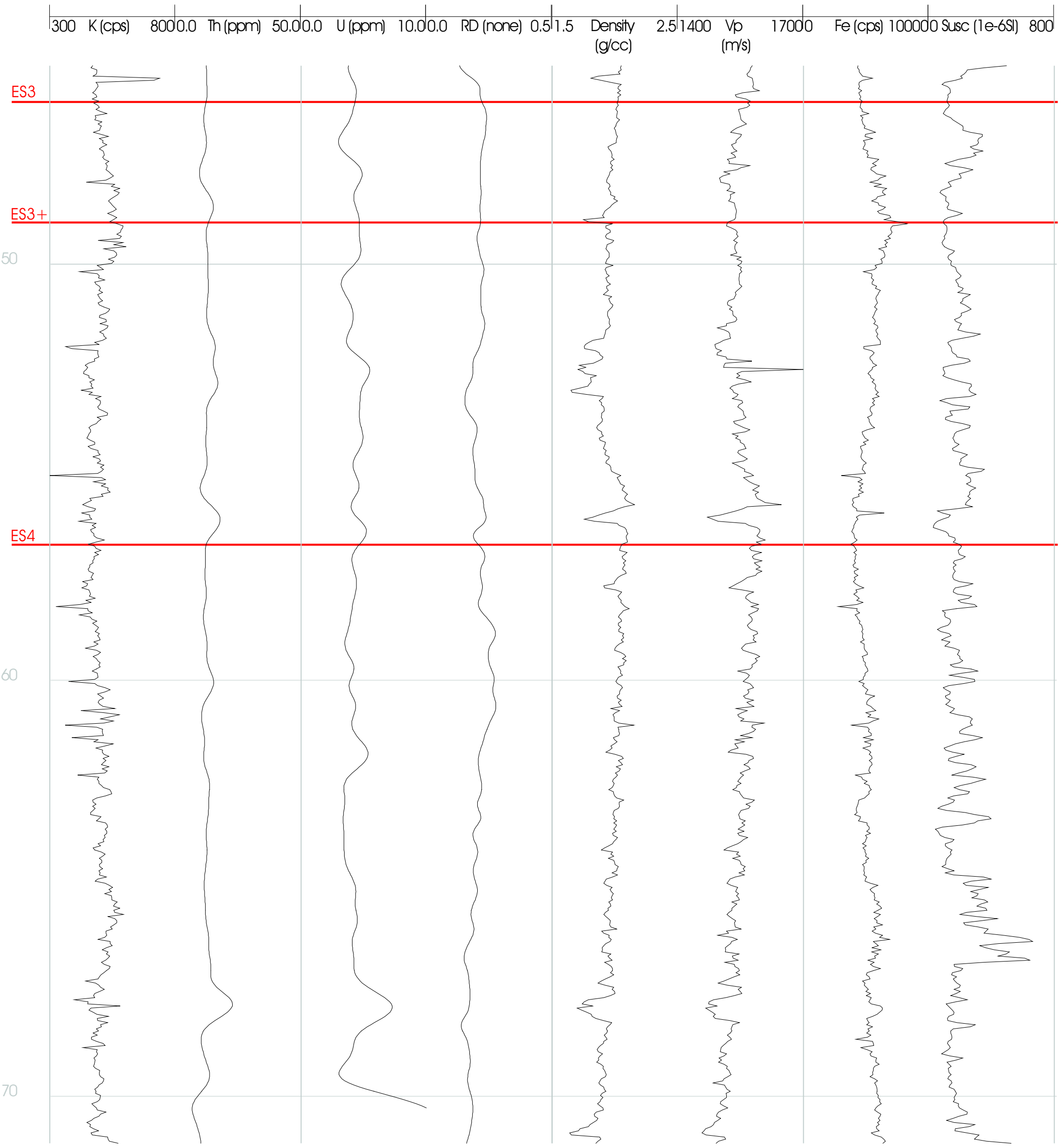


Figure B.2: Logging data of PRAD-1: K, Th, U, resistivity, density, sonic, Fe and susceptibility. Class boundaries ES3 to ES4 are shown as used for classification. Depth in mbsf. Th, U and resistivity curves have been smoothed.

C. Tool Parameters

Tool	Sample Rate (cm)	Penetration Depth (cm)	Logging Speed (m/min)
Caliper	5	0	10
Spectral Gamma-ray	10	15–20	3
Geochemical Sonde	10	5–10	1
Dipmeter	0.5	<1	5
Dual-laterolog	5	30–100	10
Micro-susceptibility	2	15–20	4
MSCL	1	Core diameter	N/A
XRF	2 or 4	Core diameter	N/A

Table C.3: Tool parameters for wireline sondes, MSCL system and XRF scanner.

D. Data Distribution (Gulf of Lion)

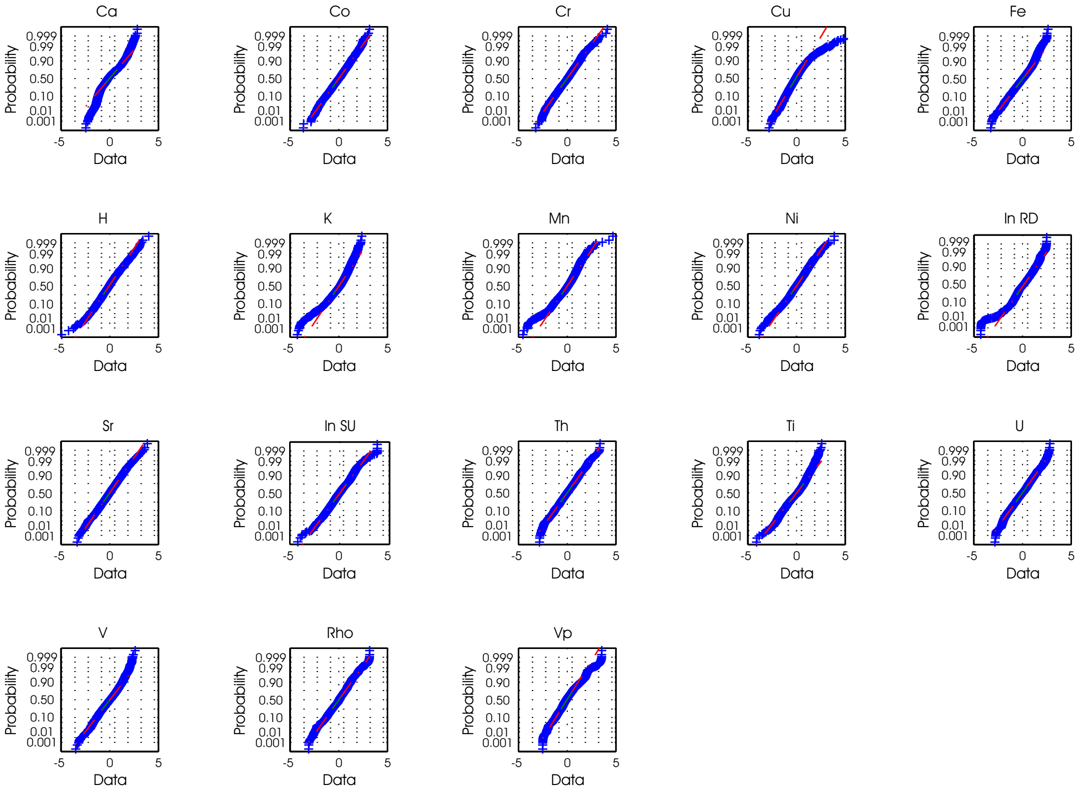


Figure D.1: Normal probability plots of the 18 logging curves at hole PRGL-1. Combined training and test data between 65 and 213 mbsf @ 5 cm sampling interval (2946 log readings) are plotted. Gaussian distributed data show as a straight line, curvatures indicate a different underlying probability density function. Ca = Calcium; Co = Cobalt; Cr = Chromium; Cu = Copper; Fe = Iron; H = Hydrogen; K = Potassium; Mn = Manganese; Ni = Nickel; RD = mean electrical micro-resistivity; Sr = Strontium; SU = magnetic susceptibility; Th = Thorium; Ti = Titanium; U = Uranium; V = Vanadium; Rho = density; V_P = sonic compressional velocity.

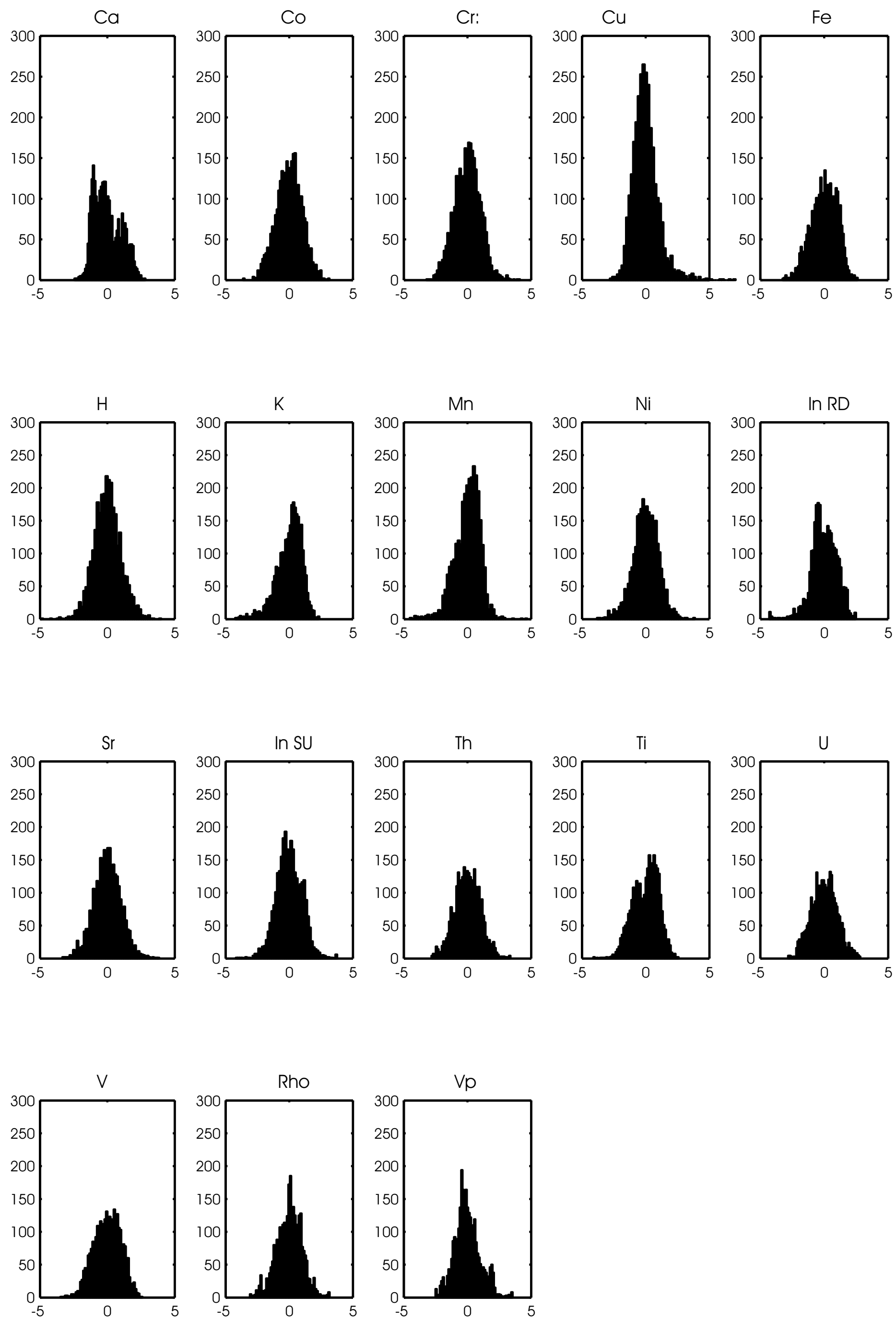


Figure D.2: Histograms of the 18 logging curves at hole PRGL-1. Combined training and test data between 65 and 213 mbsf @ 5 cm sampling interval (2946 log readings) are grouped into $\sqrt{2946} \approx 54$ bins. Ca = Calcium; Co = Cobalt; Cr = Chromium; Cu = Copper; Fe = Iron; H = Hydrogen; K = Potassium; Mn = Manganese; Ni = Nickel; RD = mean electrical micro-resistivity; Sr = Strontium; SU = magnetic susceptibility; Th = Thorium; Ti = Titanium; U = Uranium; V = Vanadium; Rho = density; V_P = sonic compressional velocity.

E. Data Distribution (Adriatic Sea)

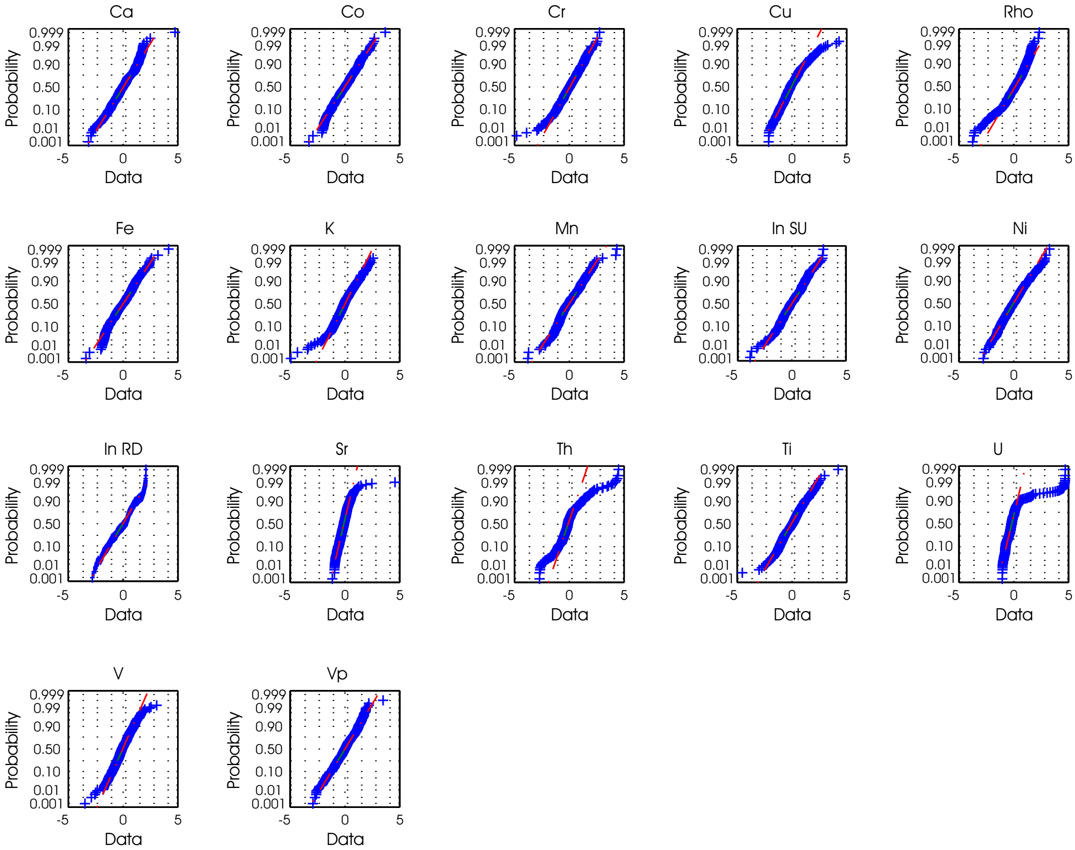


Figure E.1: Normal probability plots of the 17 logging curves at hole PRAD-1. Combined training and test data between 45 and 71 mbsf @ 5 cm sampling interval (519 log readings) are plotted. Gaussian distributed data show as a straight line, curvatures indicate a different underlying probability density function. Ca = Calcium; Co = Cobalt; Cr = Chromium; Cu = Copper; Rho = density; Fe = Iron; K = Potassium; Mn = Manganese; SU = magnetic susceptibility; Ni = Nickel; RD = mean electrical micro-resistivity; Sr = Strontium; Th = Thorium; Ti = Titanium; U = Uranium; V = Vanadium; V_P = sonic compressional velocity.

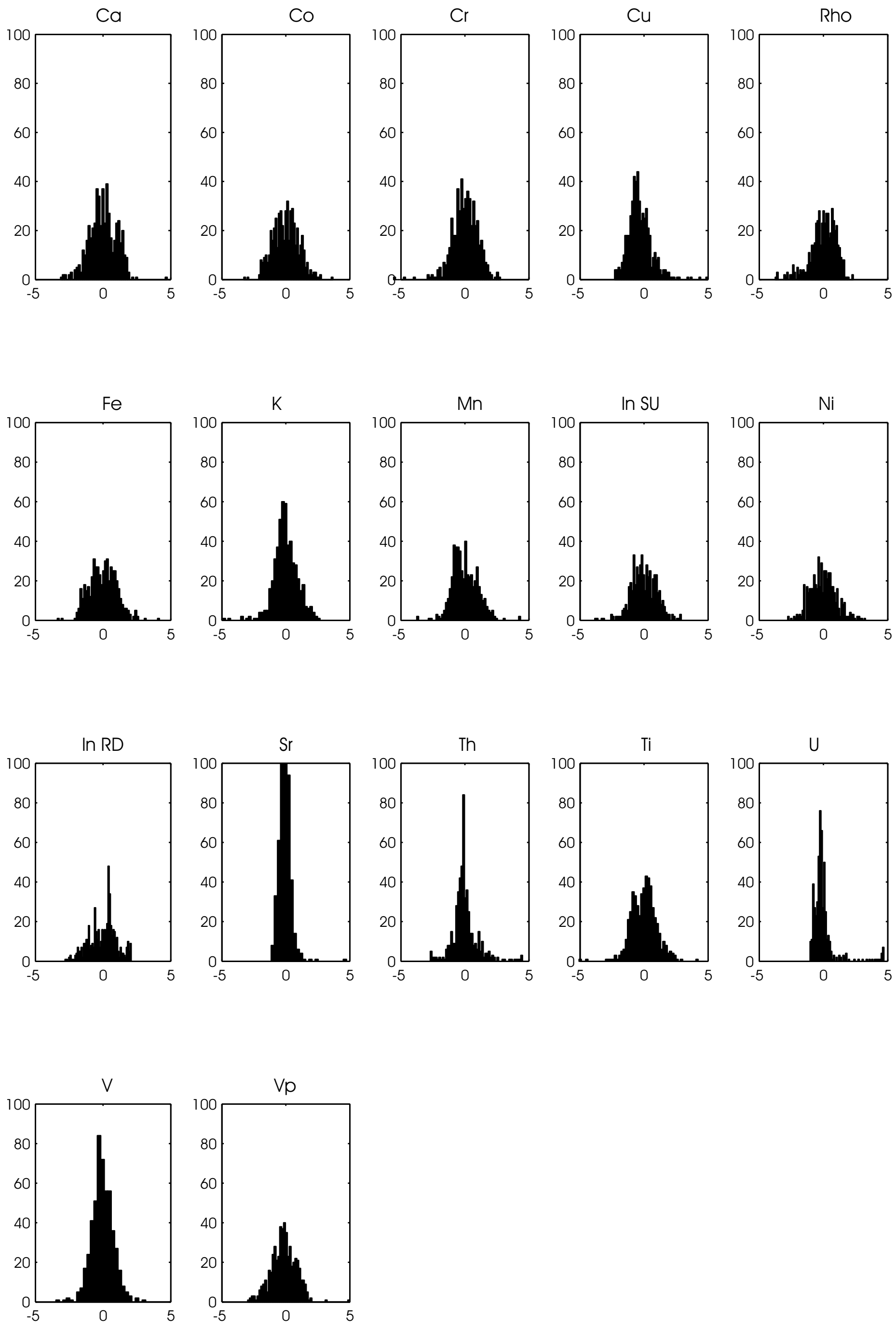


Figure E.2: Histograms of the 17 logging curves at hole PRAD-1. Combined training and test data between 45 and 71 mbsf @ 5 cm sampling interval (519 log readings) are grouped into 54 bins. Ca = Calcium; Co = Cobalt; Cr = Chromium; Cu = Copper; Rho = density; Fe = Iron; K = Potassium; Mn = Manganese; SU = magnetic susceptibility; Ni = Nickel; RD = mean electrical micro-resistivity; Sr = Strontium; Th = Thorium; Ti = Titanium; U = Uranium; V = Vanadium; V_P = sonic compressional velocity.

F. Classification Results with 50 % Training Data (Gulf of Lion)

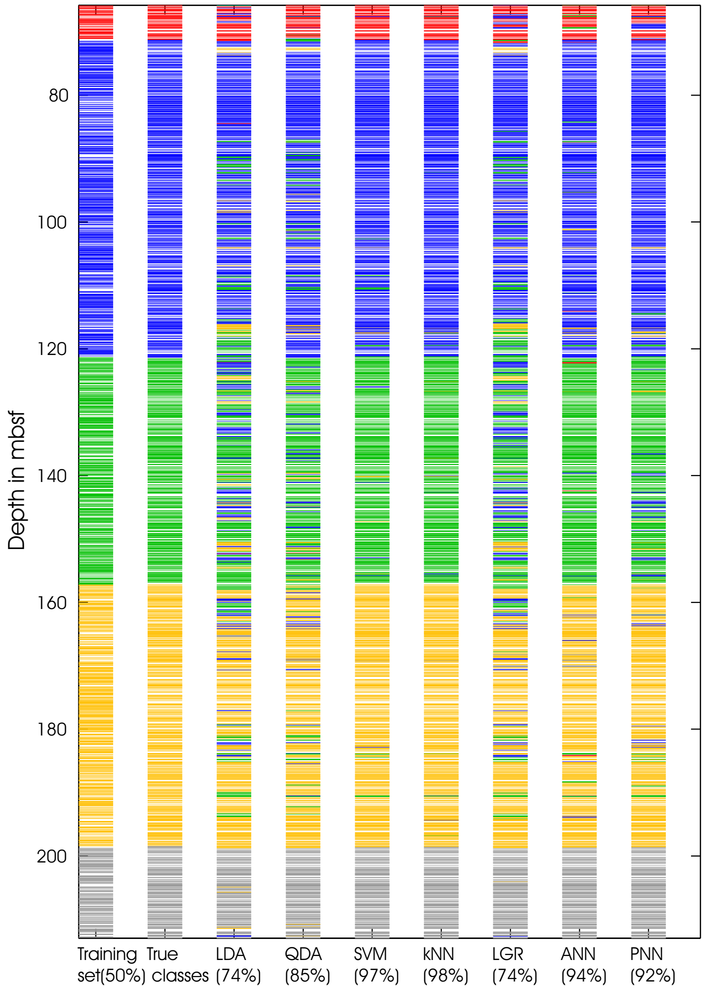


Figure F.1: Classification of hole PRGL-1. From left to right: Training data set (50 % of all available data per class); true class labels (sequence 1 = red, 2 = blue, 3 = green, 4 = orange, 5 = grey); linear discriminant analysis; quadratic discriminant analysis; support vector machine; k-nearest neighbour; logistic regression; artificial neural network; probabilistic neural network.

G. Classification Results with 50 % Training Data (Adriatic Sea)

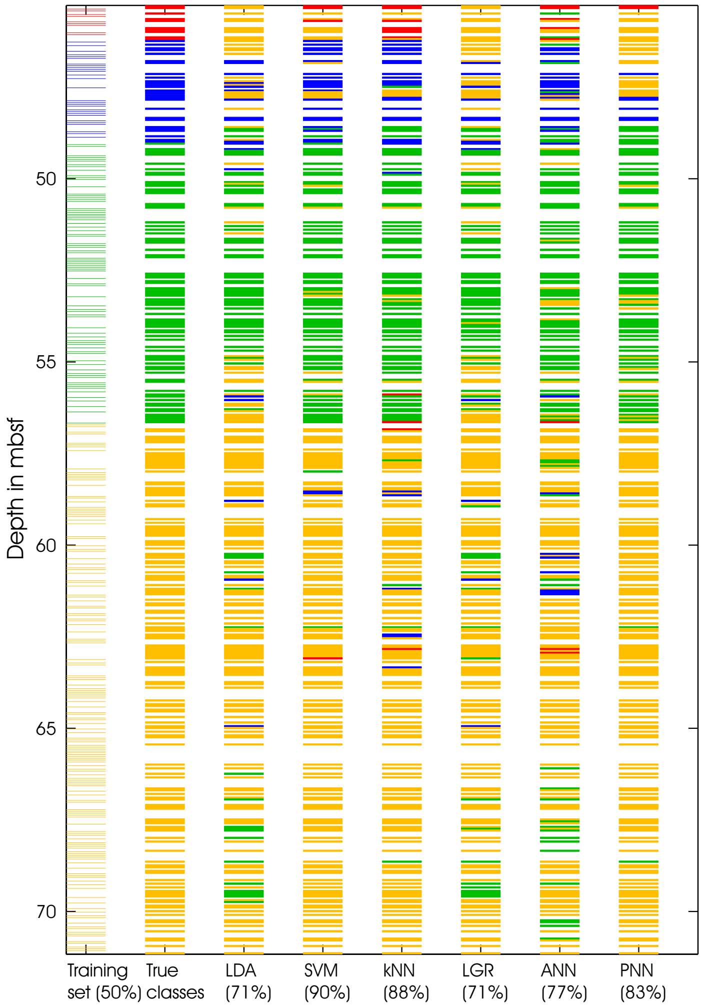


Figure G.1: Classification of hole PRAD-1. From left to right: Training data set (50 % of all available data per class); true class labels (sequence 1 = red, 2 = blue, 3 = green, 4 = orange, 5 = grey); linear discriminant analysis; quadratic discriminant analysis; support vector machine; k-nearest neighbour; logistic regression; artificial neural network; probabilistic neural network.

Bibliography

N.B.: All online references citing digital content within the world wide web are inherently non-permanent. When online links are cited, they refer to the respective online content as of June 1, 2006.

Allwein, E. L., Schapire, R. E., & Singer, Y., 2000. Reducing multiclass to binary: a unifying approach for margin classifiers, *Journal of Machine Learning Research*, **1**, 113–141.

Baldwin, J. L., Otte, D. N., & Wheatley, C. L., 1989. Computer emulation of human mental process: application of neural network simulations to problems in well log interpretation, *Society of Petroleum Engineers Paper 19619*, pp. 481–493.

Bellmann, R., 1961. *Adaptive control processes: a guided tour*, Princeton University Press.

Benaouda, D., Wadge, G., Whitmarsh, R. B., Rothwell, R. G., & MacLeod, C., 1999. Inferring the lithology of borehole rocks by applying neural network classifiers to downhole logs: an example from the Ocean Drilling Program, *Geophysical Journal International*, **136**, 447–491.

Berné, S., Satra, C., Aloïsi, J. C., Baztan, J., Dennielou, B., Droz, L., Dos Reis, A. T., Lofi, J., Méar, Y., & Rabineau, M., 2002. Carte morpho-bathymétrique du Golfe du Lion, notice explicative, *IFREMER, Brest, France*.

Berné, S., Rabineau, M., Flores, J. A., & Sierro, F. J., 2004. The impact of Quaternary global changes on strata formation: exploration of the shelf edge in the northwest Mediterranean Sea, *Oceanography, Strata formation on European margins: a tribute to EC-NA cooperation in marine geology, Special Publication*, **17**(4), 92–103.

Bhatt, A. & Helle, H. B., 2002a. Committee neural networks for porosity and permeability prediction from wireline logs, *Geophysical Prospecting*, **50**, 645–660.

Bhatt, A. & Helle, H. B., 2002b. Determination of facies from well logs using modular neural networks, *Petroleum Geoscience*, **8**, 217–228.

Bouchard, G. & Triggs, B., 2004. The trade-off between generative and discriminative classifiers, in *Proceedings of COMSTAT 2004 Symposium*, edited by J. Antoch, Physica Verlag.

Bücker, C. J., pers. comm. *RWE Dea, Hamburg, Germany*.

Bücker, C. J., Jarrard, R. D., Wonik, T., & Brink, J. D., 2000. Analysis of downhole logging data from CRP-2/2A, Victoria Land basin, Antarctica: a multivariate statistical approach, *Terra Antarctica*, **7**(3), 299–310.

Busch, J. M., Fortney, W. G., & Berry, L. N., 1987. Determination of lithology from well logs by statistical analysis, *SPE Formation Evaluation*, **2**, 412–418.

Cattaneo, A., Trincardi, F., Langone, L., Asioli, A., & Puig, P., 2004. Clinoform generation on Mediterranean margins, *Oceanography, Strata formation on European margins: a tribute to EC-NA cooperation in marine geology, Special Publication*, **17**(4), 105–117.

Caudhill, M., 1991. Neural network training tips and techniques, *AI Expert*, **6**(1), 56–61.

Cover, T. & Hart, P., 1967. Nearest neighbor pattern classification, in *Proceedings of IEEE Transactions on Information Theory*, pp. 21–27.

CSCF, 1984. Catalogue sédimentologique des côtes françaises: côtes de la Méditerranée de la frontière espagnole à la frontière italienne, in *Collections de la direction des études et recherches d'élecricité de France*, p. 290, Ministère des transports, Paris.

Davis, J. C., 2002. *Statistics and data analysis in geology*, John Wiley & Sons, 3rd edn.

Dennielou, B., pers. comm. *IFREMER, Géosciences Marines, Centre de Brest, France*.

Derek, H., Johns, R., & Pasternack, E., 1990. Comparative study of backpropagation neural network and statistical pattern recognition techniques in identifying sandstone lithofacies, *1990 Conference on Artificial Intelligence in Petroleum Exploration & Production*, pp. 41–49.

Dietterich, T. G. & Bakiri, G., 1995. Solving multiclass learning problems via error-correcting output codes, *Journal of Artificial Intelligence Research*, **2**, 263–286.

Droz, L. & Bellaiche, G., 1985. Rhône deep-sea fan: morphostructure and growth pattern, *American Association of Petroleum Geologists Bulletin*, **69**, 460–479.

Duda, R. O., Hart, P. E., & Stork, D. G., 2001. *Pattern Classification*, Wiley-Interscience, 2nd edn.

Fisher, R. A., 1936. The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, **7**(2), 179–188.

FitzGerald, E. M., Bean, C. J., & Reilly, R., 1999. Fracture-frequency prediction from borehole wireline logs using artificial neural networks, *Geophysical Prospecting*, **47**, 1031–1044.

Flores, J. A., pers. comm. *Geociencias Oceánicas, Universidad de Salamanca, Spain*.

Fort, G., 2005. Inference in logistic regression models (MATLAB codes). Software, *École Nationale Supérieure, France*.

Fort, G. & Lambert-Lacroix, S., 2005. Classification using partial least squares with penalized logistic regression, *Bioinformatics*, **21**(7), 1104–1111.

Fugro Engineers B.V., 2003. Geological report: geotechnical and operational data, scientific research on deltaic margins, Mediterranean and Adriatic Sea, unpublished field report, issue 2 N4373/01, PROMESS-1.

Hastie, T., Tibshirani, R., & Friedman, J., 2001. *The elements of statistical learning*, Springer Series in Statistics, Springer.

Hendriks, P. H. G. M., 2003. *In-depth Gamma-ray studies: Borehole measurements*, Ph.D. thesis, Rijksuniversiteit Groningen, Holland.

Hsu, C.-W. & Lin, C.-J., 2001. A comparison of methods for multi-class support vector machines, Tech. Rep. 19, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

Hsü, K. J., Cita, M. B., & Ryan, W. B. F., 1973. The origin of the Mediterranean evaporites, *Initial Reports of the Deep Sea Drilling Project*, **13**, 1203–1231.

Kecman, V., 2001. *Learning and soft computing: support vector machines, neural networks and fuzzy logic models*, MIT Press.

Kiefte, M., 1999. Discriminant Analysis Toolbox, version 0.3. Software, *Dalhousie University, Canada*.

LeCun, Y., 1985. A learning scheme for asymmetric threshold networks, in *Proceedings of Cognitiva 85*, pp. 599–604.

Lee, J. W., Lee, J. B., Park, M., & Song, S. H., 2005. An extensive comparison of recent classification tools applied to microarray data, *Computational Statistics and Data Analysis*, **48**, 869–885.

Lippmann, R. P., 1987. An introduction to computing with neural nets, *IEEE ASSP Magazine*, **4**(2), 4–22.

Liu, Y. & Sacchi, M. D., 2003. Propagation of borehole derived properties via a support vector machine (SVM), *CSEG Recorder*, **December 2003**, 54–58.

Lofi, J., Rabineau, M., Gorini, C., Berné, S., Clauzon, G., de Clarens, P., Tadeu Dos Reis, A., Mountain, G. S., Ryan, W. B. F., Steckler, M. S., & Fouchet, C., 2003. Plio-Quaternary prograding clinoform wedges of the western Gulf of Lion continental margin (NW Mediterranean) after the Messinian Salinity Crisis, *Marine Geology*, **198**(3-4), 289–317.

Michie, D., Spiegelhalter, D., & Taylor, C., 1994. *Machine learning, neural and statistical classification*, Ellis Horwood Series in Artificial Intelligence, Ellis Horwood.

Moore, A. W., 2001. Learning with maximum likelihood. Unpublished lecture slides, *Carnegie Mellon University, USA*.

Nittrouer, C. A., Miserocchi, S., & Trincardi, F., 2004. The PASTA project: investigation of Po and Apennine sediment transport and accumulation, *Oceanography, Strata formation on European margins: a tribute to EC-NA cooperation in marine geology, Special Publication*, **17**(4), 46–57.

Ori, G. G., Roveri, M., & Vannoni, F., 1986. Plio-pleistocene sedimentation in the Apenninic-Adriatic foredeep (Central Adriatic Sea, Italy), in *Foreland Basins*, edited by P. A. Allen & P. Homewood, vol. 8 Special Publications, pp. 183–198, International Association of Sedimentologists.

Pauc, H., 1970. *Contribution à l'étude dynamique et structurale des suspensions solides au large du Grand Rhône (Crau de Roustan)*, Thèse de 3ème cycle, Université de Toulouse Paul Sabatier, Toulouse.

Rabineau, M., 2001. *Un modèle géométrique et stratigraphique des séquences de dépôts quaternaires de la plateforme du Golfe du Lion: enregistrement des cycles glacioeustatiques de 100,000 ans*, Ph.D. thesis, Université de Rennes 1, Rennes.

Rabineau, M., Berné, S., Aslanian, D., Olivet, J., Jospeh, P., Guillocheau, F., Bourillet, J., Ledrezen, E., & Granjeon, D., 2005. Sedimentary sequences in the Gulf of Lion: a record of 100,000 years climatic cycles, *Marine and Petroleum Geology*, **22**, 775–804.

Rogers, S. J., Fang, J. H., Karr, C. L., & Stanley, D. A., 1992. Determination of lithology from well logs using a neural network, *The American Association of Petroleum Geologists Bulletin*, **76**(5), 731–739.

Rosenblatt, F., 1959. The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review*, **65**, 386–408.

Rothwell, R. G., Hoogakker, B., Thomson, J., & Croudace, I. W., 2005. Turbidite emplacement on the southern Balearic Abyssal Plain (W. Mediterranean Sea) during marine isotope stages 1-3; an application of XRF scanning of sediment cores in lithostratigraphic analysis, in *New ways of looking at sediment cores and core data*, edited by R. G. Rothwell, Geological Society, London.

Rubinstein, Y. D. & Hastie, T., 1997. Discriminative vs informative learning, in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, edited by D. Heckerman, H. Mannila, D. Pregibon, & R. Uthurusamy, pp. 49–53, AAAI Press.

Rumelhart, D. E. & McClelland, J. L., 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1: Foundations, MIT Press.

Schumann, A., 1995. Neural networks versus discriminant analysis: lithological classification of geophysical wireline logs, *Zentralblatt für Geologie und Paläontologie*, **1**(9), 843–852.

Schumann, A., 2002. Hidden Markov models for lithological well log classification, *Terra Nostra*, **4**, 373–378.

Serra, O., 1984. *Fundamentals of well-log interpretation*, vol. 1. The acquistion of logging data, Elsevier.

Trauth, M. H., 2003. Praktische Statistik und numerische Methoden. Unpublished handout, *Universität Potsdam, Germany*.

Trincardi, F. & Correggiari, A., 2000. Quaternary forced regression deposits in the adriatic basin and the record of composite sea-level cycles, *Sedimentary Responses to Forced Regressions. Geological Society, London, Special Publications*, **172**, 245–269.

Trincardi, F., Asioli, A., Cattaneo, A., Correggiari, A., & Langone, L., 1996. Stratigraphy of the late-Quaternary deposits in the Central Adriatic basin and the record of short-term climatic events., *Memorie dell' Istituto Italiano di Idrobiolgia*, **55**, 39–70.

Trincardi, F., Cattaneo, A., Correggiari, A., & Ridente, D., 2004. Evidence of soft sediment deformation, fluid escape, sediment failure and regional weak layers within the late Quaternary mud deposits of the Adriatic Sea, *Marine Geology, COSTA - Continental Slope Stability, Special Publication edited by Mienert, J.*, **213**(1-4), 91–119.

USGS, 2000. NLT Landsat7 images from U.S. Geological Survey, EROS data center, Sioux Falls, SD, USA, via World Wind (`http://worldwind.arc.nasa.gov/`).

Vapnik, V. N., 1979. *Estimation of dependencies based on empirical data*, Nauka.

Vapnik, V. N., 1998. *Statistical learning theory*, Wiley.

Vapnik, V. N., 2000. *The nature of statistical learning theory*, Information Science and Statistics, Springer, 2nd edn.

Vapnik, V. N. & Chervonenkis, A. J., 1968. On the uniform convergence of relative frequencies of events to their probabilities, *Doklady Akademii Nauk USSR*, **181**(4).

Vapnik, V. N. & Chervonenkis, A. J., 1989. The necessary and sufficient conditions for consistency of the method of empirical risk minimization, in *Yearbook of the Academy of Sciences of the USSR in Recognition, Classification and Forecasting*, no. 2, pp. 217–249, Nauka.

Werbos, P. J., 1974. *Beyond regression: new tools for prediction and analysis in the behavioral sciences*, Ph.D. thesis, Harvard University, Cambridge, MA.

Weston, J., Elisseeff, A., Bakir, G., & Sinz, F., 2005. The Spider, version 1.6. Software, *Max Planck Institute for Biological Cybernetics, Germany*.

White, J., pers. comm. *Schlumberger Oilfield UK Plc*.

Wiener, J. M., Rogers, J. A., Rogers, J. R., & Moll, R. F., 1991. Predicting carbonate permeabilities from wireline logs using a back-propagation neural network, in *Expanded abstracts of the technical program with author's biographies*, vol. 1, pp. 285–288, Society of Exploration Geophysicists.

Wolpert, D. H., 1995. The relation between PAC, the statistical physics framework, the Bayesian framework, and the VC framework, in *The Mathematics of Generalization*, edited by D. H. Wolpert, Addison-Wesley.

Wong, K. W., Ong, Y. S., Gedeon, T. D., & Fung, C. C., 2003. Intelligent well log data analysis for reservoir characterization, in *Proceedings of the Fourth International Conference on Intelligent Technologies (Intech'03)*, pp. 203–208, Chiang Mai, Thailand.

Wong, P. M., Gedeon, T. D., & Taggart, I. J., 1995. An improved technique in porosity prediction: a neural network approach, *IEEE Transactions on Geosciences and Remote Sensing*, **33**(4), 971–980.

Zhu, J. & Hastie, T., 2004. Classification of gene microarrays by penalized logistic regression, *Biostatistics*, **5**(3), 427–443.

This thesis was set with $\LaTeX\,2_\varepsilon$ provided by the MiKTeX 2.4 distribution. The KOMA class "scrreprt" was used along with the following packages: inputenc, fontenc, scrpage2, color, array, colortbl, setspace, newcent, charter, graphicx, natbib, amsfonts, bm, dsfont, textcomp, units, makeidx, footmisc and hyperref. The bibliography style is "gji".

Index

Activation function, 50
Adriatic Sea, 22
ANOVA, 101, 104

Bagging, 78, 91, 98
Bayes formula, 39
Bias, 59
Black box, 100
Bootstrap aggregation, *see* Bagging
Box-and-whisker plot, 77

Cable slack, 30
Cable slippage, 30
Cable stretch, 29
Calliper, 32
Chi-square test, 67, 83
Classification, 14
Classifier
  component, 58
  discriminative, 40, 97
  generative, 40
  informative, 40
  linear, 41
  non-linear, 41, 98
  non-parametric, 40, 97
  parametric, 41
Communalities, 38
Corer
  Fugro, 26
  PISTON, 26
  WIP, 26
Cross-plot, 16
Curse of dimensionality, 37, 62

Depth control
  core, 26
  logs, 29
Detrital regime, 63, 81
Dimensionality reduction, 37
Dipmeter, 33
Discriminant analysis
  linear, 16, 42, 69, 85
  quadratic, 43, 69, 85
Drilling mud, 30

Eigenvalue, 66, 83
ENCOR processing, 32
Erosional surface, 63, 81
Error
  generalisation, 59
  training, 50, 51, 59

Factor analysis, 37, 66, 83, 94
Factor loadings, 38

Gaussian distribution, 67, 84
Generalisation, 15
Geochemical Tool, 33
Gradient descent, 51, 98
Gulf of Lion, 18

Hidden neuron, 71, 87

Histogram, 67, 83

k-nearest neighbour algorithm, 16, 48, 69, 86
Kernel, 46, 70, 86
  Gaussian product, 46
Kernel function, 46
Kolmogorov's theorem, 50
Kolmogorov-Smirnov test, 67, 83

Laterolog, 33
Law of large numbers, 54
Layer
  hidden, 49
  input, 49
  output, 49
Learning, 15
  supervised, 15
  unsupervised, 15
Log odds ratio, 45
Logistic Model, 44
Logistic regression, 16, 70, 86
Logit transformation, 44
Loss function, 39

Maximum-likelihood estimation, 41
Mediterranean Sea, 18
Messinian Salinity Crisis, 18
Micro-resistivity, 33, 101
Micro-Susceptibility
  log, 34
  MSCL, 27
Mode, 73
Model complexity, 59
Multi-Sensor Core Logging, 27

Nearest neighbour rule, 48, 69, 86
Neural network, 16
  backpropagation, 70, 87
  probabilistic, 71, 87
Neuron, 49
Normal probability plot, 67, 83

Ockham's razor, xiii, 16
Off-training data, 68
Overfitting, 15

Parameter vector, 41
Parzen window, 16, 46, 52, 71
Pelagic regime, 63, 81
Perceptron, 49
Performance cross matrix, 73, 87, 99
Po, 22
Probabilistic neural network, 47, 52
Probability
  posterior, 39
  prior, 38
Probability density function
  class-conditional, 38
PROMESS-1 project, 23

Radial basis function (RBF), 70, 86
Reference Datum, 30
Regression, 14
Rhône, 18
Risk
  Bayes, 39
  conditional, 39
  empirical, 53
  expected, 53

Saw-tooth pattern, 65
Sequence boundary, 63, 82
Smoothing, 73
Spectral Gamma-Ray, 32
Stop error condition, 71, 87
Supervised learning, 68
Support vector machine, 16, 70, 86

Synapse, 49

Test data, 68, 84
Tool drag, 30
Topology, 49, 71, 87
Training data, 68, 84
Transformation function, 44

Underfitting, 100

Variance, 66, 83, 98
    Analysis of, *see* ANOVA

Weight, 49
Window function, 46
Window width, 46

X-Ray Fluorescence (XRF), 29

Curriculum Vitae

Name: Ralf Wilfried Gelfort
Date of Birth: August 27, 1974
Place of Birth: Karlsruhe, Germany
Nationality: German
Marital Status: Single

Professional Record:

since 2006	Technical advisor for Baker Hughes INTEQ GmbH, Celle, Germany. Member of LWD technology transfer group.
2003 – 2006	Research assistant at GGA Institut, Hannover, Germany. Principal member of borehole geophysics research group. Contribution to EC project PROMESS-1.
June/July 2005	Wireline logging consultant for Antares Datensysteme GmbH, Stuhr, Germany. Training and tool presentions for Weatherford Bin Hamoodah, Abu Dhabi, UAE.
2000 – 2002	Senior field engineer for Schlumberger Oilfield Services, Kazakhstan. Specialist for open and cased hole logging. Lead engineer for Maersk Oil Kazakhstan project.
1999	IT consultant for "Gabinet Tècnic d'Innovació Educativa" (gtie), Barcelona, Spain.

Education:

1997 – 1998	M.Sc. in Marine Geotechnics at University of Wales, Bangor, UK. Master thesis "Adapted Processing of Seismic Reflection Data Recorded with a Marine Deep-Towed System".
May – August 1998	ERASMUS "Mercator" exchange at Renard Centre of Marine Geology (RCMG), University of Ghent, Belgium.
1994, 1996 – 1997	Studies in geophysics at University of Kiel, Germany. "Vor-Diplom" in 1996.
1988 – 1993	Secondary school "Friedrich Wilhelm Gymnasium", Cologne, Germany. "Abitur" in 1993.