UNIVERSITY OF OSLO
Faculty of Mathematics and Natural Sciences, Department of Informatics

Prognostics from adaptive spatial entropy in early ovarian cancer cell nuclei

Master of Science in Informatics, field of study Image analysis

Andreas Kleppe

May 2011

Preface

This thesis has been submitted to the Faculty of Mathematics and Natural Sciences at the University of Oslo in partial fulfilment of the requirements for the degree *Master of Science* in *Informatics*. The study was started in August 2010 and completed in May 2011 and was carried out at the Institute for Medical Informatics at The Norwegian Radium Hospital and the Department of Informatics at the University of Oslo.

Acknowledgements

I would like to thank my supervisor Professor Fritz Albregtsen for his contagious commitment, eminent advises and sincere interest in my work, and also for introducing me to the field of biomedical image analysis. My thanks and great appreciation also go to my co-supervisors Professor Håvard E. Danielsen and Professor Anne H. Schistad Solberg for their comments, fruitful discussions and teachings.

I would like to thank Dr. Birgitte Nielsen for skillful technical assistance and discussions. The technical assistance of Cand.Scient. Tarjei Sveinsgjerd Hveem and Dr. Wanja Kildal are also appreciated, the same is the social welcomeness of all employees at the Institute for Medical Informatics.

I wish to thank my family for their great support and love. I especially thank my wife Kine for her patience and understanding. I also thank my brother Dag Otto and friend Charlotte for their comments to the biomedical introduction.

Sources and citation

Citation are only included when the source is more or less directly used to write the corresponding sentence or paragraph, or to refer to a recommended or the original text for further reading. Indirect use of supervisor's or any other person's contributions of information is not cited, though much appreciated. The same goes for information obtained in connection with previous completed courses, as long as this information is not revisited in order to produce this thesis. Though normally left uncommented, such limited use, or even a more restricted use, of citations is very common, the main reason being that it is too comprehensive to keep track of the source of each piece of information obtained in all informal and unrelated situations.

When citing books, articles or other papers with page numbers, page numbers of all used pages in the corresponding sentence or paragraph are included

for easier verification of the statements and targeted further reading on the subject. Page numbers are also included when referring to a source for further reading alone, though the number of pages in these citations are commonly much larger. In the special case when all pages are used or referred to, no page numbers are included.

The inclusion of page numbers has several other advantages, e.g. making it easier for the author to come back to his/her line of thought and limits an uncritical use of sources. The use of page numbers is uncommon in science and is possibly a growing problem, although some citing guides recommend or require its use [23, pp.99,103–106].

Assumptions about the reader

The target reader of this thesis is a computer scientist familiar with image analysis and at least fundamental knowledge in probability theory, statistical methods and linear algebra, or in essence, myself prior to working on this study. Only general basic knowledge is assumed in the research fields connected to this study, biology and medicine, including biomedicine. Theory and results that are assumed known are in general not discussed, but are included whenever it is natural to do so in order to provide a complete presentation of the subjects.

Notation

When working across different disciplines, it may be impossible to satisfy even the basic notational conventions of all disciplines. Addition challenges arise when appealing to different geographical location. It is thus reasonable to comment on some of the basic notational conventions that will be used in this study:

- In general, the notational conventions of the originating discipline will be used, e.g. if some image analysis theory is based on statistics, which again is partly founded on mathematics, then the mathematical conventions have precedence over the statistical conventions which in turn will have precedence over the image analysis conventions. Commonly, the originating disciplines have no conventional notation for the exact situation, so collisions and the use of the precedence rules are rare. One particular case in which the use of the notational conventions from an originating discipline may be found disturbing to some readers, is the use of conventional statistical notation, e.g. the use of P as the probability marker, p as a probability mass function (pmf) (also called a frequency function) and f as a probability density function (pdf) [11, pp.56,99,156; 57, pp.4,36,47]. When a probability function (pf) can be either a pmf or a pdf, the term probability function or pf is used with the notation f. Such conventional statistical notation is not always used in image analysis theory, e.g. see for instance the use of P as the pmf and p as a pdf or pf in chapter 2 of the standard textbook in image analysis by Duda et al. in [13]. It could also be explicitly noted that all estimators are written with capital letters and its estimates are written in its lower case equivalent.

- All matrices and vectors are written with square brackets, $[\dots]$, but parentheses and intermediate commas may be used to write a column vector on

a single line, e.g. we have that:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = (a_1, a_2, a_3, a_4) \neq \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix}$$

- I_n is the identity matrix of size nxn.
- To be unambiguous about whether the set of natural numbers includes zero or not, the notations $\mathbb{N}_0 = \{0, 1, 2, \dots\}$ and $\mathbb{N}_1 = \mathbb{N}_0 \backslash \{0\} = \{1, 2, 3, \dots\}$ are applied.
- The set of non-negative number is denoted as $\mathbb{R}^+ = [0, \infty)$.
- log will be used as the logarithm with an arbitrary base and $\ln = \log_e$.
- $\| \cdot \|$ will be used as an arbitrary norm and $\| \cdot \|_2$ as the Euclidean norm.
- $\lfloor \cdot \rfloor$ will be used as the flooring operator.
- Given a set K, $|K|$ denotes the *cardinality* of the set, i.e. the number of unique occurrences in the set [22, p.26].
- The *vast majority* will be used as at least 75 %.

Abstract

Providing a robust and reliable estimation of a patient's prognosis is necessary to make a qualified selection of the appropriate treatment for that patient. Digital image analysis of cancer cell nuclei is useful to make such estimation. In particular, texture analysis of the DNA organisation of nuclei has through a substantial number of studies proved to provide quantitative information of prognostic relevance.

Most previous studies have used the first, second or higher order statistics to estimate the prognosis, i.e. applied statistical texture analysis. We will in this study take a different approach where we attempt to exploit the internal structure of DNA-specific stained nuclei. In our novel approach, we apply a novel, refined adaptive segmentation method to extract small dark and bright structures within the nuclei, and estimate the spatial entropy of the dark or bright structures of each nucleus based on the area of the segmented objects. Finally, we will use the spatial entropies to obtain some very few, but powerful novel adaptive texture features by adaptively estimating the discrimination value of each spatial entropy using the combined knowledge of all relevant spatial entropies of all nuclei across a number patients.

We have analysed our novel approach on a dataset containing 134 patients with early ovarian cancer when using a proper evaluation method based on statistical bootstrapping. The results are very promising. Our method performs significantly better than the previously most promising method based on texture analysis. Moreover, it performs consistently at least about equally well as all other approaches based on image analysis. Combining the best feature of our novel approach with a single other feature, we also obtain the best performance among all approaches based on image analysis.

If selecting a subset of the dataset based on a set of predefined criteria unrelated to digital image analysis, our novel approach attains a correct classification rate of 84 %. This facilitate to a two-step recognition system. Again, our novel approach is consistently better, perhaps also significantly better, than all other approaches based on image analysis.

In conclusion, our novel approach seems to hold a promise of reliable estimation of the prognosis, which is necessary to make a qualified selection of the appropriate adjuvant treatment. Due to a very low dimensionality and the use of proper performance estimation, we expect that our approach will generalise well on an independent validation dataset. Moreover, because of the combination of high adaptivity in all stages of our approach and an addressed concern for the overfitting problem, we expect relatively good generalisation beyond the case under study. Nevertheless, caution must be called for, and new proper tests must as always be performed in the case of generalisations.

Contents

Chapter 1

Introduction

We will in this thesis study a dataset of digital images of DNA-specific stained nuclei captured from 134 patient with early ovarian cancer. Our general aim was to develop an automatic algorithm that reliably estimates the outcome, or *prognosis*, of novel patients with early ovarian cancer. Providing a robust and reliable estimation of a patient's prognosis is necessary to make a qualified selection of the appropriate treatment for that patient. By analysing the methodology behind such an estimation, an improved understanding of the biological processes involved in *carcinogenesis*, the development of cancer, may also be achieved.

More specifically, we will attempt to capture textural properties of the digital images in our dataset which are of prognostic value. Such analysis is related to the analysis of the DNA organisation of the nuclei. To perform this analysis, we will introduce a set of matrices which attempts to capture some specific contextual information of each segmentation class of a nucleus. From each of these matrices, a few features are adaptively extracted. The main aim has been to develop and evaluate the prognostic value of these class specific spatial texture features for early ovarian cancer.

We will begin this chapter with a brief biomedical introduction with focus on relevant theory for this thesis. Our attention will in particular be restricted to humans because these are the subjects in our dataset. The biomedical introduction will end with an informal justification of why we can even hope that the DNA organisation contains valuable prognostic information. We will then introduce the field of digital pathology in general, and conclude the chapter with the aim, strengths and limitations and organisation of the present study.

1.1 Biomedical background

1.1.1 The human cell

In biology, the cell is *the smallest structural unit of living matter that is able to function independently* [16]. Humans consist of multiple specialized cells organised into tissues and organs, making us a part of the class called *multicellular organisms*. The human cell is enclosed by a membrane and consists of specialized compartments called *organelles* which perform specific functions. The largest and most prominent organelle is called *nucleus*, meaning kernel.

The genetic information in humans is coded in the *deoxyribonucleic acid (DNA)*. Most of the human DNA is contained within the nucleus, but a small proportion (about 1 %) is contained in mitochondria. We will in this study use the term *genome* to refer to the genetic information encoded in the nucleic DNA.

DNA organisation

DNA is a double helix with backbones made of sugar-phosphate and bases on each helix oriented toward each other, forming a base pair which is connected by a hydrogen bond. These bases code the information of the DNA; adenine base (A) and thymine base (T) always form a pair, the same do the guanine (G) and cytosine base (C).

Chromatin is the complexes of DNA and proteins. It is possible to differentiate between several chromatin structures based on the level of DNA packaging. The lightest packed chromatin structure, often referred to as the *basic structural unit of chromatin*, is the *nucleosome*, a chromatin fibre of approximately 10 nm in diameter. A nucleosome is a segment of DNA wounded around the core of a *histone* protein. The nucleosomes form a 30 nm chromatin fibre, which again forms a third chromatin structure known as *DNA loops*, but the exact nature of both these structures are still a controversy, the same is the further winding and stacking of DNA loops into more condensed chromatin structures.

The human genome

The human DNA is about 1.8 metres long and is entirely contained within each cell, except the gamete (egg and sperm) cells, where each only carry half the genetic information of an individual. Less than 2 % of the genome consist of *protein-coding DNA*, stretches of DNA that each codes for a specific type of protein [52, p.1556]. We will call such DNA stretches for *genes* in this study. When a complete gene is stored in nucleosomes, it can be used to synthesise a copy of the stretch in the RNA coding scheme, a coding scheme injective to the DNA coding scheme. This copying process is known as *transcription* and the DNA stretch being copied is called *expressed*. The synthesised RNA, specifically a *messenger RNA (mRNA)*, is in turn used as a template for creating the specific protein coded in the mRNA, a process named *translation*.

Even though only less than 2 % of the genome consists of genes, studies have shown that about 80 % of the genome shows signs of being expressed at some point. Some DNA segments not coding for proteins are coding for RNA in itself, i.e. RNA is the end product and thus no translation is performed after the transcription. Previously researchers believed that only a small amount of DNA expression led to RNA as the end product, but it has more recently been found that about half of the synthesised RNA has the RNA in itself as the end product. Such end product or non-protein-coding RNA has been found to regulate the DNA expression - the expression of a particular RNA-coding DNA may give rise to either a suppression or an enhancement of the DNA expression of particular gene(s) [50, p.46]. [52, p.1556]

While the entire genome is contained within most cells, only a small portion of the genome is expressed in a single cell at any given time. The ability of cells to regulate and differentiate in the use of the genome makes them able to

specialise by performing differently based on e.g. cell type, location in the body and local and distant needs of the body.

1.1.2 Cancer

The function and growth of each cell is normally carefully regulated to meet local and distant needs of the human body. There exist an vast amount of mechanisms controlling the different processes taking place and correcting any fault - the entire system is so complex that it is nearly impossible for a cell to escape all the controls and survive in an escaped state as an abnormal cell would normally destroy itself, a process known as *apoptosis*. However, through a multistep complex process which may last more than half of the individuals life time, cells may escape the carefully controlled environment and form a tumour.

A tumour may either be *benign* or *malignant*. A benign tumour neither invades adjacent tissues nor *metastasises*, which is the spread of a disease to a non-adjacent organ or part, e.g. the spread of a tumour to a new tumour in a non-adjacent organ. A benign tumour may still cause the carrier complications as it may grow so large that it interferes with adjacent environments.

A malignant tumour is called *cancer* and is the family of diseases characterised by both uncontrolled growth and invasion into adjacent tissues. A tumour is classified as malignant if it invades adjacent tissues, which clearly separates them from the benign tumours which are self-limiting. Another commonly accepted property of malignant tumours is its the ability to change, e.g. adapt to its surroundings, gain new properties or loose old restrictive properties. It is also typically assumed to be capable of metastasis, but prior to metastasis it is still unknown whether a particular malignant tumour possesses this capability as it requires a whole series of fundamental changes in its cells. However, if left untreated, it is reasonably assumed that all malignant tumours will be able to metastasise at some point in the future and they thus pose a serious threat to the carrier.

Prognosis

A malignant tumour is diagnosed according to the cell type and tissue of origin, the extent of spread and other observations. However, given a set of diagnoses, there always exist some who relapse and others who do not. It is therefore interesting to supplement the diagnoses by attempt to estimate what separates the patients who relapse from the other patients with the same diagnosis. In some cases, such prognosis estimation could also be performed across different diagnoses.

Since the task in this study is prognosis estimation, it makes sense to point out why someone do relapse at all. There are mainly two reasons, one being that the treatment, e.g. the surgery, the chemotherapy, the radiotherapy or any combination of multiple treatments, failed to completely remove or permanently disable the cancer, and the other being that an undetected spread had occurred prior to or before the completion of the treatment.

1.1.3 DNA organisation and carcinogenesis

As mentioned in the introduction of this chapter, the analysis found later in this thesis is related to the analysis of the DNA organisation in the nuclei. The discussion in this subsection should not be seen as a part of a research text, it is merely intended to inform the reader of this thesis of why we can expect to find prognostic valuable information in the DNA organisation.

Oncogenes and suppressor genes

Much research in carcinogenesis has the last decades been focused on specific genes called *oncogenes* and *suppressor genes* or *anti-oncogenes*. The basic idea was that carcinogenesis was caused by multiple mutations in such genes.

A oncogene is a gene responsible for normal growth and differentiation of cells, but their erroneous expression may also cause abnormal cells that normally should have undergone apoptosis to survive and proliferate instead. As the name anti-oncogene indicated, a suppressor gene has approximately the opposite function as a oncogene, more precisely it may slow down the cell cycle and thus effectively decrease the cell division rate, and it may promote apoptosis. If a suppressor gene is not functioning as normal, e.g. due to mutation or erroneous lack of expression, it may substantiate to the development of tumours.

The oncogenes and suppression genes are closely related to the cell cycle and in particular the process of controlling the replication of the DNA. Because of this, the mutation of some such genes may both increase the cell division rate and increase the probability of other mutations, both due to fewer control mechanisms during replication, and cause proliferation despite incorrect replication. As an example, the most frequently mutated gene in human cancer, a suppressor gene called *TP53*, codes for a protein, called *p53*, which can (among other things) be compared to an 'emergency brake' that halts proliferation if conditions are not adequate for correct DNA replication [63, pp.231,233]. It is thus interesting to note that a mutation in this single gene is found in over 50 % of all human cancer tested for this mutation [63, p.232].

There has in many patients suffering from cancer not been located any mutation in known relevant oncogenes and suppressor genes. One can naturally assume that these cancers were caused by mutations in unknown oncogenes and suppressor genes, but this theory becomes less probable as gradually more genes get characterised.

The theory of oncogenes and suppressor genes does not include the expression of the DNA. This questions the generality of the theory, because it is the DNA expression that results in production of RNA and/or proteins, which in turn performs a wide variety of function, and thus both reflects and partially controls the function of the cell. Furthermore, we have already mentioned that RNA-coding DNA stretches are relevant to the DNA expression and thus the function of the cell, which indicates that it may in general not be sufficient to only study the mutations in and expression of genes. From these observations and the information that about 80 % of the genome show signs of being expressed at some point, we can conjunct the hypothesis that the theory of oncogenes and suppressor genes is a part of a larger picture involving the majority of the DNA and its expression.

DNA organisation in malignant tumours

In many patients suffering from cancer, a general abnormality in the DNA organisation can be observed, which may be the results of e.g. an increased amount of DNA content or different DNA expression. Moreover, if mutations have occurred in known oncogenes and suppressor genes, then one can typically also observe a general change in the DNA organisation. In particular, the majority of TP53-mutations result in the most common form of genomic instability known as *aneuploidy*, a ploidy type that will be discussed in section 3.1, which is associated with an increased amount of DNA content and thus a general change in the DNA organisation [58, p.293]. It seems thus reasonable that an analysis of the DNA organisation is likely to both capture the effect of multiple mutations in known oncogenes and suppressor genes and the limitations with this theory, e.g. mutations in unknown oncogenes and suppressor genes and the importance of DNA expression.

It is worth noting why a general change in the DNA organisation is a result of e.g. an increased amount of DNA content or different DNA expression. The main reason is the strict organisation of the DNA; a single, connected double helix of about 1.8 meters is required to fit in the nucleus with a diameter of about 6 micrometres. Of course, the change may be more or less dramatic with respect to the entire DNA organisation, but because the organisations is so strict it is likely that even a minor alteration, e.g. slightly more DNA or a slightly different DNA expression, would result in rather significant changes.

The analysis of DNA organisation can also be justified by using theory. We have already commented that the DNA and its expression reflects and partially controls the function of the cell. As the function of the cells in malignant tumours is abnormal, it is only natural to assume that these abnormalities are reflected in or based on changes in the DNA and its expression. Evidence of such relationship has also been shown in studies [8, p.45]. It is thus possible to view cancer as a disease of the DNA organisation. Some studies that have revealed relationship between DNA organisation and cancer have also proposed and made probable that the changes in DNA organisation is associated with carcinogenesis in itself [8, pp.39–41; 9, p.6].

1.2 Digital pathology

Pathology is the branch of medical science that studies the causes, nature and effects of diseases. A *pathologist* is a medical doctor who specialises in pathology. *Digital pathology* is the digital subdivision of pathology, which attempts to assist and automate, though not replace, the pathologist. An obvious way of assisting the pathologist is to let the pathologists subjective evaluation to be completely moved into the electronic domain. Such a transition will require the production of digital images with a resolution which is, for all practical purposes, equivalent to or better than the physical view the pathologist traditionally has evaluated. This field of digital pathology has received much clinical and commercial interest and is today to a large extent achieved. In particular, there exists today only a few special situation within cancer research where the technology is not sufficient for this transition to be performed. [45, p.90]

While enabling the pathologist to move into the electronic domain is an

important promise within digital pathology, it is far from the only way to assist the pathologist. Three other important promises within digital pathology have been recognised; *diagnostic*, *response prediction* and *prognosis*. All these three promises can assist the pathologist through automation of routine tasks needed or helpful when performing the subjective evaluation, through verification of the subjective evaluation, and to automatic, or in combination with the pathologists expertise, evaluate cases with subvisually essential attributes. [45, p.90]

1.2.1 Digital pathology in cancer research

In (digital) diagnostic, the aim is to automatically classify any desirable characteristics of a disease, which may include the type of disease, or in any way assist the pathologist in doing so. For instance, much effort within cancer research has been made to make a diagnose about some characteristic with, or the presence of, cancer [39, pp.4–6; 45, pp.139–144,146–149]. Going forth to the estimation of an unknown future, (digital) prognostics attempts to provide a reliable estimate of the patient's outcome and (digital) response prediction predicts how a given cancer is likely to respond to a specific treatment [45, p.90]. Such estimates about an unknown future is not something even the pathologists are willing to provide on their own because they are not able to make a sufficiently reliable prediction. This makes it likely that these tasks are in general subvisual, a property that has been recognised in previous studies, if the true outcome or prediction is at all estimable in all cases [8, p.39].

However difficult, providing robust and reliable estimates about the general outcome or the outcome with a specific treatment may lead to improved understanding of the biological processes involved in carcinogenesis, thus in turn making more reliable estimates possible and maybe even result in better novel or improved treatments. Also highly relevant, it can be used for a wiser selection of appropriate treatment for a given patient. In particular, because adjuvant treatment after surgery can cause the patients serious complications, both physically and socially, it may be better to not perform any adjuvant treatment on patients with very low probably of relapsing. This is especially relevant for the material analysed in this thesis, where a study using a well established statistical regression analysis called *Cox proportional hazards regression* could not find significant difference between different adjuvant treatments and no adjuvant treatment (giving a P-value of 68 % (!))[1], thus making it likely that the effect on the outcome of skipping adjuvant treatment is small and likely worth the risk for many low-risk patients [29, pp.1495–1496].

Nuclear image analysis

Nuclear image analysis is the field of image analysis dealing with nuclei, both intra and inter relationships. Within cancer research, popular features extracted from the digital images of nuclei uses the *grey-level cooccurrence matrix (GLCM)* [20; 45, pp.96–105; 65, pp.12–14] the *grey-level run length matrix (GLRLM)* [18; 45, pp.106–109] and fractal estimates [45, pp.114–117; 65, pp.22–24]. [45, p.91]

[1] It should be noted that the study in [29, p.1495] only included 13 (of 284) patients with no adjuvant treatment and that practical circumstances could make any dataset more biased than statistically expected, in particular that the assumed statistical independence could be slightly questionable in general, but it is difficult to question the result because of this due to the highly significant P-value.

Unfortunately, a majority of researchers violated the assumptions of the statistical evaluation methods, in fact, in a recent study only 30 of 160 reviewed papers used acceptable statistical evaluation methodology [45, p.137]. The typical effect of this violation are overoptimistic results [60, p.75; 54, pp.293–294]. Even more severely, the violation has different influences on the performance of different features, thus researchers may be led to wrong conclusions on which features are appropriate [54, pp.293–294]. As will be more fully explained in section 6.6, the effect of adding more features under such violation is also misleading because the result becomes gradually more overoptimistic and rather quickly, depending on the dataset under study, completely useless to classify novel cases [60, pp.72–76].

1.3 The present study

1.3.1 Aim

The main aim of the present study has been to develop an automatic algorithm that reliably estimates the prognosis of novel patients with early ovarian cancer using adaptive features based on the spatial entropy within each of a couple of segmentation classes. These class dependent spatial features are novel features which are adaptively extracted from some matrices coined *class specific dual entropy matrices (CSDEMs)*. Each class specific dual entropy matrix (CSDEM) attempts to capture some specific contextual information present within its corresponding segmentation class. The segmentation will be the result of applying one of some proposed segmentation methods. The study will also include an evaluation of the performance of these features and other promising features for the given dataset.

1.3.2 Strengths and limitations

This study is based on a specific set of digital images acquired from women with early ovarian cancer who have undergone a nearly identical surgery and most are given one of two adjuvant treatments (details will be given in chapter 2). Furthermore, the same pathologist has selected the relevant part of the cancer tissue and the same trained personnels have prepared the tissue segment using the same standardised techniques, acquired the digital images using the same equipment, selected the useful cell images and segmented the cell images using a manually chosen global threshold, and, in addition, all patients lived under similar environmental conditions as country and period of time.

These precisely defined circumstances have mainly two advantages. First of all, the precise definition makes it *relatively* easy to reproduce a similar scenario. Secondly, the mixture of different circumstances may significantly reduce the prognostic value of our methods, thus it will be relatively easy to detect whether our methods are of prognostic value for such precisely defined circumstances. This latter advantage does however also imply a limitation as it restricts the generality of our results to the defined circumstances, and therefore also the extent to which we can claim our methods are of prognostic relevance. However, most of these factors can be assumed to be of minor importance or irrelevant, e.g. the adjuvant treatment (as commented in section 1.2.1) and the

environmental conditions, or at least minor under similar conditions, e.g. by using the same standardised techniques, similar equipments and other similarly trained personnel, and thus the methodology or maybe even the results may be appropriate to use for more general situations. Any generalisation must of course be performed with caution and new proper tests should be performed to evaluate the generalisation. This is particularly important if either the type or stage of cancer or the type of surgery is changed, as such changes alter the foundation of the analysis. In terms of generalisation and the fundamental changes mentioned, it is duly noted that such fundamental changes may cause the type of adjuvant treatment to be prognostically significant and thus central in the limitations of these generalisations.

1.3.3 Organisation

This thesis is organised in the same manner as the stages in the design of a pattern recognition system [56, p.252]. We will begin with describing the dataset in chapter 2, both how it is collected and potential challenges associated with it. In the search of relevant features, we will describe previous work relevant for our study in chapter 3. This description will emphasise the relevant features for this study, but also include the description of a method called *DNA ploidy analysis*, which is not based on image analysis. We will then continue to describe the proposed features based on the CSDEMs in chapter 4. The search for relevant features is concluded in chapter 5 by a summary of the features that will be applied in this study.

The used classification methods and the most important challenges associated with supervised learning will be discussed in chapter 6. This chapter will also include a discussion of the evaluation methods. The evaluation of the relevant features will be performed in chapter 7, which also will include a thorough discussion of the features, the classification results and their interpretation. We will finally conclude in chapter 8 with some general comments and present a list of suggestions for further work in chapter 9.

The structure of this study aims to provide a fluent reading of the entire thesis and is inspired by the world's most recommended format for scientific papers [67], *Introduction, Material and Methods, Results, and Discussion (IMRaD)*, but is not restricted to this recommendation. In particular, we note that the results and the discussions are interleaved to ease the reading. A discussion is also presented in connection with every topic where it is feels natural to do so in order to ease the reading while providing a good understanding of the topics. Also, a few results from previous work are reprinted in chapter 3.

Chapter 2

Material

We will in this thesis study a dataset of digital images captured from 134 patients treated for early ovarian cancer during the period 1982–1989. There exists on average about 281 digital images for each patients (ranging from 220 to 314), each imaging a DNA-specific stained nucleus of the patient. In all patients, the ovarian cancer is not a metastasis, i.e. it is the *primary cancer*. Both ovaries and the uterus were completely removed in all patients, either in a surgery at a county hospital or at The Norwegian Radium Hospital, or in two surgeries, one at both mentioned locations. The vast majority of the patient had either chemotherapy or intraperitoneal instillation of ^{32}P as adjuvant treatment, but, as earlier mentioned, a well established statistical regression analysis on a superset of the patients in our dataset could not find significant difference between different adjuvant treatments and no adjuvant treatment, thus it seems reasonable to treat the dataset as homogeneous with respect to treatment. [29, p.1495–1496]

To be able to make a precise presentation, a couple of definitions are needed. Define *relapse of ovarian cancer* as the occurrence of a cancer which is assumed to be related to the surgically removed ovarian cancer. Let *relapse-free survival rate* denote the proportion of patients who did not relapse the ovarian cancer within a specified time after the last relevant surgery. We note that this quantity should be computed using survival analyses to allow censoring of the patients who died of other causes, i.e. not of a relapsed ovarian cancer, where *censoring a patient* refers to the ability of survival analyses to use the information that a patient did not relapse before its disease-unrelated death, while ignoring its presence after its death.

All patients where followed up until their death or 31st December 1998 [29, p.1495]. For each patient, the relapse of ovarian cancer and time of death where recorded.

Our dataset can be seen as a learning dataset extracted from a superset containing 284 patient, 28 of whom died of causes unrelated to ovarian cancer. No patients who died from other causes within ten years where included in our dataset. The patients in our dataset are categorised as either relapse-free survival or relapse of ovarian cancer, both within ten years, and these categorised are named *good prognosis* and *bad prognosis*, respectively. In total, 94 patients were categories as good prognosis and 40 patients as bad prognosis.

2.1 Ovarian cancer

Ovarian cancer is one of the most common gynaecologic malignancies and the fifth most frequent cause of cancer death in women. Under some restriction, e.g. at least twenty years old women where their ovarian cancer is the primary cancer, over 95 % of ovarian cancers are located in epithelial cells. Such cancers are called *carcinoma* and the patients in our dataset all have a specific type of carcinoma called *adenocarcinoma*, which contributes to nearly 90 % of all cases of ovarian cancer. [28, pp.133,136]

Ovarian cancers are as other cancers staged according to the extent the cancer has spread. For ovarian cancers, the cancer is diagnosed as stage I, the most restrictive stage, if its growth is limited to the ovaries [28, p.134]. All patients in our dataset are diagnosed as this stage, but without lymph node staging, and there exist no borderline cases.

2.2 Imaging procedure

The following imaging procedure is known as *monolayer preparation* and will project each complete nucleus on the surface of the camera's sensor chip, which stands in contrast to *histological sectioning* where the sections are cut much thinner which emphasis the analysis of the nuclei internal structures.

After surgery, a single pathologist has selected the relevant part of the cancer tissue of each patient. Each tissue sample were fixed in 4 % buffered formaldehyde, and then paraffin-embedded before it was cut in two 50 μm sections. The sections were then enzymatically digested (SIGMA protease, type XXIV, Sigma Chemical C., St. Louis, MO) for preparation of isolated nuclei. After placing the nuclei on a glass slide, they were Feulgen-Schiff stained according to an established protocol and another glass slide was mounted on top of the stained nuclei. This concluded the preparation of the nuclei for imaging. [47, p.77]

The imaging was preformed using the Fairfield DNA Ploidy System (Fairfield Imaging LTD, Kent England) which consisted of a Zeiss Axioplan microscope equipped with a 40/0.75 Zeiss objective lens, a 546 nm green filter and a black and white high resolution digital camera (C4742–95 Hamamatsu Photonics K.K., Hamamatsu, Japan). This imaging technique is in the category of *light microscopy imaging*. By moving the slide under the camera and using manual focus with a physical focus level of about 1.5 μm, digitalisation of the nuclei were stored in virtually overlapping monochrome images of 1024x1024 pixels, corresponding to a physical resolution of 166 nm per pixel, with 10 bits pixel depth. The digital images were then shade corrected, see details in section 2.2.1 below. Then trained personnels segmented the nuclei using a manually chosen global threshold and removed non-epithelial, incomplete and connected nuclei. [47, p.77]

Some examples of the resulting images are visualised in figure 2.1. Since the imaging is based on the proportion of the emitted illumination that reaches the camera's sensor chip (rather than just reflection of the nuclei) and the DNA-specific staining is designed to absorb the emitted illumination, the nuclei will be visible as dark objects on a light background (before segmentation) and the level of darkness is positively proportional to the density of the DNA. The technique of monolayer preparation and the use of a narrow focus will result in an averaging

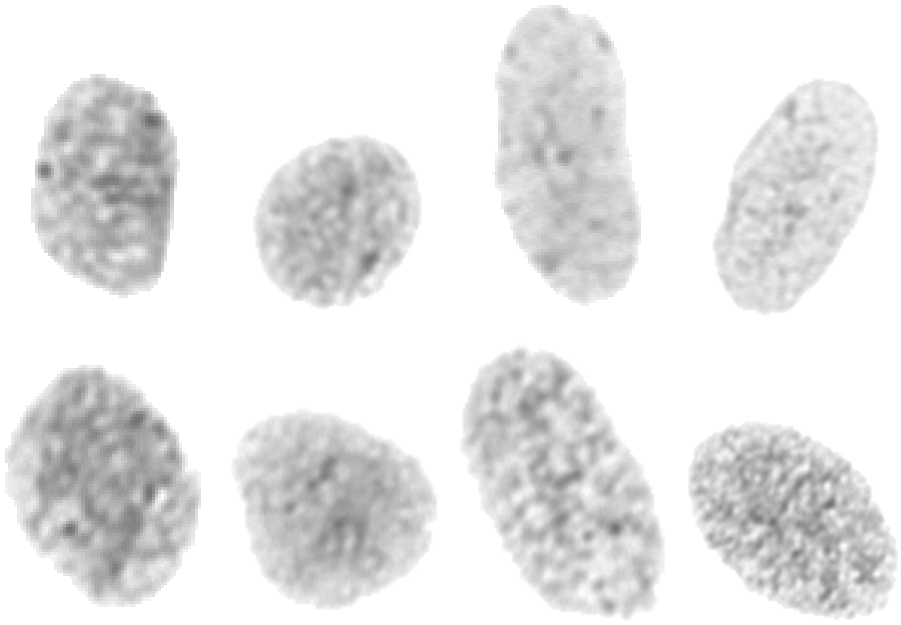

Figure 2.1: Examples of images resulting from the described imaging procedure. The images in the top row are taken from a patient with good prognosis while the bottom row images are taken from a patient with bad prognosis. The number of pixels that are segmented to be a part of the nucleus, hereafter called cell pixels*, are from the left to the right 2475, 3215, 3690 and 5014 for the top row and 3670, 4419, 5485 and 8847 for the bottom row.*

which also includes contribution of parts of the nuclei which are not inside the optical focus depth. More precisely, each resulting pixel may be viewed as the average of trivariate normal distributions of infinite sections perpendicular to the axis from the corresponding point on the camera's sensor chip to the illumination source, where each distribution is centred at the intersection between the section and the mentioned axis and have uncorrelated and equal variances that are positively related to the least physical distance from its centre to the level where the image is in focus.

2.2.1 Shading correction

Let us refer to the virtually non-overlapping 1024x1024 images of the nuclei, captured as described above, as the *original images* of the patient. In addition to these, a single image is acquired for each patient when imaging a region clear of tissue sample. This image will be referred to as the *reference image* of the patient. Ideally, the reference image would be completely white, indicating that no light is absorbed, reflected or refracted when the tissue sample is not present. However, some absorption, reflection or refraction will occur. The glass slides are in particular subject to both absorption and reflection, thus likely giving a generally darker region, though probably not a dark one. The illumination

may also be somewhat uneven, thus the reference images may be an unevenly illuminated greyed image. Furthermore, minor defects in the microscope can contribute to artifacts visible on exactly the same spots in all the original images and the reference image. This last artifact, which may be relatively large in comparison with the small nuclei, may be the most severe for *texture analysis methods*, a class of methods that can loosely be characterised by those methods that measures or exploits the interpixel relationships. In general, all causes that makes the reference image deviate from a completely white image may also result in a decreased classification performance, thus, correcting the original images using the reference image may improve the performance of the classifier and is of course the reason to capture the reference images.

Let us imagine that each pixel in the original image as the result of a light beam from the illumination source, through the glass slides and potentially a part of the nucleus, and onto the corresponding point on the camera's sensor chip. From basic physics we now that the absorption, reflection and refraction occurring at any point (in the physical space) will result in a multiplicative proportional decrease of the intensity of the light beam. Thus, in our simplistic model of the origin of each pixel in the original image, the associative and commutative properties of multiplication allows us to isolate the contribution of the nucleus by dividing the original image with its reference image.

Even if assuming equal environmental conditions when capturing the different image pairs, the mentioned simplistic model is in general not sufficient to isolate the contribution of the nucleus. The reason is that the presence of nuclei may cause light beams to refract and to end up hitting the camera's sensor chip on a different region than it originally was directed toward. The resulting measured deviation on the camera's sensor chip is thus correlated with the original images in itself. Some of such deviations are associated with the density of the DNA and can be accommodated for, and we will in section 3.1.3 see a study using DNA ploidy analysis which does such accommodation, but, in correspondence with previous image analysis studies on our dataset, we will not bother to correct for these deviations here.

Due to the mentioned problem with the refraction and because slightly different circumstances can be present when acquiring the images, the measured intensity at some elements in some original images may exceed the measured intensity at the corresponding elements in the reference image. Because we generally expect these error to be small, we will simply deal with them by setting the relevant elements to one. Finally, each element is multiplied by the maximum grey level value and rounded down to the nearest integer[1].

To summarise, we perform the shade correction of each element (i, j) in each original image O by computing:

$$\lfloor (G-1) \min \left\{ 1, \frac{O(i,j)}{R(i,j)} \right\} \rfloor \tag{2.1}$$

where G is the number of grey levels in the original image ($2^{10} = 1024$ for our image) and R is the reference image of the same patient. Figure 2.2 illustrates

[1] The type of rounding is unimportant, though one may argue that rounding down is better because these image typically use much of the higher intensity values, but few or none lower intensity values, thus rounding down can give a slightly higher *dynamic range*, i.e. a slightly higher ratio of the maximum grey level value to the minimum grey level value of the resulting images.

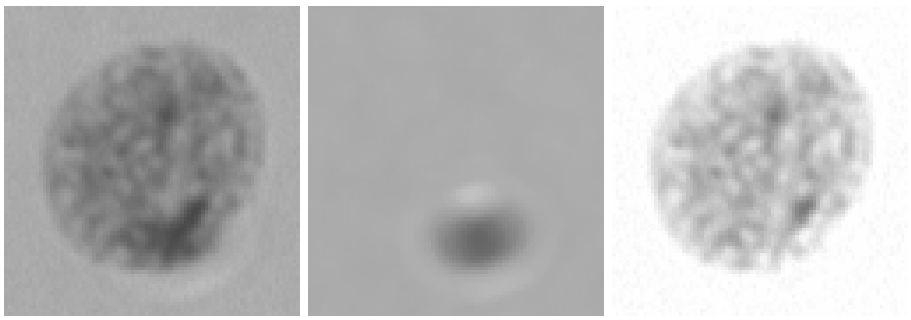

Figure 2.2: At the left we see the nucleus part of an original image (which is naturally much smaller than the entire 1024x1024 original image it is contained within) and the same region of the reference image is shown in the middle. Both images are taken from the dataset under study. The reference subimage clearly shows an artifact (e.g. in the microscope), and the same artifact is also visible in the original subimage. At the right we see the result of the shading correction performed by using equation (2.1) *on these subimages.*

the result for the nucleus part of an original image in our dataset.

The images resulting from applying the shade correction are called the *shade corrected images*. After applying the mentioned global threshold and separating different nuclei, the resulting images are called the *cell images*, though they really are images of DNA-stained nuclei of epithelial cells. We will in our texture analysis only use these cell images.

2.3 Challenges with the dataset

Before we look at the previous and present work on our dataset, let us discuss some challenges associated with it.

A general problem with imaging microscopic objects is that it is difficult to produce images with high spatial resolution. Because the images in our dataset are acquired some time ago, their resolution is also relatively low in comparison with what we could have been obtained today. In total, the spatial resolution of the cell images are low. The average number of cell pixels, hereafter called *cell area* of the cell image, is less than 4000. This, especially in combination with the projection of entire nuclei, makes the analysis of detailed textural structures difficult.

From figure 2.1 and an inspection of other cell images, it seems reasonable to assume that the human visual system are not able to decide the prognosis of all patients based only on their cell images. This assumption is enforced by previous studies and the mentioned fact that even the pathologists are not able to provide reliable estimation of a patient's prognosis (see section 1.2.1). Our task can thus be claimed to be subvisual in general, if such cell images at all contains the relevant information to perform the correct prediction in all cases. It should however be noted that there exists visual, preattentively discriminable[2] elements that has been shown to be of prognostic value, see for

[2]The term *preattentively discriminable* was introduced by Julesz to mean that the a texture

instance the results of using the GLCM in [47, p.84].

We will in the following discuss the challenges associated with the imaging procedure, then with the collection of cell images and finally with the assertion of diagnosis and outcome.

2.3.1 Challenges with the imaging procedure

Because the pathologist's selects the relevant tissue manually, this will introduce a major subjective element which may cause challenges. Also, the fixation, the enzymatic digestion and the staining may introduce artifacts. In particular, the enzymatic digestion may cause liquid to be unevenly absorbed by the nuclei which in turn affects both their absolute and relative area and texture. While these artifacts may cause deviations in the resulting cell images in our dataset, the most severe cases are identified on patient level by using the non-epithelial cells which are naturally included of the cancer tissue sample, and removed prior to forming the dataset.

A major challenge with our dataset when attempting to use texture analysis is caused by the use of the monolayer imaging technique. This is because the projection of entire nuclei is likely to hide much texture information. In particular, it is likely that many more and/or less condensed chromatin structures will partly or completely overlap. This can make the observed chromatin structures unrealistic, e.g. by being only partial or being composed of multiple chromatin structures (possibly of different condensation), and can completely hide less condensed structures behind more condensed ones.

The mentioned problem in section 2.2.1 that refraction may cause increased response in other relevant regions on the camera's sensor chip than the region the light beam originally was directed toward, may also cause some complications. Moreover, it is known that mitochondria can be connected to the nuclei at the point of imaging, which will cause increased absorption, reflection or refraction, but this is not a severe problem as the mitochondria only contributes to about 1 % of the DNA.

The net effect of the challenges with the imaging procedure is an increased uncertainty in the relationship between the observed intensity in pixels in the cell images and the true DNA content at the corresponding locations (i.e. the true measure of how condensed the DNA is a specific location). In particular, the uncertainty in pixels with high intensity, i.e. with a small amount of measured DNA, will be large. In comparison, the uncertainty in pixels with low intensity will be much less.

2.3.2 Analytical unit

An obvious difference from many image analysis problems is that we in our dataset have images of cells, but we wish to classify the patients which have many cells each. Of course, if we allow ourself to assume that a single cell can have cancer, then we could classify the cells and base the classification of the patients on this by using e.g. a cutoff value, for instance prevalence ('majority vote'), or a function of a measurement of each cell within a patient that describes both the affiliation of the cell and the certainty of this affiliation.

---

pair was immediately perceived as two differently textured regions by the human visual system [65, pp.4–5].

There are several problems with basing the classification on the cells. Firstly, it is not normal to claim that a single cell can have cancer. This can be seen from the presented definition of cancer in section 1.1.2, which is based on the invasive growth of the entire tumour. However, we have seen that the invasive growth is likely to be founded in DNA changes in its nuclei, thus this problem can be said to be more of a theoretical than practical art.

Other problems are however also of practical importance. Two correlated problems of this art follows from the definition of cancer as invasive and the assumption that it is sufficient that a single survival proficient cell has spread for a metastasis to be formed in time. The first of these problems makes it likely that there exists several normal cells within even the most essential part of the tumour[3], while the other makes it likely that a highly invasive malignant tumour may already have spread while a more compact, but still invasive, tumour may still be restricted to its original location (the ovaries in our case). Both these problems indicates that making the decision on patient level based on any cutoff value or any function that weights each cell contribution equally is likely to be suboptimal in general. It also indicates that claiming that all cells within a bad prognosis patient are 'bad prognosis cell', or even cancer cells at all, is highly questionable and likely to mislead the classifiers constructed using this assumption.

Lastly, another important problem with classifying on a cell level arises from the fact that cells can not be seen as independent samples in general, even cells within the same patient are in most circumstances dependent [45, p.119; 60, p.65]. A classification of patients based on treating the cells as independent samples have been shown to lead to highly biased outputs for all tested statistical procedures, and this error can neither be diminished nor controlled [64, pp.191,200].

As an alternative to basing the classification of patient on multiple cell classifications, we could classify the patient directly. We could state this as using the patient, and not the cell, as the *sampling unit* [45, p.119] or the *analytical unit* [64, p.192]. Using the patients as the analytical units is a recommended approach [64, p.200] despite the fact that this drastically reduces the number of patterns, see section 6.3 for a discussion on the importance of many patterns. It can be noted that a cell analysis based on a cutoff value could be mimicked in a patient analysis by simply obtaining the same feature vectors from each cell image as would be done in the cell analysis, and then using the average of these feature vectors within each patient as the feature vector of the corresponding patient. This procedure would correspond to a weighted cutoff rule where the cutoff value would be dynamically determined (indirectly in the design of the patient classifier). Other cell analysis could also been mimicked, e.g. if using the function mentioned above that describes the affiliation and its degree of a particular cell, changing the average of the feature vectors in the procedure just described to a similar function of the feature vectors will result in a patient analysis procedure that mimics the corresponding cell analysis approach.

[3]The reader may reason that because the pathologist classifies the tumour as malignant based on invasive growth, it must be possible for such trained personnel to separate the normal cells from the cancer cells using this intercellular relationship. Indeed, this is possible in general [60, p.70]. However, the monolayer imaging procedure has destroyed this relationship at the point of imaging, thus making this type of separation impossible as a postprocedure for our situation.

By using the patient as the analytical unit, we potentially eliminates all the above mentioned problem with classifying cells. However, as features in image analysis are generally extracted from each image, many nuclear image analysis approaches simply averages, sums or obtains other characteristics from the distribution of cell feature vectors [45, p.119]. When obtaining such a characteristic, the effect of possible dependencies within a single patient should be noted, but more importantly, the mentioned problems associated with the tumours heterogeneity and their differences in degree of invasive growth will still be present. These two latter problems could be accounted for by selecting one or multiple subsets of cells with specific prognostic value and extracting tuned characteristics from this or these feature distributions (possibly resulting in several features). A method related to this approach is DNA ploidy analysis which groups the cells in different categories based on their DNA content, see section 3.1 for details. Similar approaches could also be suspected to have prognostic value when using texture analysis, but other subsets could also be obtained. In particular, several studies have attempted to detect such subsets by applying clustering [45, p.118], but not necessarily used the estimated clusters to extract specialised features from subsets of cells for classification purposes. Another strategy that also will account for the same two problems is to look for special types of cells with decisive prognostic value. Such an approach is similar to the pathologists work, but the problem will then be to appropriately defined the characteristics of these special cells. [60, p.71]

Lastly, we should mention a third way of analysing the dataset which is based on *nested variance analysis*. Such approach allows a direct analysis of the highly hierarchical structure of patients and cells found our dataset, and also allows us to perform independent tests on both cell and patient level. We note that such methods seems adequate and could also have been recommended, but this approach will not be persuaded in this study due to a limited development of classification methods in this context [64, pp.193,200]. [60, p.71]

2.3.3 Why relapse?

A natural question to ask is *why do anyone relapse?* For cancer in general, we mentioned two main reasons in section 1.1.2. The first was that the treatment, here the surgery (or surgeries), failed to remove the entire malignant tumour. This reason is highly unlikely for the patients in our dataset, because the ovaries are strongly separated from the surrounding organs and all patients in our dataset had both their ovaries and their uterus completely removed in surgery. We are thus left with the second main reason, which is that a spread had already occurred at the time of surgery, or before the last surgery for the patients who underwent a partial surgery at a county hospital prior to a new surgery at The Norwegian Radium Hospital. For our patients, this could either be an unrecognised metastasis in a lymph node or a spread that had not yet grown enough to be discovered [29, p.1499]. It is duly noted that such a spread contradicts the mentioned definition of the stage I ovarian cancer, thus these cases are in fact not stage I ovarian cancer. Thus, when applying the mild assumption that the entire malignant tumour was removed in surgery, our dataset should not have contained any patients with bad prognosis because we should have been dealing with only stage I ovarian cancer.

Apparently, some patients in our dataset do not have stage I ovarian cancer

and have thus been misdiagnosed to this category. This is however not the only problem related to the recorded prognoses. Another important problem is that some patients may have been recorded as relapse of ovarian cancer when they really died of a unrelated cancer occurrence - or not cancer at all. The opposite error is also possible, i.e. that a patient's relapse of ovarian cancer within ten years after surgery was not recorded.

We should comment the effect of this last possible error, the incorrectly recorded outcome, on the classifiers we will later design. While most of the mentioned challenges are likely to 'only' result in some more or less dramatical changes in the cell images, this last error will directly confuse our classifiers when its designed and a single incorrectly recorded outcome can cause multiple misclassifications. The need to build robust and general classifiers are always strong in image analysis problems, but we can note that the possible presence of recording errors makes the need of robust classifiers even more prominent in this study. It should also be noted that these errors, along with the other challenges or errors, makes it probable that even a perfect classifier, i.e. a classifier that always predicts the true prognosis, does not achieve a 100 % correct classification rate, thus a minor error rate is not only acceptable, it is probably also preferable.

Chapter 3

Previous work

Several studies have been published on datasets overlapping with our dataset. We will in this chapter look into the most effective methods that have been tested on such datasets and include a discussion of their positive and negative properties.

We will begin this chapter with a discussion of the DNA ploidy analysis, which will be the only presented method that is not based on image analysis. The relevant published study using this approach is the already mentioned study by Kristensen et al. [29] which applies survival analysis on the superset of our dataset containing 284 patients.

A discussion of some of the image analysis methods previously used on datasets overlapping with our dataset will follow. Because of the assumed prognostic value of the DNA organisation, see section 1.1.3 for details, most image analysis methods will in this context be texture analysis methods. Also, the subsequently proposed method in chapter 4 is based on texture analysis, thus such methods are of primary interest in this study.

3.1 DNA ploidy analysis

In DNA ploidy analysis, we obtain a histogram of the estimated DNA content of each cell belonging to a single patient and classifies the histogram into different ploidy types, e.g. *diploid*, *tetraploid*, *polyploid* and *aneuploid*. Each patient can then classified as good or bad prognosis according to their ploidy type, e.g. the study by Kristensen et al. [29, p.1495] indicates the assignment of the diploid and tetraploid cases to good prognosis and the polyploid and aneuploid cases to bad prognosis.

3.1.1 DNA content

The DNA content of a single cell is estimated as the *integrated optical density (IOD)*. Define $A \in \mathbb{N}_0^{m,n}$ as an image with height m and width n, and define also $f : \mathbb{N}_0^{m,n} \to [0, \infty)$ as the function that gives the IOD of the specified image. Then f is defined as:

$$f(A) = -\sum_{i=1}^{m}\sum_{j=1}^{n} \log_{10} \frac{A(i,j)}{B(i,j)} \tag{3.1}$$

where $B \in (\mathbb{R}^+)^{m,n}$ is the background intensity image corresponding to A and the entries of A and B is one-indexed. The negation of the summand in the equation above is called the *optical density (OD)* of the corresponding pixel and is a measurement of the DNA content in that specific pixel. In practise, some ODs may be slightly negative because of measurement errors. Such elements are simply ignored when computing the IOD.

3.1.2 Ploidy classification

Diploid cells are cells with normal DNA content, i.e. 46 chromosomes. *Tetraploid cells* have twice the normal DNA content, i.e. 92 chromosomes. Because some cells can be expected be in the mitotic phase and the genome of such cells have been duplicated prior to entering this phase, some tetraploid cells can be expected in any normal tissue. Cells with four and eight times the normal DNA content is called *octaploid* and *hexadecaploid*, respectively. All cells with a positive power of two times the normal DNA content can be called *euploid cells*.

The ploidy classification is typically performed by specially trained personnel using the histogram of the IOD of each nucleus belonging to a single patient and without knowledge of its recorded true prognosis, but automatic detection algorithms have also been developed. Defining precise and general classification criteria are however difficult. We will therefore only outline the classification to the four mentioned ploidy types here before we specify the criteria used in a relevant study.

A diploid histogram typically contains a large proportion of IODs corresponding to diploid cells and no other significant population, save maybe a small proportion of IODs corresponding to tetraploid cells. Tetraploid histograms are characterised by a significant proportion of IODs corresponding to tetraploid cells. The polyploid histograms are characterised by at least one significant proportion of IODs corresponding to other euploid cells. Finally, aneuploid histograms contain a significant proportion of IODs corresponding to non-euploid cells.

3.1.3 A relevant study

We will now turn to the study by Kristensen et al. [29] which uses DNA ploidy analysis to classify the superset of our dataset containing 284 patients. We will first describe the technical details made in this study and then present its results.

Firstly, let us describe the choice of the images A and B in the computation of the IOD in equation (3.1). Because the segmentation is in this dataset done after the shade correction, just as it is in our dataset, it is natural to let A be a cell image, even though we could have let it be an original image and include a shade correction in B. Instead, we will let B include two other accommodations. One of these is related to the density of the DNA, which is a part of the problem that was indicated section 2.2.1 with the refraction that is correlated with the presence of the nuclei. The other attempts to accommodate for the average effect of any non-nuclei contribution of the tissue sample at the point of imaging.

For ploidy classification, the study used specially trained personnel without knowledge of the recorded true prognosis to manually performed the classification by applying some defined criteria. By using the average IOD of some imaged

non-epithelial cells, a patient-dependent estimate of the IOD corresponding to diploid cells is obtained. It may seem unnecessary that this estimate is dependent on the patient, but it is actually rather important as the true DNA content of diploid cells varies significantly between patients.

Using the estimated IOD of diploid cells, the trained personnel manually selected any present *euploid peaks* in the IOD histogram, where a euploid peak is in this thesis defined as a peak in the IOD histogram that approximately corresponds to a positive power of two times the estimated IOD of diploid cells. The trained personnel also selected any non-euploid peaks. We will in the following refer to all selected peaks, euploid or not, as *present* if they are visually selected by the trained personnel.

The used criteria for diploid histograms were that only a diploid peak was present, that the proportion of estimates in the tetraploid peak did not exceed 10 % and that the proportion of IODs above 2.5 times the estimated IOD of diploid cells did not exceed 1 % when excluding estimates in euploid peaks. A histogram was classified as tetraploid either if a tetraploid peak was present and the proportion of its IODs exceeded 10 %, and that the proportion of IODs above 2.5 times the estimated IOD of diploid cells did not exceed 1 % when excluding estimates in euploid peaks, or if a tetraploid and octaploid peak was present and that the proportion of IODs above 4.5 times the estimated IOD of diploid cells did not exceed 1 % when excluding estimates in euploid peaks. The presence of a diploid peak was optional in both cases. Furthermore, a histogram was classified as polyploid if a octaploid and hexadecaploid peak was present, optionally with the presence of a diploid peak and/or a tetraploid peak. Finally, the histogram was classified as aneuploid if none of the above characteristics fits, i.e. if a non-euploid peak was present or if the number of DNA content above the specified limited exceeds 1 %[1]. Figure 3.1 shows the result of applying the described IOD and ploidy classification on the images of some patients in our dataset (and this superset). [29, p.1495]

The result of applying the obtained ploidy type to estimate the prognosis of the patients are convincing. If separating on ploidy type, the estimated ten year relapse-free survival rates are 95 %, 89 %, 70 % and 29 % for the diploid, tetraploid, polyploid and aneuploid classification group, respectively. We can note that only ten patients are classified as polyploid, thus this estimate is relative unreliable, and the constructed confidence intervals (CIs) reveals that the uncertainty in the tetraploid group with 64 patients are also large. The diploid and aneuploid group contains 113 and 91 patients, respectively. If we use the indicated patient classification of assigning patients with diploid or tetraploid histogram to good prognosis and patients with polyploid and aneuploid histogram to bad prognosis, the estimated ten year relapse-free survival rates are respectively 92 % and 34 %. [29, pp.1495–1496]

For better comparison with subsequent classification results, it is interesting to note the performance of this DNA ploidy analysis on our 134 patients. The classification result when using the indicated patient classification is shown in table 3.1. If we also exclude the patients with tetraploid or polyploid histogram, as we will do in some of our subsequent classification results, we obtain the clas-

[1] If we are very strict, we see that the case when only a diploid peak was selected, but the tetraploid peak contained more than 10 % of the number of estimated DNA contents, is not included. This is only of theoretical concern because the trained personnel would in practise have selected the tetraploid peak in this case and thus classified the histogram as tetraploid.

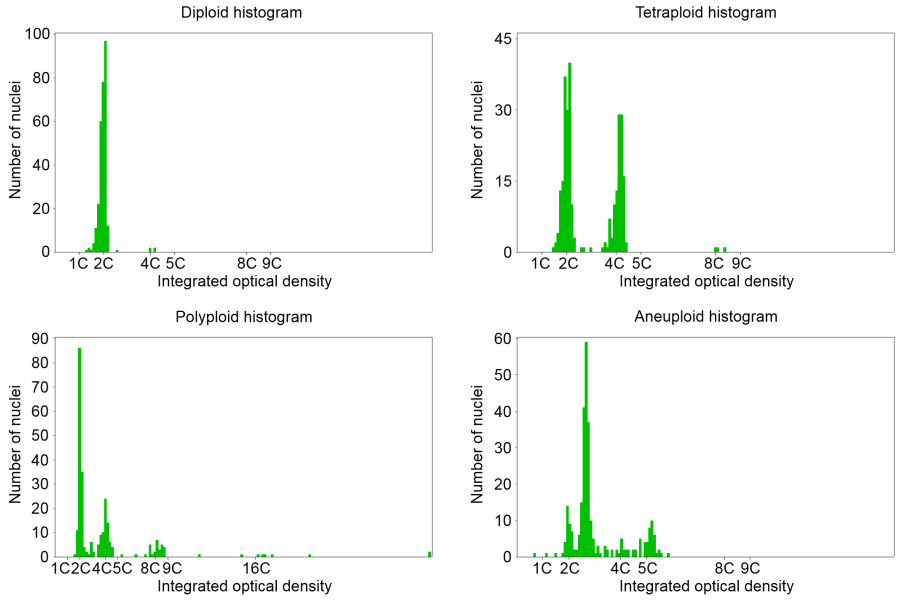

Figure 3.1: The IOD histogram with the result of applying the described IOD and ploidy classification on the images of some patients in our dataset. 2C *is the estimated IOD of diploid cells and the other C's are multiples of this value. The true prognosis of these four patients are good for the IOD histograms classified as diploid, tetraploid and polyploid, and bad for the IOD histogram classified as aneuploid.*

Table 3.1: Patient classification obtained by assigning patients with diploid or tetraploid histogram to good prognosis and patients with polyploid and aneuploid histogram to bad prognosis when using all 134 patients in our dataset. CCR *is an acronym for* correct classification rate.

	Prognosis	Patients	Correctly classified	Misclassified	CCR
	Good	94	80	14	85.1 %
	Bad	40	32	8	80.0 %
Total:		134	112	22	83.6 %

Average of the CCRs for good and bad prognosis: 82.6 %

sification result in table 3.2. We see that the estimated performances increases with about 2 % for this method when excluding the patients with tetraploid and polyploid histogram.

Table 3.2: Patient classification obtained by assigning patients with diploid histogram to good prognosis and patients aneuploid histogram to bad prognosis when using the 102 patients in our dataset with diploid or aneuploid histogram.

	Prognosis	Patients	Correctly classified	Misclassified	CCR
	Good	66	57	9	86.4 %
	Bad	36	30	6	83.3 %
Total:		102	87	15	85.3 %

Average of the CCRs for good and bad prognosis: 84.8 %

3.2 Texture analysis

In section 1.1.3, we informally justified why we hope the DNA organisation is a prognostic marker for cancer. Because of this, most image analysis studies which use a dataset overlapping with our dataset are based on texture analysis methods. We will in this section discuss some of these methods.

We will begin with an introduction to texture analysis in general with some comments regarding its general use on datasets overlapping with ours. This introduction is followed by a description of the texture features which are extracted from any (possibly multidimensional) array; first the basics, then an adaptive extraction technique, thirdly a discussion of adaptive texture features in general and finally the relevant usage of such features on datasets overlapping with ours.

We will end this section with the description of a different texture analysis method that has recently been proposed and evaluated on exactly the same dataset as ours with promising results [49, pp.1,5–6,71]. This is a structural and statistical approach that segments each cell image using two adaptively chosen thresholds and then computes features as statistical characteristics of the segmented structures within each cell image. The combination of such a structural approach and the adaptive extraction technique will form the foundation for the subsequently proposed method which includes a novel choice of the mentioned array.

3.2.1 The basics

A number of different definitions of *texture* can be found in image analysis, but no definition is generally agreed upon. We can get a grasp of the meaning of texture by indicating some of its assumed properties. A commonly agreed upon property is that texture is a feature of a region, in particular, a single pixel can not have texture. Texture can also be said to be characterised by its inclusion of the interpixel relationships. Often, one assumes that a texture consists of a large number of basic elements known as *primitives*, which also makes the presence and category of textures depended on the spatial resolution. [65, pp.1–3,11–12]

Tuceryan and Jain [65, pp.11–28] described four main texture methods; *statistical*, *geometrical*, *model based* and *signal processing methods*. In the statistical methods, one typically computes some statistics of the grey level values and then extracts some features from these statistics. The geometrical methods

are based on the assumption of primitives. Typically geometrical features are properties (e.g. statistical measurements) of an estimate (e.g. a tessellation (tiling) of the image plane or a segmentation if multiple primitive types are assumed to exist) of these assumed basic texture elements. Model based features are typically estimates of some specific parameters or prominent properties of a model that the image is assumed to be a realisation or a generation of. Lastly, the signal processing methods typically perform frequency analysis of the image. The use of such methods can be based on some evidence which indicates that the human visual system is sensitive to limited ranges of spatial frequencies [5, p.551], possibly quite narrow and including the local orientation [10, p.545].

It is usual in texture analysis to standardise the mean and standard deviation of the grey level values [48, p.85]. Commonly, this is done by a complete standardisation of the first-order grey level statistics by using a histogram matching, e.g. the relative gentle approach for stochastic textures of matching the normalised grey level histogram to a normal distribution [48, pp.85–86]. A recent study on our dataset has however shown that the standardisation of the mean and standard deviation in grey level values significantly reduces the prognostic value of all tested texture features, and also that ignoring the cell area has significant negative effect on the prognostic value of all tested texture features (when not standardising the mentioned first-order statistics) [48, p.94]. In light of the good results obtained on the superset of our dataset by applying DNA ploidy analysis, which is based on the sums of slightly modified grey level values, it is natural that the standardisation of either the first-order grey level statistics or the area will decrease the prognostic value because the sum of grey level values is the product of the mean grey level and the cell area. Because of this, we will not standardise the grey level values in this study, and we will keep in mind that we may want to extract area-dependent texture features.

3.2.2 The basics of texture features from property arrays

In many texture analysis methods, we will compute some (possibly multidimensional) array describing some properties of a particular scene, e.g. a particular pattern in the dataset or a subimage of a specific local window around a particular centrepixel. Such arrays will naturally be computed in many statistical methods, where two classical arrays are the GLCM [20] and the GLRLM [18]. Other texture analysis methods may also end up computing such arrays. In particular, the present method, which can be said to be a geometrical method, will compute up to a few such arrays for each scene[2], and the study by Nielsen et al. [41] proposes an approach which is based on the fractal assumption, i.e. a model based method, which includes the computation of such arrays.

We will in the following call an array describing some properties of a particular scene the *property array*. From this property array, one wishes to obtain one or some feature values which captures the discrimination essence of the property array. Such feature values will indirectly describe a part of the properties

[2]In correspondence with the discussion in section 2.3.2, the analytical unit used in this study will be the patient. This makes 'a particular scene' be 'a particular patient' for this study. The array of a particular scene will typically be the average of the arrays describing the same properties of each cell image from the particular patient, but we could have used other characteristics and/or subsets of the cell images from the particular patient, see section 2.3.2.

which the property array describes, ideally the part which is most important for discriminating different scenes.

There are several ways to extract features from the property arrays. In general, because the extraction technique must be independent of the class of the scene to allow classification of novel, unclassified scenes, the most general extraction technique is to independently assign a weight for each element in the array and let the feature value be the sum of the product of each array element and its weight. Any extraction technique can thus be formulated as specifying a *weight array* of equal size as the property array where each element contains the weight of corresponding element in the property array. The feature value can then be obtained by first computing the entry-wise multiplication of the property array and the weight array, and then summing all elements in the resulting matrix.

Because the weight array must be independent of the scene, we understand that some criteria of the property array are needed if the resulting texture feature should be of any discrimination value. We will thus begin with the specification of these criteria. This specification will be followed by some proposed fixed choices of the weight arrays.

Reasonable criteria of the property arrays

All property arrays should be computed in the same way from its corresponding scene. This will make them describe the same properties of each scene. Each element of the property array should have the same interpretation independently of the particular scene. However, the specific interpretation or meaning of each element is arbitrary, the same is the presence of any relationship between the elements.

All property arrays must be of equal size, i.e. have both equal number of dimensions and equal possible interval within each dimension. Because there is no requirement of non-zero elements, the property arrays may be zero-padded to have equal size. This criterion is still of importance because the way the property arrays are zero-padded is not necessary equal because of the requirement of same interpretation independently of the particular scene. In most cases however, the zero-padding should be done by adding any required number of zero-elements to the end of each dimension of the property arrays.

Predefined texture features

Predefined texture features are features resulting from using fixed weight arrays. When Haralick et al. [20] introduced the GLCM back in 1973, they included 14 such fixed weight arrays. If we define $M \in \mathbb{R}^{G_S,G_S}$ as the normalised GLCM computed from the input scene which is scaled down to a specific number of grey level values, G_S, and $W \in \mathbb{R}^{G_S,G_S}$ as the weight array (here a matrix), then some examples of the fixed weight arrays in [20, p.619] are:

- Two examples where the weights in W are based on the *values* in M:
  - *Angular second moment (ASM)*: $W = M$. The ASM-feature can thus be computed as: $\sum_{i=1}^{G_S}\sum_{i=1}^{G_S}(M(i,j))^2$. Because a homogeneous scene will contain only a few grey levels, the resulting GLCM will

contain only a few, but relatively high values, thus the GLCM ASM-feature will be relatively large for such scene in comparison with inhomogeneous scenes. We can thus say that the GLCM ASM-feature measures the homogeneity of a scene. In general, the ASM-feature measures the homogeneity of a normalised property array.

- *Entropy*: $W(i,j) = -\log M(i,j)$. The Entropy-feature can thus be computed as: $-\sum_{i=1}^{G_S}\sum_{i=1}^{G_S} M(i,j)\log M(i,j)$[3]. Because a inhomogeneous scene will contain many grey levels, the resulting GLCM will contain many relatively small values, thus the GLCM Entropy-feature will be relatively large for such scene in comparison with homogeneous scenes. We can thus say that the GLCM Entropy-feature measures the inhomogeneity of a scene. In general, the Entropy-feature measures the inhomogeneity of a normalised property array.

- An example where the weights in W are fixed with respect to their position, i.e. only based on the *position* in W:
  - *Inverse difference moment (IDM)*: $W(i,j) = \frac{1}{1+(i+j)^2}$. The IDM-feature can thus be computed as: $\sum_{i=1}^{G_S}\sum_{i=1}^{G_S} \frac{M(i,j)}{1+(i+j)^2}$. The IDM-feature weights the off-diagonal elements of the property array quadratically increasingly. We can thus say that also this feature measure the homogeneity of a scene.

It is of course not hard to suggest new fixed weight arrays. It is also possible to create datasets for each of the suggestions where that specific suggestion performs the best. The problem is however to find, or design, a weight array that performs at least relatively well for the specific problem at hand. This should however not be done by evaluating a large amount of weight arrays because of problems with overfitting (see section 6.3), not even if we are to use dimension reduction or feature selection. Instead, we should only evaluate a few weight arrays which we believe perform reasonably, or at maximum select only several weight arrays and apply dimension reduction or feature selection. It is therefore important to be able to make qualified suggestions or to have methods that are able to generically design reasonable weight arrays.

3.2.3 A set of adaptive texture features computed from a particular property array

We will now discuss a technique for adaptive extraction of a few features from each property array. The technique was first described by Albregtsen et al. in [1]. The basic principle is to let the property arrays of all scenes in the learning dataset design the weight array by estimating the discrimination value of each element in the property array with respect to the true class. Because the weight array in such a technique will depend on the learning dataset, this is an adaptive approach, and the resulting texture features may therefore be called *adaptive texture features*.

[3] Haralick et al. [20, p.619] recommends to add an arbitrary small positive constant to each element in the GLCM M prior to computing the Entropy-feature because of problems with log(0) being undefined. Nielsen et al. [45, p.99] describes a different approach, here zero-elements in M are simply excluded from the sum.

We will begin with a description of the *Mahalanobis distance*, which will provide us with one way of estimating the discrimination value of each element in the property array based on the property arrays of all scenes in the learning dataset. The details of how to apply the Mahalanobis distance to obtain such estimates of the discrimination value will follow.

In addition to the criteria of the property arrays that are mentioned above, the technique described in this subsection is most appropriate when any particular element in all property arrays of each true class is a set of independent realisations from a normal distribution and these class-dependent normal distributions have equal variances. This is a consequence of basing the technique on the Mahalanobis distance. To restrict the extent of this thesis, we will not describe (or use) other adaptive extraction techniques in this study, but we will return to possible generalisations when we suggest further work in chapter 9. Also, because our dataset only includes two classes, we will restrict our attention to this case in the following description of the adaptive extraction technique based on the Mahalanobis distance.

The Mahalanobis distance

The *Mahalanobis distance* [34, p.50] of $\vec{x} = (x_1, \ldots, x_n)$ from a distribution with expectation $\vec{\mu} = (\mu_1, \ldots, \mu_n)$ and an invertible covariance matrix Σ can be defined as:[4]

$$m(\vec{x}) := \sqrt{(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})} \tag{3.2}$$

Each contour line of this function corresponds to a unique and precisely equal contour line of the multivariate normal distribution with expectation $\vec{\mu}$ and covariance matrix Σ. This function is thus particularly appropriate for measuring the distance of a vector $\vec{x}$ relative to a specific multivariate normal distribution, but the definition does not required this as an assumption. The measurement is appropriate for measuring the distance of a vector $\vec{x}$ relative to any specific nearly symmetric distribution, but this appropriateness will decreases with the level of asymmetry in the distribution.

Lets analyse equation (3.2) to obtain a better understanding of it and the effect of asymmetry. Firstly we note that the covariance matrix is by definition symmetric and positive semi-definite. As we also have assumed that it is invertible, it is both symmetric and positive definite and thus defines a vector norm[5]. Therefore, equation (3.2) simply defines the vector norm of a centralised version of $\vec{x}$, with respect to the expectation $\vec{\mu}$, measured in terms of the norm defined by Σ^{-1}. Using this knowledge, it is easy to understand the severity of asymmetry because any norm is by definition symmetric, thus the asymmetry can not be captured by the covariance matrix. This observation could also be indicated directly from the symmetry property of the covariance matrix, however, the information that equation (3.2) defines a norm is more restrictive and

[4]To emphasise that this function defines a distance or norm, it could be reasonable to denote it as d, but m has here been chosen to prevent confusion when this distance is applied in a subsequent context where d is already defined.

[5]The inner product norm is a vector norm, see [33, p.222] for the non-trivial part of the proof, and any symmetric and positive definite matrix rather obviously defines an inner product and thus also a vector norm, see [33, p.221] and [33, p.91] for a formal definition of an inner product and a vector norm, respectively.

also tells us that the increased deviation of $\vec{x}$ from $\vec{\mu}$ in any direction will strictly increase the measured distance (this is a result of the positive definiteness of Σ).

Assuming we have two points, $\vec{x}$ and $\vec{y}$, we can define the Mahalanobis distance between them as:

$$m(\vec{x}, \vec{y}) := \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})} \tag{3.3}$$

Similarly with the first definition, this distance measurement is best applicable when $\vec{x}$ and $\vec{y}$ are related to the same at least nearly symmetric multivariate distribution or two at least nearly symmetric multivariate distributions with common covariance matrix Σ, in both cases particularly appropriate for the case of multivariate normal distributions, but can be defined in general though its relative meaning decreases with the level of asymmetry of the distributions.

The *Mahalanobis distance between two classes* with distributions with means $\vec{\mu}_1 = (\mu_{1,1}, \ldots, \mu_{1,n})$ and $\vec{\mu}_2 = (\mu_{2,1}, \ldots, \mu_{2,n})$ and common covariance matrix Σ are defined by simply inserting the means in equation (3.3) [13, pp.35–36,107]:

$$m(\vec{\mu}_1, \vec{\mu}_2) := \sqrt{(\vec{\mu}_1 - \vec{\mu}_2)^T \Sigma^{-1} (\vec{\mu}_1 - \vec{\mu}_2)} \tag{3.4}$$

Again, this distance measurement is best applicable in the case of at least nearly symmetric multivariate distributions with common covariance, preferably equal with the exception of the expectation, and in particular appropriate for the case of multivariate normal distributions, but also this distance measurement can be defined in general though its meaning decreases with the level of asymmetry of the distributions.

If we further assume that the covariance matrix is diagonal, which is equivalent to independent components in the case of normal distributions, the Mahalanobis distance between the classes reduces to:

$$m(\vec{\mu}_1, \vec{\mu}_2) = \sqrt{\sum_{i=1}^{n} \frac{(\mu_{1,i} - \mu_{2,i})^2}{\sigma_i^2}} \tag{3.5}$$

where σ_i^2 is diagonal element number i in the covariance matrix Σ, $i = 1, \ldots, n$. In particular, the Mahalanobis distance between the classes reduces in the univariate case to:

$$m(\mu_1, \mu_2) = \sqrt{\frac{(\mu_1 - \mu_2)^2}{\sigma^2}} \overset{\sigma \geq 0}{=} \frac{|\mu_1 - \mu_2|}{\sigma} \tag{3.6}$$

where μ_1 and μ_2 is the means of the distributions and σ is their common variance.

The adaptive texture features

The Mahalanobis distance between two classes uses the assumed known true parameters $\vec{\mu}_1$, $\vec{\mu}_2$ and Σ. If these parameters are unknown, but we instead have two sets of observations which we can reasonably assume are independent realisations of two normal distributions corresponding to each of the classes, we can use these sets to estimate the Mahalanobis distance between the two classes. We will now describe how such an estimate can be obtained, but we will restrict

our discussion to the univariate case because this is the relevant case for this study.

Let μ_i and σ_i^2 be the expectation and variance of each distribution, $i = 1, 2$, and m_i and s_i^2 its standard estimates. Under the assumption of equal variances, $\sigma := \sigma_1 = \sigma_2$, the true Mahalanobis distance between the classes are as given in equation (3.6). Estimating the expectations with the means and the common variance with $(s_1^2 + s_2^2)/2$, as is in agreement with previous studies on datasets overlapping with ours (see e.g. [39, p.26; 43, p.75; 47, p.79; 49, p.65]), we obtain the following estimate of the distance:

$$\hat{d}(m_1, m_2) = \sqrt{\frac{2(m_1 - m_2)^2}{s_1^2 + s_2^2}} \tag{3.7}$$

This choice of the common variance estimate is however not likely to be the most appropriate as the standard pooled variance estimate in statistics is[6] (see e.g. [11, p.492]):

$$s_p^2 := \frac{n_1 - 1}{n_1 + n_2 - 2} s_1^2 + \frac{n_2 - 1}{n_1 + n_2 - 2} s_2^2 \tag{3.8}$$

where n_1 and n_2 are the number of realisations used to estimate s_1^2 and s_2^2, respectively. The advantage with this estimate in comparison with the one used in connection with equation (3.7) is that it weights the individual estimates s_1^2 and s_2^2 based on the number of samples used to attain these sample standard deviations, i.e. relative to the expected accuracy of the estimates.

To compute the adaptive texture features, we will as mentioned allow the property arrays of all scenes in the learning dataset to design the weight array by estimating the discrimination value of each element in the property array with respect to the true class. When separating the scenes in the learning dataset with respect their true class, we for each element in the property arrays obtain two sets of observations. If we assume that each of these sets can approximately be seen as independent realisations from a normal distribution and the variance of the two normal distributions are approximately equal, we know from above that an estimate of the Mahalanobis distance between the classes is a reasonable measure of the typical deviation between the two classes at that particular element of the property array and thus also of the discrimination value of that element.

An estimate of the Mahalanobis distance between the two classes can thus provide a reasonable weight for the discrimination value of each element of the property array with respect to the true classes. However, if the estimated Mahalanobis distance between the classes at two different elements are equal, but the class with the highest expectation are opposite, then these elements are likely to contribute equally to the feature value, but they should have contributed oppositely because they describe the opposite situation. From this it should be clear that we need to separate the contributions in two depending on which expectation is greatest at a particular element. This creates two possible adaptive texture features from a single type of property array, each defined by a weight array that on average contains 50 % zeros, and each describing either only the

[6] If the maximum likelihood (ML) estimates are used instead of the standard estimates, as equation (3.8) assumes, then all subtractions should be excluded to obtain the formula for the corresponding biased ML estimate of the pooled variance.

positive or only the negative contribution with respect to one of the classes. We will define class ω_1 as the reference class and let the *positive adaptive texture feature* be defined by the weight array with the positive contributions with respect to ω_1, and similarly define the *negative adaptive texture feature* by the weight array with the negative contributions with respect to ω_2. Naturally, the positive contributions with respect to ω_1 is the negative contributions with respect to ω_2 and similarly for the negative contributions with respect to ω_2, so this definition is only of the interest for the interpretation and a precise discussion.

Because we do not known the true expectations, we must separate the contribution depending on which estimated expectation is the greatest. The problem of choosing the wrong class can be ignored because the expectation in such elements should be relative small with respect to the variance, thus the weight in these positions should be negligibly small.

The positive and negative adaptive texture feature can be combined to a single feature. Following the line of thought by Unser [68] who introduced the sum and difference histograms as an alternative to the GLCM for texture analysis, Albregtsen et al. [1] described the use of two combinations of the positive and negative adaptive texture feature, namely the sum and difference of these features. Because Unser [68, p.124] showed that the sum and difference histograms represent the entire GLCM well in terms of discrimination value, the thought was that the sum combination, which defines the *sum adaptive texture feature*, and the difference combination, which defines the *difference adaptive texture feature*, could represent the initial weight array well in terms of discrimination value.

We note that both the sum and difference adaptive texture features could easily be defined directly as a weight array. The weight array defining the sum adaptive texture feature is precisely the original weight array before the separation of positive and negative weight contributions. From the discussion above, we expect this feature to have less discrimination value than the other adaptive features. The weight array defining the difference adaptive texture feature, where the difference is the positive minus the negative adaptive texture feature, is the signed weight array resulting from signing each element in the original weight array according to the sign of $m_1 - m_2$, but maintaining the absolute value of the elements. In this case, the estimated discrimination value of each element of the property array is precisely the standard T-statistic used in pooled two-sample t-tests under the null hypothesis of equal expectations, if we use the standard pooled variance estimate to estimate the Mahalanobis distance between the classes [11, p.492]. This adaptive texture feature is likely to be the generally best performing feature because it includes both the positive and negative contributions, but these contributions have the opposite effect on the feature value.

In total, we have four adaptive texture features that can be computed from a single type of property array. While there are only two independent features, the performance of e.g. the combination of the positive and negative adaptive texture feature may be lower than the performance of only the difference adaptive texture feature because of dimensionally concerns and its risk of overfitting, see section 6.3. Thus, the four adaptive texture features are all potential feature candidates, but it makes no sense that more than two such features from a single type of property array are included after the feature selection.

3.2.4 Discussion of adaptive texture features

There are several advantages with the use of the just described and other adaptive texture features. Most obvious is the fact that they weight the elements of the property arrays according to their relevance, thus allowing high discrimination value in a few features. This allows us to select only a few weight arrays while being reasonably confident that the choices are relatively good, and in this way decreasing the risk of overfitting. Moreover, it also implies a better expected generalisation than the predefined texture features, provided that we address the drawbacks mentioned below.

Another important property is that we do not need any information about the underlying scenes to make the choice of the weight array. This makes it easier to use the features which are based on property arrays in situations where the expected structure of the property arrays is unknown. This study is an example of this as we in the subsequently proposed property arrays expect that there are elements of discrimination value, but we are not sure where the relevant elements are located. The use of adaptive fractal features in [41] is another example of this, which along with our proposed method indicates that the adaptive texture features also makes it easier to use property arrays in other contexts than the statistical methods.

While the adaptive texture features are easy to apply and are likely to provide relatively good performance, they also have several drawbacks in comparison with the predefined texture features. First of all, while it is always difficult to obtain a realistic estimate of the feature efficiency when only using the learning dataset, it is particularly difficult for adaptive texture features. This is because the weight array is designed using the learning dataset, thus the discrimination value of the resulting feature values of the learning patterns is optimistically biased compared to the true discrimination value of the adaptive texture feature. Secondly, the use of adaptive texture features may reduce the understanding of the underlying problem in comparison with the predefined texture feature. The reason is that a predefined texture feature typically attempts to describe some specific characteristics of the property array, which in turn corresponds to some specific characteristics of the underlying scenes, while the adaptive texture features have no such interpretation. In order to get an understanding of the underlying scenes when using adaptive texture features, it is thus essential to manually inspect and interpret the designed weight array.

Another drawback with the adaptive texture features are their dependence on the number of scenes in the learning dataset, a dependence which is not present for the predefined texture features. This drawback may cause overfitting if the number of scenes is not sufficiently large, a problem which will be discussed in general in section 6.3. In particular, the adaptive texture features require at least several scenes in the learning dataset to perform reasonably, and the true discrimination value of the resulting weight arrays will generally increase with the number of scenes. The dependence is greatly influenced by how accurate the property arrays of the scenes are expected to be with respect to their interpretation, which again depends on the type of property array, the scenes and the number of elements in the property array. The number of elements in the property array is often the easiest of these three to regulate (by using quantification), thus the use of adaptive texture feature versus predefined texture features may boil down to a choice between more discrimination value

in each element of the property arrays versus more precise property arrays. It should be noted that each element of a larger property array will always be less accurate with respect to its interpretation than each element of a smaller one of the same type, but the importance of this is much less for the predefined texture features than for the adaptive texture features because the predefined texture features are typically relatively much coarser.

3.2.5 The usage of adaptive texture features in relevant studies

Several studies have extracted the just described adaptive texture features on a dataset overlapping with ours. The property array has in these studies been the GLCM, the GLRLM, the *cooccurrence of grey level run length matrix (CGLRLM)* [1; 45, pp.109–110], the *grey level entropy matrix (GLEM)* [74; 45, pp.110–112] a fractal estimate and/or the *complexity curves* [2; 45, pp.112–113]. In all studies, the arrays were computed from 1D grey level signals resulting from *peel-off scanning* each cell image, a process that peels the cell image into 1D signals from its periphery and inward. The technique was used to separate the 30 % peripheral segment of the nuclei from its 70 % central segment. Two different datasets have been used in these studies, either a subset of ours which contains twenty patients in each prognosis class [43, pp.74–76,78; 47, pp.77–79.81–82], or exactly the same dataset as ours [42; 44, pp.176–179; 48, pp.86–88].

To restrict the extent of this thesis, we will among the choices of the array only discuss the most relevant one for this study, the GLEM computed from each cell image. This choice of property array (or one of its relatives) is probably also the currently most promising choice of the array for our dataset.

Grey level entropy matrix

The *grey level entropy matrix (GLEM)* [74] describes the general irregularity in the input image $A \in \mathbb{N}_0^{m,n}$, where m is its height and n its width. The value of an element of the GLEM gives the probability in the input image of observing a specific entropy value in a local window where its centre has a specific grey level value. Each element, i.e. each such probability, can be viewed as an estimate of the probability of observing the specific combination of grey level value and entropy value in the texture that the input image can be seen as a realisation of.

The GLEM is based on three parameters, or up to five if we include the type of scaling and entropy as parameters, but we will in this study always use linear scaling and the *Shannon entropy* [61, pp.393–394] with binary base, i.e. the expectation of the binary logarithm of the probabilities. The first parameter can be called *number of grey levels*, G_S, and this allows the input image to be scaled down to a specific number of grey level values. The reason to perform this quantification is to produce a more dense GLEM and thus more reliable estimates in each of its elements. The drawback is that a reduction of the number of grey levels reduces the detailed description of the assumed underlying texture and thus potentially the performance of the resulting feature(s).

The second parameter is the *window size*, w, which gives both the width and height of the local window. This parameter should be odd in order to have

a unique centre of the local window. The last parameter gives the number of quantification levels of the entropy value, Q. Quantification of the entropy value is necessary because the entropy is in general a floating point number, thus it needs to be quantified to be allowed as the variable along one of the discrete axes of a matrix. We will in this study not use the number of quantification levels of the entropy value directly as the third parameter, instead we will specify the number of quantification levels per integer entropy value, q, and compute the total number of quantification levels by using the maximum possible local entropy value.

Normally we say that $\log_2 G_S$ is the maximum Shannon entropy with binary base of an image with G_S grey levels. This bound is however not tight in general. If the local window has less pixels than the number of grey levels, i.e. if $w^2 < G_S$, then the maximum diversity (greatest information per pixel) even with unlimited number of grey levels gives a maximum Shannon entropy with binary base of $\log_2 w^2$. In general, the maximum Shannon entropy with binary base is thus $\log_2(\min\{w^2, G_S\})$. Using q quantification levels for each entropy integer, the total number of quantification levels of the entropy value, Q, is thus:

$$Q = 1 + \lfloor q \log_2(min\{w^2, G_S\}) \rfloor \tag{3.9}$$

where the possible entropy value 0 causes the addition of one.

We can now define the computation of the GLEM with G_S grey levels, window size w and q quantification levels per integer entropy of the image A as:

1. If G_S is less than the number of grey levels in A, scale A to G_S grey levels.
2. Define $X \in \mathbb{N}_0^{G_s,Q}$ as the zero matrix where Q is computed using equation (3.9).
3. For each pixel (i, j) in the image A, increment $X(A(i, j), e)$, where e is the entropy of the window with size w centred in (i, j) (if w is even, any of the four possible centres could be chosen, but the same choice should be made for every element).
4. Normalise X by dividing each element by the number of elements in A, mn. The sum of all elements in X is now precisely one.

For the computation of GLEM from our cell images, we will use a slightly adapted version of the algorithm described above because the segmentation of the nuclei gives a non-rectangular input images. In step 3, only cell pixels are included in the local window. Thus the local windows at the periphery will contain fewer pixels, which is similar to the local windows close to the border of the image for the GLEM computation described above. Moreover, the normalisation of X in step 4 is done by dividing each element with the cell area, i.e. the number of cell pixels.

A relevant study

The study of Nielsen et al. [42] was the first to use GLEM on a dataset overlapping with ours. As that study applied the technique of peel-off scanning which is irrelevant for our study, we will instead focus on a more relevant study of Nielsen et al. [46] that is just submitted. In that study, the positive and negative adaptive texture features described in section 3.2.3 were extracted from

each GLEM computed from each cell image and using a superset of our dataset containing 246 patients. The used parameters of the GLEM were $G_S = 64$ and $w = 9$. The study also grouped the cell images according to the cell area and only used the cell area groups $[2000, 2999]$, $[3000, 3999]$ and $[4000, 4999]$. This is in correspondence with the findings in [48, p.94], which identified these area groups to contain most of the information relevant to discriminate between the prognosis classes for our dataset.

The study also uses an relative of the GLEM which we will call the *4D-GLEM* property array. This property array adds two new axes to the GLEM; the window size and the cell area group. $G_S = 64$ where also used as parameter for this property array, in combination with the window sizes $3, 5, \ldots, 31$) and the cell area groups $\{[1000, 1999], \ldots, [9000, 9999], [10000, \infty)\}$.

As our study, the study in [46] also uses the patient as the analytical unit. The property array of the patient is set be the average of the arrays describing the same properties of each cell image, which is the common choice [45, p.119]. This choice is in the context of the described adaptive texture features enforced by the *central limit theorem* which states that the average of independent realisations of any distribution will converge to a normal distribution [11, p.293]. Because of some dependencies between the cells of a single patient, the direct application of the central limit theorem is in theory illegal, but we can still expect that the normal approximation of the patient averages of each element in the property array is good enough to justify the appropriateness of an estimate of the Mahalanobis distance between the classes. The assumption of common variance is however not justifiable in general and must be investigated exclusively to reveal its appropriateness, though the study does not mentioned such investigation.

When evaluating on an independent dataset containing 105 patients, the study [46] report a CCR of 62.9 % and 65.7 % when using the negative adaptive texture feature of the GLEM and 4D-GLEM, respectively. They concluded that the adaptive texture features contain prognostic information. We note that because the features based on the GLEM and 4D-GLEM property arrays will measure the general irregularity in the DNA organisation, this indicates that such irregularity seem to be a prognostic marker, which was indicated in the informal discussion in section 1.1.3.

3.2.6 A structural and statistical texture analysis

Recently, a new texture approach was proposed and evaluated on exactly the same dataset as ours with promising results [49]. The study was based on the assumption of three different types of primitives which corresponded to the dark, grey and bright regions within the cell images. Segmentation was used to estimate these primitives and several properties were extracted from the estimated dark and bright primitive types, in addition to a property extracted from the collection of all primitives within each cell image for each of the dark and bright primitive type [49, pp.20,44]. Using the definitions in section 3.2.1, this is a geometrical method.

The segmentation was based on Niblack's adaptive segmentation algorithm [38, pp.115–116]. Given a window size w, a value k and an input image $A \in \mathbb{N}_0^{m,n}$, where m is its height and n its width, the segmentation image $N \in$

$\{0,1\}^{m,n}$ is given by:

$$N(i,j) = \begin{cases} 0 & \text{if } A(i,j) \leq \mu_w(i,j) + k\sigma_w(i,j) \\ 1 & \text{if } A(i,j) > \mu_w(i,j) + k\sigma_w(i,j) \end{cases} \tag{3.10}$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, m$, and where $\mu_w(i,j)$ and $\sigma_w(i,j)$ is the expectation and standard deviation of the grey level elements in the local window of size w with centre (i,j) in A.

Because the study assumed the existence of three primitive types, this definition of Niblack's adaptive segmentation algorithm was extended to include two thresholds:

$$N(i,j) = \begin{cases} 0 & \text{if } A(i,j) \leq t_d(i,j) \\ 1 & \text{if } t_d(i,j) < A(i,j) \leq t_b(i,j) \\ 2 & \text{if } A(i,j) > t_b(i,j) \end{cases} \tag{3.11}$$

where:

$$t_d(i,j) = \mu_w(i,j) + k_d\sigma_w(i,j) \tag{3.12}$$

$$t_b(i,j) = \mu_w(i,j) + k_b\sigma_w(i,j) \tag{3.13}$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, m$, and where k_d and k_b are two parameters of this generalised Niblack's method, the third and last parameter is the window size w.

While Niblack's method is adaptive in the sense that the threshold of a pixel adapts to the local grey levels in the specified window, it required the specification of its parameters; k and w or k_d, k_b and w for the generalisation that includes two thresholds. A proper specification of these parameters are essential to obtain an appropriate segmentation, but such specification is typically only found by trial and error. Therefore, the study in [49] propose a method for estimating the parameters adaptively for each input image, which introduces a new level of adaptivity. The estimation is based on the assumption that the true underlying grey level distribution of each primitive type, i.e. each segmentation class, are normally distributed with a common variance σ. It further assumes that the average mean and standard deviation of the grey level elements in the local windows of size w is given by:

$$\mu_w = D\mu_d + G\mu_g + B\mu_b \tag{3.14}$$

$$\sigma_w^2 = \sigma^2 + D(\mu_d - \mu_w)^2 + G(\mu_g - \mu_w)^2 + B(\mu_b - \mu_w)^2 \tag{3.15}$$

where D, G and B is the true probability for a grey level to be of the dark, grey and bright primitive type, respectively, and μ_d, μ_g and μ_b are the true expectations of the assumed underlying normal distributions of each primitive type. From these assumptions, the study derives explicit expressions for both k_d and k_b. These expressions depends on estimates of D, G, B, μ_d, μ_g, μ_b and *sigma*, which in the study is estimated using the *expectation-maximization (EM)* algorithm, an algorithm which iteratively optimises the parameters as the ML estimates when maximising the expected value of the likelihood function given the current estimates of the parameters (the expected value will indirectly depend on the assumed true value of the missing or hidden data, which here is the true origin of each grey level). [49, pp.26–29,32]

To determine the window size w, the study iteratively tests the possibilities $w = 5$, $w = 7$ and $w = 9$, and chooses the value which obtains the maximum of a criterion function. The criterion function is a modification of the validation step of Yanowitz and Bruckstein's proposed segmentation method [72, p.86]. More precisely, for each w, the input cell image is segmented using Niblack's method with the already estimated values of k_d and k_b. For all 4-connected objects in the segmentation result, the average gradient magnitude of its 4-connected edge pixels are obtained. The gradient magnitude is estimated using the absolute values (the L_1-norm) of the first order derivatives of the input image, where the Sobel operators are used to estimate the first order derivatives [19, pp.166-167]. The chosen window size is the one which attains the maximum average of all its objects average gradient magnitudes. [49, p.20]

To divide connected and remove small dark and bright objects, the study proposes and applies the following morphological algorithm [49, p.40]:

1. Find all 4-connected objects that belongs to either the dark or the bright segment. Dark and bright objects are treated separately in the following, i.e. a single object can not contain both dark and bright elements.

2. For each object where its size is greater than a certain threshold (5 for dark objects, 15 for bright objects) and the *solidity* is less than 0.8 (the solidity is the ratio of the size of the object to the size of the convex hull of the object), attempt to divide the object as follows:

   (a) Erode sequentially with linear structure elements of size two:

   1) $\begin{array}{|c|c|}\hline 1 & 1 \\ \hline \end{array}$, 2) $\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline \end{array}$, 3) $\begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline \end{array}$, 4) $\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$

   If an erosion divides the object (still in the 4-connectivity sense), no further erosion is performed.

   (b) Dilate the object(s) with the same structure elements as it was eroded with (up to four).

   (c) If the previous two steps do not divide the object, repeat the steps by applying the similar four linear structure elements on the original object (i.e. not the result of the previous two steps) when the size of the structure elements are increased by one. The steps are repeated with increasing sizes until the object is divided or has completely vanished.

   (d) If the object is divided, the resulting division objects replace the original object and the complete step 2 is repeated for each of the resulting objects. If the object has vanished, the original object is kept.

3. Apply a morphological opening with a filled, flat 2x2-structure element.

Features

The study extracts five features from each estimated primitive; the area, the area relative to the cell area, the *compactness*, the *eccentricity* and the *orientation relative to the radial direction* [49, p.44]. The compactness is defined as the ratio

of the squared perimeter (8-connectivity is used, the diagonal distance is set to $\sqrt{2}$) to the product of 4π and the area, which makes it a positive value and the compactness of a (true) circle is one. The eccentricity is defined in terms of the ellipse with the same second order central moments as the object and is defined as the ratio of the distance between the two foci of the ellipse to the length of its major axis, which makes it a value in the interval $[0, 1]$ and 0 and 1 for a circle and a line, respectively. Finally, the orientation relative to the radial direction is defined as the angle from the major axis of the ellipse with the same second order central moments as the object to the line that passes through the centre of mass of both the cell image and the object.

The study also extract seven features on the cell level; the area, the compactness, the eccentricity, the mean and variance of grey level, and the number of dark and bright objects in the segmentation [49, p.44]. These features will in the following be referred to as the *cell features*. The five above-mentioned features will be referred to as the *object features*, and each of these features are averaged for each cell image to obtain features on the cell level.

Classification results

The classification results given in the study are inspiring. When including all 134 patients, using the cell as the analytical unit and the prominence to decide the classification the patients, the obtained CCRs were 53.0 %, 71.6 % and 72.4 % for the object features, the cell features and the combination of both the object and cell features. The number of dark and bright objects in the segmentation gave 65.7 % CCR in themselves, and including these features with the object features gave a CCR of 68.7 %, but excluding them from the cell features also increased this CCR, from 71.6 % to 73.1 %. The geometric features (the area, the compactness and the eccentricity, all both on object and cell level, and the object area relative to the cell area) and the radiometric features (the object orientation relative to the radial direction and the mean and variance of grey level) gave a CCR of 70.9 % and 64.9 %, respectively. [49, pp.64–65,71–72]

When excluding tetraploid and polyploid patients, but still using the cell as the analytical unit and the prominence to decide the classification of the patients, the obtained CCRs were 60.8 %, 81.4 % and 81.4 % for the object features, the cell features and the combination of both the object and cell features. When using the patient as the analytical unit with the average cell feature value within each patient as the same patient's feature value, and applying feature selection using the *plus l - take away r* selection method (see section 6.5) and the classification method which assumes normality and equal covariance matrices (see case 2 in section 6.2.1), the obtained CCRs were 61.8 %, 76.5 % and 77.5 % for the three same groups of features, ordered equally. Including the negative adaptive texture feature of the 4D-GLEM property array from the study in [46], which according to [49, p.71] gave a CCR of 81.4 % in itself, all CCRs where significantly increased to 87.3 %, 89.2 % and 89.2 % when still ordered equally and still using the patient as the analytical unit and the mentioned feature selection and classification method. [49, pp.71-73]

Chapter 4

Class specific dual entropy matrices

We will in this chapter propose a novel choice of the property array. The general definition of this property array is based on the assumption of at least two different texture primitive types and attempts to capture some specific contextual information present within the estimated primitives of a specific primitive type. Using the definitions in section 3.2.1, this is a geometrical method, but it also has strong resemblance with the statistical methods in its way of computing the property array (for the specific primitive type).

We will begin with a definition of the proposed property array. Following this is a discussion of how to extract reasonable features from such an array. A description and discussion of the segmentation methods and then the contextual measurement used in this study, two choices the proposed property array depends heavily on, will conclude this chapter.

4.1 Definition

Following the line of Maître et al. [35, pp.212-213] and Tupin et al. [66, p.725], we will characterise every pixel by the following three quantities:

- Its grey level, $g \in \{0, 1, \ldots, G-1\}$, where G is the number of grey levels.
- A class label, $l \in \{0, 1, \ldots, L-1\}$, where L is the number of classes resulting from a recognition process, e.g. an adaptive segmentation method.
- A context value, v. To simplify the following description, we will assume that the context values are discrete, non-negative and with V levels, i.e. $v \in \{0, 1, \ldots, V-1\}$, but we would like to note that it is easy the generalise the method for the case of continuous context values (by requiring the specification of a quantification). We can also note that the context values do not even need to be of textural origin from a methodical point of view, but the intended application of the proposed property array will be based on this assumption.

For all pixels, we gather each of these three values in their respective matrices; the image $A \in \{0, 1, \ldots, G-1\}^{m,n}$ (which is the standard grey level image),

the segmentation image $B \in \{0, 1, \ldots, L-1\}^{m,n}$ and the context value image $C \in \{0, 1, \ldots, V-1\}^{m,n}$. m and n is the height and width, respectively, of all three images.

Let $q(g, l, v)$ be the discrete probability that the combination of grey level g, class label l and context value v occurs in a specific (A, B, C)-tuple. The global Shannon entropy is then:

$$e := -\sum_{g=0}^{G-1} \sum_{l=0}^{L-1} \sum_{v=0}^{V-1} q(g, l, v) \log q(g, l, v) \ , \ q(g, l, v) > 0 \tag{4.1}$$

Using the definition of conditional probabilities and the law of total probability, the class marginal pmf can be written as:

$$q(g, v|l) := \frac{q(g, l, v)}{\sum_{g'=0}^{G-1} \sum_{v'=0}^{V-1} q(g', l, v')} \tag{4.2}$$

when the denominator is positive; if not, we will define $q(g, v|l)$ as zero (this only happens if no pixels have class label l). From this, we see that the class specific global Shannon entropy is given as:

$$e_l := -\sum_{g=0}^{G-1} \sum_{v=0}^{V-1} q(g, v|l) \log q(g, v|l) \ , \ q(g, v|l) > 0 \tag{4.3}$$

We can also derive the class specific grey level histogram $q(g|l)$ and the class specific context histogram $q(v|l)$ by using the law of total probability. The results can be written as:

$$q(g|l) := \sum_{v=0}^{V-1} q(g, v|l) = \frac{\sum_{v=0}^{V-1} q(g, l, v)}{\sum_{g'=0}^{G-1} \sum_{v=0}^{V-1} q(g', l, v)} \tag{4.4}$$

$$q(v|l) := \sum_{g=0}^{G-1} q(g, v|l) = \frac{\sum_{g=0}^{G-1} q(g, l, v)}{\sum_{g=0}^{G-1} \sum_{v'=0}^{V-1} q(g, l, v')} \tag{4.5}$$

where we in both last transitions again assumed the positivity of the denominators (which are equal); it follows from the mentioned special case in the definition of the class marginal pmf $q(g, v|l)$ that both these histograms, $q(g|l)$ and $q(v|l)$, are zero in this case (which only occurs if no pixels have class label l).

From the definitions of the class specific grey level histogram and the class specific context histogram, we obtain the class specific grey level entropy ϵ_l and the class specific spatial entropy ζ_l, respectively, as:

$$\epsilon_l := -\sum_{g=0}^{G-1} q(g|l) \log q(g|l) \ , \ q(g|l) > 0 \tag{4.6}$$

$$\zeta_l := -\sum_{v=0}^{V-1} q(v|l) \log q(v|l) \ , \ q(v|l) > 0 \tag{4.7}$$

The *class specific dual entropy matrix (CSDEM)* of the class $l \in \{0, 1, \ldots, L-1\}$ and the image A with segmentation image B and context value image C, when

using q_G and q_V quantification levels per integer entropy for the class specific grey level and spatial entropy, respectively, is defined as:

$$\delta(r(q_G\epsilon_l), r(q_V\zeta_l)) \tag{4.8}$$

where δ is the Dirac delta function and $r : [0, \infty) \to \mathbb{N}_0$ is any rounding function. The CSDEM is thus a binary matrix with value one only in the pixel $(r(q_G\epsilon_l), r(q_V\zeta_l))$. The CSDEM could optionally be defined with the inclusion of a parameter G_S which can be used to reduce the number of grey levels in the input image A before the class specific grey level entropy is computed.

Given a set of class labels $K \subseteq \{0, 1, \ldots, L-1\}$, the *class specific dual entropy matrices (CSDEMs)* of K and the image A with segmentation image B and context value image C is defined as the set $\{M_1, \ldots, M_{|K|}\}$ where M_l, $\forall l \in K$, is defined as the CSDEM of the class l and the same input images A, B and C. The quantification parameters q_G and q_V should also here be specified, and the definition of the CSDEMs could also optionally be defined to include a grey level quantification parameter G_S.

Because all elements in a CSDEM are zero except a single element (which is one), any CSDEM of an image is extremely sparse in comparison with standard property arrays (like the GLCM) of the same image. The interpretation that a CSDEM estimates the probability of occurrence of each (ϵ_l, ζ_l)-pair in an assumed underlying true distribution of ϵ_l and ζ_l, which is the common interpretation when using the standard property arrays, is therefore bad for any CSDEM of an image.

When using a CSDEM as a property array, it will be computed for each scene in the learning dataset. In correspondence with the general description in section 3.2.2, we know that a scene is not equivalent with an image. The interpretation that a CSDEM of a scene estimates the probability of occurrence of each (ϵ_l, ζ_l)-pair may thus be valid, but only if the property array of the scene is the average of the property array of many subordinate images and also that the number of subordinate images is high relative to the total number of elements in the CSDEM.

4.1.1 Implementation friendly algorithm description

We will here provide a implementation friendly description of the computation of CSDEMs.

Define $t : (\mathbb{R}^{m,n}, \{0, 1\}^{m,n}) \to \mathbb{R}^x$ as the function which extracts the elements of the first input matrix that is labelled one in the binary second input matrix and $e : \mathbb{Z}^n \to [0, \infty)$ as the entropy function which computes the entropy (e.g. the Shannon entropy) of a set of n pixels. Define the input image as $A \in \{0, 1, \ldots, G-1\}^{m,n}$, where m is its height, n is its width and G the number of grey levels. The CSDEMs when using G_S grey levels and q_G and q_V quantification levels per integer entropy for the class specific grey level and spatial entropy, respectively, can then be computed using the following steps:

1. If $G_S < G$, scale A to G_S grey levels.
2. Segment A into L classes using an arbitrary segmentation method. Let the segmented image be denoted as $B \in \{0, 1, \ldots, L-1\}^{m,n}$ where its elements uniquely define the class label resulting from the segmentation. Note that C could also be the result of any pixel-based classification.

3. Define $C \in \{0, 1, \ldots, V-1\}^{m,n}$ as the context value image of B, i.e. the image where each pixel is a measurement of the local contextual information of its neighbourhood in B, and where $V-1$ is the maximum obtainable context value (possibly after applying specific quantification and/or translation). We could expand the definition of C to also or only use the input image A.

4. For each class label $l \in K$, where $K \subseteq \{0, 1, \ldots, L-1\}$ defines the classes that we wish to obtain property arrays for, the CSDEM is defined as:

$$\delta(r(q_G e(t(A, B == l))), r(q_V e(t(C, B == l))))) \tag{4.9}$$

where δ is the Dirac delta function and the notation $B == l$ gives the binary matrix of equal size as B where each element has value one if and only if the corresponding element in B has value l.

To obtain CSDEMs of fixed size within each class, which was mentioned as one of the reasonable criteria of the property arrays in section 3.2.2, we compute the total number of quantification levels of the class specific grey level and spatial entropy, respectively Q_G and Q_V, by applying equation (3.9):

$$Q_G = 1 + \lfloor q_G \log_2(min\{A_l, G_S\}) \rfloor \tag{4.10}$$
$$Q_V = 1 + \lfloor q_V \log_2(min\{A_l, V\}) \rfloor \tag{4.11}$$

where A_l is the number of pixels with class label l in B.

If the range of the context values is unknown, it could in the most comprehensive case be computed as the range of all context values for all scenes in the learning dataset. If some scenes in the validation dataset attains context values outside this range, these contributions could simply be ignored. In fact, if using the set of adaptive texture features described in section 3.2.3, ignoring these contributions does not change the resulting feature values because the estimated discrimination value at these elements in the weight array would be zero for this set of features.

4.2 Extracting reasonable features

The expected structure of a CSDEM is highly dependent on the chosen contextual measurement and also influenced by the segmentation method and the problem at hand. We are also not sure what structure to expect for the relevant choices and in particular where the discrimination value should be located in the property array. In connection with the discussion in section 3.2.4, adaptive texture features are suitable for such situations because they automatically estimates the discrimination value of each element in the property array, which directly result in an estimate of the 'optimal' weight array. We will therefore use such features in this study. More specifically, we intend to use the set of four adaptive texture features described in section 3.2.3 as our feature candidates for each specific type of property array. We should again note that a manual inspection of the designed weight array is needed to interpret and discuss what the resulting feature actually measure. Figure 4.1 shows an example of the designed weight array of a difference adaptive texture feature. Notice that

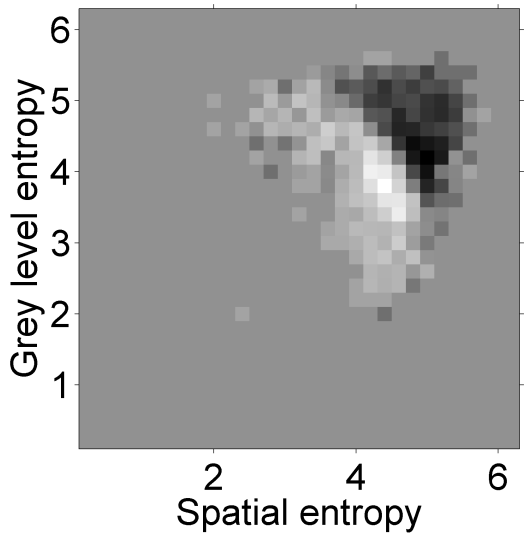

Figure 4.1: An example of the designed weight array of a difference adaptive texture feature. The image is linearly scaled to fill the entire grey level range; the true ranges is $[-1.2, 0.91]$*. This is the same weight array as in the lower right corner in figure 7.8.*

the weight array is a nearly smooth surface expect in the transition to the uniform grey region which corresponds to an estimated weight of 0, which indicates reliable estimations in most relevant elements of the CSDEM.

We noted in the discussion in section 3.2.4 that the performance of the adaptive texture features depends on the number of scenes in the learning dataset and the accuracy of the property arrays with respect to their interpretation. This becomes a critical point when using CSDEMs and adaptive texture features because each image in the learning dataset only contributes with a single occurrence in each CSDEM, thus the number of images in the learning dataset must be very high in comparison with the total number of elements in each CSDEM if the resulting features are to be reasonable.

In our dataset, we have a total of 134 patients and about 300 cell images for each patient, thus about 40000 cell images in total. While this may seem like a large number, it is really not. For comparison we note that if we have a dataset containing only ten images and each image has 4000 pixels (which is representative for our cell images), the average of a standard property array of these images would be about equally good with respect to its interpretation as the average of any CSDEM of all cell images in our dataset, if both average property arrays have equal size. This is because both average property arrays would be the average of about 40000 occurrences. In this perspective, we understand that 40000 cell images is really not much when we wish to extract adaptive texture features from CSDEMs, thus we should be restrictive with the precision of the axes in the CSDEMs and perhaps also look for coarser adaptive texture features, i.e. coarser ways of designing the weight array. Because we

only have a relatively small dataset with respect these features, we note that we should look in particular for the overfitting problem when inspecting the designed weight array of a CSDEM.

4.3 Segmentation

In correspondence with the study described in section 3.2.6, this study is based on the assumption of three different types of primitives which correspond to the dark, grey and bright regions within the cell images. The number of classes, L, is thus three. Features are not extracted from the grey primitives, but both the dark and the bright primitive type is relevant.

We will begin this section with a discussion of some challenges associated with segmenting our cell images. We will continue by discussing the appropriateness of using the gradient magnitude to describe the fitness of a segmentation for our cell images. We will then present the segmentation methods which will be used in this study. These methods do in particular make use of the gradient magnitude to find an 'optimal' segmentation and will also attempt to overcome the probably most important challenge associated with segmenting our cell images. One of the methods inherits much from the segmentation method proposed in [49], but we will also see that we expect both our methods to perform slightly better than the segmentation method used in that study. We will end this section with examples of some segmentations resulting from using the proposed segmentation methods on some cell images and a discussion of these results.

4.3.1 Some segmentation challenges with our cell images

The most prominent segmentation challenge with our cell images is caused by the use of the monolayer imaging technique. We mentioned in section 2.3.1 that the implied projecting of the entire nuclei on the camera's sensor chip in this technique will cause many chromatin structures to partly or completely overlap in the direction of the projection. The segmentation challenge caused by this overlapping is severe and probably the most important challenge associated with segmenting our cell images. Because it has different effects for the more and less condensed chromatin structures, we will discuss the effect separately for these structure types.

Most less condensed chromatin structures will be partly or even completely hidden behind more condensed chromatin structures. This does not only make it very difficult to segment these structures, it also makes it unclear how to determine the fitness of a segmentation of these structures, both manually and automatically. With few indications of how to properly segment these structures, we will in this study assume that they share enough similarities with the more condensed structures to be segmented using the method which is found appropriate for these structures. Because this assumption is questionable in general, we should determine the parameters used in the method independently for each structure type.

While the more condensed chromatin structures may also partly or completely overlap each other and the less condensed chromatin structures, they will never be hidden. This makes it possible to detect these structures in the

cell images. However, we wish to detect separate structures as separate primitives, thus our segmentation method should attempt to analyse the estimated dark primitives to detect and divide overlaps.

Figure 4.2 shows the cell image of a case which was manually selected to be representative for analysing the dark primitives. When viewed at a low spatial resolution, there seem to exist long and non-elliptical dark primitives. However, when viewed at the maximum spatial resolution available, we see that most of the long structures have multiple intensity valleys, i.e. multiple dark valleys within the same dark region. From the discussion above, we assume that each of these dark regions consists of multiple overlapping dark primitives. We have attempted to perform the separation of these dark regions manually in figure 4.3. On the left we have separate most of these dark regions into multiple dark primitives. We see that the separated dark primitives typically have a realistic nearly circular shape. There were however three cases where it was difficult to find a good separation between multiple intensity valleys in a dark region. These cases are indicated on the right in figure 4.3. For such cases, it seems reasonable to assume that some dark primitives are almost completely contained in others or that more than two dark primitives overlap. For the three cases in the representative cell image, it is likely that each region consists of at least three dark primitives. It is unclear what is the best segmentation of such regions, but if the number or size of the dark primitives shall be used, the regions should be separated into multiple dark primitives even if an unquestionable segmentation

Figure 4.2: A cell image which was manually selected to be representative for analysing the dark primitives.

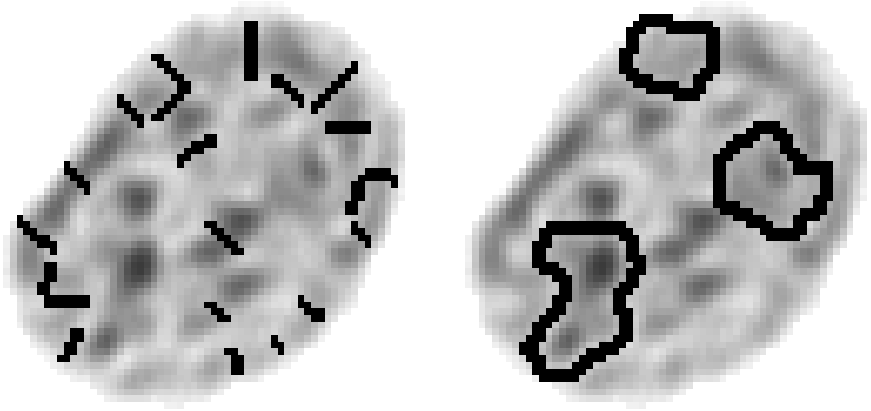

Figure 4.3: Left) a manual separation of most dark regions in figure 4.2 which have multiple intensity valleys, right) highlighting of the three dark regions with multiple intensity valleys which were difficult to separate and likely to each consist of at least three overlapping dark primitives.

can not be found.

While the implied projection in the monolayer imaging technique causes the probably most important challenge associated with segmenting our cell images, there also exist other challenges. In particular, the discretisation causes mainly two challenges. First of all, it limits the spatial resolution of the cell images. This may seem like a vital problem, but its importance decreases with the true amount of overlapping chromatin structures at the point of imaging. As we suspect that this amount is high, we expect the segmentation challenge caused by limited spatial resolution to be relatively small for our cell images in comparison with the segmentation challenge caused by the use of the monolayer imaging technique.

Secondly, the discretisation can also cause the true edges of the chromatin structures, if we assume that these are in truth relatively sharp with respect to our spatial resolution, to be located between different pixels. If we use an edge-based segmentation method, a good edge approximation is important for the result. Because an interpixel edge will be averaged among the neighboring pixels, the intensity of the neighboring pixels may be as low as one fourth of the value attained if the edge perfectly fitted a single pixel. However, an inspection of the cell images reveal that the transitions between chromatin structures are gradual in our cell images, thus the difference between interpixel and intrapixel edges are likely to be small in our case. This makes the edge approximation nearly independent of this discretisation challenge and this challenge to be of minor importance in general.

4.3.2 The appropriateness of the gradient magnitude to describe the fitness of a segmentation

We will now discuss whether it seems appropriate to use the gradient magnitude to describe the fitness of a segmentation for our cell images. Because of

the severity of the overlapping problem for the less condensed chromatin structures, we will only consider the more condensed chromatin structures in this subsection.

Three dark primitives of the representative cell image in figure 4.2 are enlarged in the middle column of figure 4.4 and their gradient magnitude is shown in the right column. These enlargements indicate that the gradient magnitude at any particular point is a reasonable measure of the segmentation value of letting the point be an edge. We also see that the thick edge response, which is a well known property of the gradient magnitude image in standard image processing theory, is positive when we use the gradient magnitude as a measure of the segmentation value because it weights the different edge locations according to their estimated segmentation value. In particular, if using a clean edge detection image with edges of width 1 to measure the segmentation value at each pixel, it will not separate between an edge proposed to be located one, two or more pixels from the edge in the edge detection image; all these locations would have zero estimated segmentation value (when ignoring the effect of other edges).

While the thick edge response is positive, the right column of figure 4.4

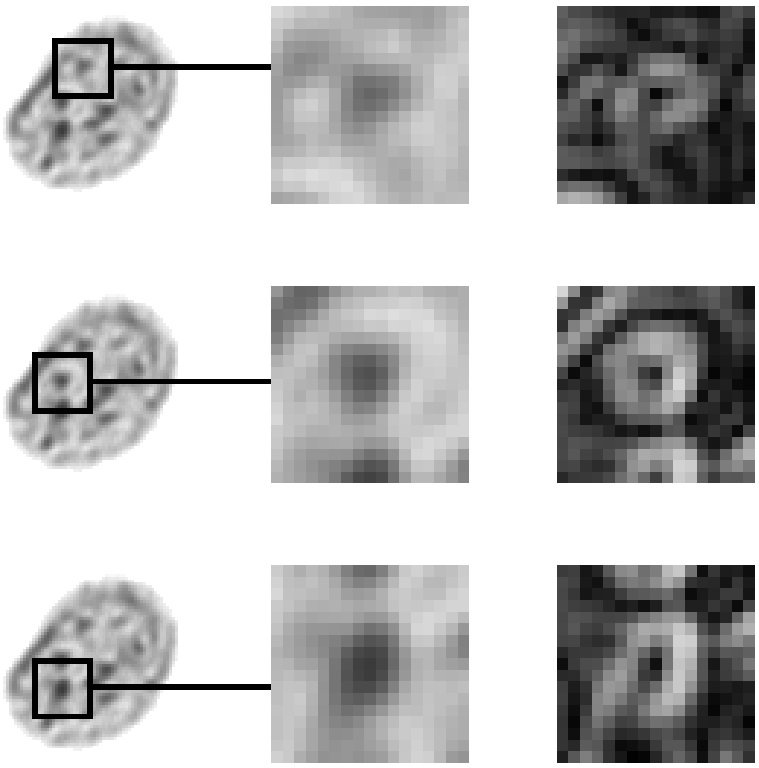

Figure 4.4: The enlargement of three dark primitives of the cell image in figure 4.2 and their gradient magnitude when using the Sobel operators. The gradient magnitude image of the entire cell image is linearly scaled to fill the entire grey level range.

indicates that the strongest edge response often occurs before the intensity valley has completely ascended to a normal level. We therefore risk to on average estimate somewhat smaller objects than what is real if we use the gradient magnitude as a measure of the segmentation value. Because the right column of figure 4.4 also indicates that the strongest edge response do not occur far from the location where the intensity valley has completely ascended to a normal level (approximately), and the edge response is also not much smaller at the location where the intensity has reached the normal level, we still expect that the gradient magnitude is a reasonable measure of the segmentation value of a proposed edge.

A perhaps more severe drawback with using the gradient magnitude as the measure of the segmentation value is that it is negatively affected by the overlapping problem; the gradient magnitude is typically low between multiple primitives of the same type if they overlap. Of course, this challenge will also be a problem for many other measures of the segmentation value, but more adapted measures may tackle this problem better than the gradient magnitude does. The enlargement and gradient magnitude in the last row of figure 4.4 indicates this problem for a case where the overlap between the dark primitives is small, but present. If we apply the gradient magnitude as our measure of the segmentation value of a proposed edge, we can thus not expect it to separate all overlapping primitives, which makes it necessary to add another step to the segmentation method for this purpose.

Because the study in [49] uses the L_1-estimated gradient magnitude as the measure of the segmentation value of a proposed edge, we will briefly discuss whether the use of this estimate instead of the standard gradient magnitude can be justified. The L_1-estimated gradient magnitude differs from the standard gradient magnitude in that its value is the L_1-norm of the estimated first order derivatives and not the Euclidean norm (L_2-norm) which is used in the definition of the gradient magnitude. The advantage of using the L_1-estimated gradient magnitude instead of the standard gradient magnitude is that it is computationally less expensive and also allows the use of precise integer arithmetic.

A normal distribution with $\Sigma = 6I_2$ and its standard and L_1-estimated gradient magnitude is shown in figure 4.5. It is from this figure evident that the L_1-estimated gradient magnitude is not rotation invariant like the standard gradient magnitude. More precisely, it is quite clear from this figure that the L_1-estimated gradient magnitude overestimates the gradient magnitude near the diagonal directions relative to the gradient magnitude near the horizontal and vertical directions. It is easy to show that the L_1-norm and the Euclidean norm will in general be equal along the horizontal and vertical directions, but the L_1-norm is $100(1-\sqrt{2})\% \approx 41.42\%$ larger than the Euclidean norm along diagonal directions. The L_1-estimated gradient magnitude thus emphasise diagonal intensity changes significantly more than the standard gradient magnitude. We therefore conclude that it does not seem to be justifiable to use the L_1-estimated gradient magnitude instead of the standard gradient magnitude because of the slightly decreased computational burden and the possibility of using precise integer arithmetic.

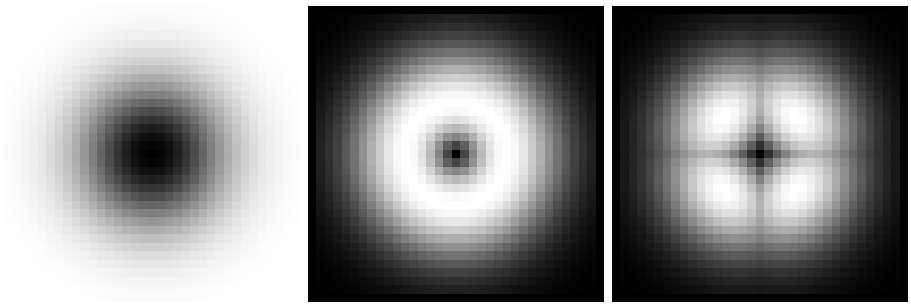

Figure 4.5: Left) a normal distribution with $\Sigma = 6I_2$ (measured in pixels), middle) its gradient magnitude, right) its L_1-estimated gradient magnitude. The first order derivatives is in both cases estimated using the Sobel operators. The gradient magnitude images are independently linearly scaled to fill the entire grey level range.

4.3.3 The method

While the discussion above indicates both a way to measure the fitness of a segmentation and that a step which attempts to separate overlapping primitives should be included in the segmentation method, it gives few indications of how to produce the initial set of segmentations for a specific cell image. This choice is naturally also important. In particular, it is important that at least one proper segmentation is included in this set (we hope that the segmentation fitness measure identifies which segmentation this is).

Because of the typically uneven physical size of the nucleus in the direction of projection, the segmentation method should analyse the cell images as if they were unevenly illuminated. We should thus apply a locally adaptive segmentation method. There are however still many relevant alternatives.

To restrict the extent of this thesis and because of the promising classification results in the study in [49, p.73] which uses the same dataset as ours, we choose to use the extension of Niblack's method to include two thresholds, see equations (3.11)-(3.13), because this is the basic segmentation method which is used in that study [49, p.25]. The use of this basic segmentation method will however differ. In particular, the set of parameters which Niblack's method heavily depends on, will in this study be iteratively optimised for each input image. This differs from the usage in the mentioned study because the optimisation of k_d and k_b is not performed iteratively in that study, it is instead estimated under some assumptions on the distributions of the primitive types, see section 3.2.6. Also, but less importantly, while a modification of the validation step of Yanowitz and Bruckstein's segmentation method [72, p.86] will be used as the criterion function for the optimisation both here and in the mentioned study, the modification is slightly different.

For each cell image, the initial segmentation is computed as follows:

1. Find the gradient magnitude of the cell image. In correspondence with the discussion in section 4.3.2, we will use the true gradient magnitude, i.e. the L_2-norm of the estimated first order derivatives along two orthogonal direction, and not an estimate like the L_1-estimated gradient magnitude.

The first order derivatives are estimated using the Sobel operators:

1	0	-1
2	0	-2
1	0	-1

,

-1	-2	-1
0	0	0
1	2	1

2. For each triplet of window size w and uncertainty parameters k_d and k_b, compute a segmentation using Niblack's method. Since the nucleus does not occupy the entire cell images, the local windows at the periphery of the nucleus may contain pixels which are not cell pixels. In such cases, these pixels are simply ignored when computing the local expectation and standard deviation which is used in Niblack's method.

   The set of possible values for the window size w is set to $\{5, 7, 9\}$. Both k_d and k_b is allowed to take values from the set $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. These three sets of values are founded on empirical testing on several hundred cell images in our dataset. The uncertainty parameters are set to be independent as was suggested in section 4.3.1. If these possible values for either uncertainty parameter are to high to result in any objects within the corresponding class, then the value of the relevant uncertainty parameter is decreased by 0.1 until the segmentation result contain at least one object in the corresponding class.

   For each segmentation, compute the criterion function of the segmentation as the average of the following criterion value computed separately for the dark and bright primitive type:

   (a) Find all 4-connected edge pixels of the objects, i.e. all object pixels which are 4-connected to a non-object pixel.

   (b) Compute the average gradient magnitude of all 4-connected edge pixels of the objects. The average of the average gradient magnitude of all 4-connected edge pixels of each object is not used because this weights small and large object equally, thus decreasing the influence of each edge pixel in large objects relative to each edge pixel of small objects. The gradient magnitude of an edge pixel is computed as the average gradient magnitude of the object pixel and all its 4-connected non-object pixels. This average is used instead of the gradient magnitude of the object pixel in itself because the edge of an object is typically better modelled to be between the object edge and its 4-connected surroundings than on the object edge in itself.

   We see that the criterion value computation above, and thus also the entire criterion function, assumes that the gradient magnitude is a reasonable measure of the segmentation value of a proposed edge, an assumption which was justified in section 4.3.2. From the discussion in that section, we also understand that a high value of the criterion function proposed above will indicate that the objects are well separated from their surroundings. We therefore selection the segmentation which maximises this criterion function as the initial segmentation of the cell image.

The worst case complexity of this algorithm is limited by the number of cell pixels, which is the worst case complexity of Niblack's method when using

cumulative matrices. For arbitrary possible value sets of w, k_d and k_b, this worst case complexity must be multiplied with the cardinality of w's set and the maximum cardinality of k_d and k_b's set. The worst case complexity does not need to be multiplied with the cardinality of both k_d and k_b's set because the criterion function in step 2 above can be optimised separately for k_d and k_b.

The initial segmentation obtained by the algorithm above is optimal with respect to a reasonable criterion function and set of segmentation candidates. It has however some potential flaws. First of all, most of the periphery of the nucleus is likely to be classified as bright primitives when using Niblack's method. This is because the projection of the nucleus is typically relatively small in these regions, which makes the intensity in these pixels typically higher than the average in a local window which also includes pixels further away from the edge of the nucleus. Because it is difficult to separate falsely estimated bright primitives in such regions from the real ones, we will simply set all pixels sufficiently close to the edge with bright class label to the grey class label. More precisely, we morphologically erode the *mask image* of the nucleus, i.e. the binary image where each pixel is one if and only if it is a cell pixel, by the circular 7x7-structure element (see table 4.1) and set all pixels with bright class label which are in the mask image of the nucleus, but not in the erosion result, to the grey class label. The initial segmentation of the representative cell image in figure 4.2 and the result after setting pixels with bright class label that are sufficiently close to the edge to the grey class label is shown on the left and right in figure 4.6, respectively.

Secondly, we see from the discussion in section 4.3.2 that a criterion function which bases the segmentation fitness on the gradient magnitude, which our proposed criterion function do, can not be expect to separate all overlapping primitives. The problem with overlapping primitives is thus likely to be present in the obtained initial segmentation, thus we should investigate the estimated primitives to see whether some should be divided. Such investigation is, as mentioned in section 4.3.1, especially important if the number or size of the primitives shall be used, which they both will be in this study.

To attempt to separate overlapping primitives, we will apply two different algorithms. The first is similar to the morphological algorithm that was proposed in [49, p.40], see section 3.2.6. The second is based on our analysis of the dark regions in section 4.3.1, where we observed that the dark regions with atypical shape (in a representative cell image) had multiple intensity valley. This algorithm will use the watershed transformation to separate overlapping primitives. Both algorithms will also attempt to remove small objects.

Table 4.1: The circular 7x7-structure element.

0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0

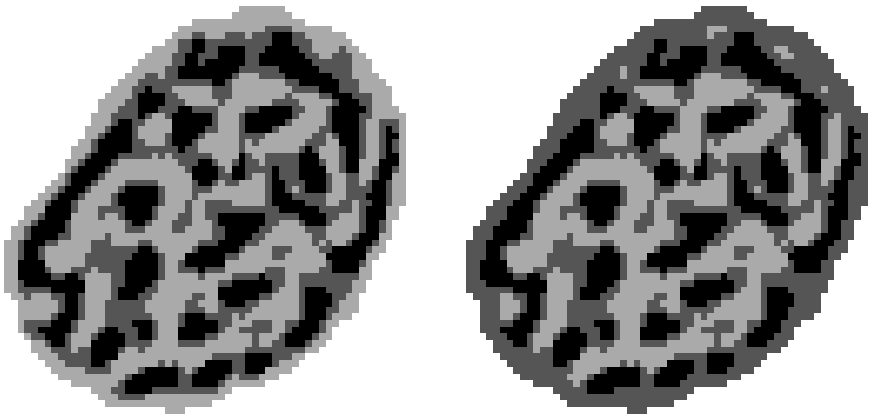

Figure 4.6: Left) the initial segmentation of the representative cell image in figure 4.2, right) the result after the setting pixels with bright class label that are sufficiently close to the edge to the grey class label.

Morphological algorithm

We can formulate the morphological algorithm used in this study to separate overlapping primitives and remove small objects from a segmentation as follows:

1. Find all 4-connected objects that belongs to either the dark or the bright segment. Dark and bright objects are treated separately in the following, i.e. a single object can not contain both dark and bright elements.

2. For each object where its size is greater than a certain threshold (5 for dark objects, 15 for bright objects) and the solidity is less than 0.8, attempt to divide the object as follows:

   (a) Open (erode and then dilate) the object using a linear structure element until one that divides (still in the 4-connectivity sense) the object is found. The set of possible linear structure elements of size two are:

   1) $\begin{array}{|c|c|}\hline 1 & 1 \\ \hline \end{array}$, 2) $\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline \end{array}$, 3) $\begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline \end{array}$, 4) $\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$

   If none of these structure elements divide the object, then the size of the linear structure elements above is increased by one and the opening continues with these structure elements. This is repeated until a linear structure element which divides the object (when used to morphologically open the object) is found. Note that this step differs from the algorithm proposed in [49, p.40] in that it only uses a single structure element to divide the objects.

   (b) If no linear structure element divided the object, then the original object is kept. If a dividing linear structure element was found, then the first such structure element is used to divide the object and the

original object in the segmentation is replaced by the resulting divided objects. The complete step 2 is then repeated for each of the resulting divided objects.

The threshold values in this step are the same threshold values that were used in the study in [49]. They can be founded in the discussion in section 4.3.1. In particular, the use of solidity can be based on the observation that the true structure of the primitives seem to be nearly circular, an observation which was made in connection with the manually separated dark primitives on the left in figure 4.3. Also, the higher size threshold for the bright objects indicates that a bright primitive with non-circular shape can be real if it is small enough, an assumption which can be based on the effect of a partial hiding of a less condensed chromatin structure. We could also note that the difference in the size threshold is in correspondence with the suggestion in section 4.3.1 of determining the method parameters independently for each primitive type.

3. Apply a morphological opening with a filled, flat 2x2-structure element.

Algorithm based on the watershed transformation

We can formulate the watershed transformation algorithm used in this study to separate overlapping primitives and remove small objects from a segmentation as follows:

1. Find all 4-connected objects that belongs to either the dark or the bright segment. Dark and bright objects are treated separately in the following, i.e. a single object can not contain both dark and bright elements.

2. For each object, apply the following algorithm which divides the object if it contains multiple intensity valleys (for dark regions) or intensity peaks (for bright regions):

   (a) Extract the grey level intensities of the object pixels from the corresponding cell image. We will call the result for the *grey level object*.

   (b) To reduce the risk of over-fragmentation, i.e. segmentations with an unnaturally large amount of objects, the grey level object is averaged by convolving with the following filter:

0	1	0
1	2	1
0	1	0

   Note that pixels outside the object is ignored, e.g. the average of a pixel with no neighbours to the left and above is the average of the intensity of that pixel and the average of the intensities of its two 4-connected neighbours. This is important as making use of neighbours outside the object can create new intensity valleys (for dark regions) or peaks (for bright regions) within the object because edge-pixels will be relatively more affected than the other pixels, thus contradicting our aim of reducing the risk of over-fragmentation. Moreover, the averaging is small because of the low spatial resolution of the cell

images and the filter is centre-peaked to avoid the creation of new intensity valleys (for dark regions) or peaks (for bright regions). The latter problem can be understood by convolving the following grey level object (left) with the proposed centre-peaked filter (centre) and the corresponding flat filter (right):

-	2	-
2	1	2
2	1	2
-	1	-

,

-	1.67	-
1.75	1.50	1.75
1.75	1.50	1.75
-	1.67	-

,

-	1.50	-
1.67	1.60	1.67
1.67	1.60	1.67
-	1.50	-

Notice how the flat filter removes the true intensity valley (the two pixels with intensity 1) and creates two new intensity valleys, one at the top and one at the bottom, while the proposed centre-peaked filter do not change the intensity valley.

(c) Apply the watershed transform on the grey level object (for dark regions) or the inverse of the grey level object (for the bright regions). The pixels of the watershed lines, which are the pixels that do not belong to a single intensity valley, is used to separate the original object. We will apply 8-connectivity in the watershed transform to reduce the risk of over-fragmentation.

3. Remove all objects with a size of less than 5.

4.3.4 Some segmentation results

The result of applying the proposed segmentation method with the watershed transformation algorithm and with the morphological algorithm on the representative cell image in figure 4.2 is shown on the left and centre in figure 4.7, respectively. These segmentations are remarkably good in light of the manual separation of the dark primitives shown on the left in figure 4.3. Indeed, all dark primitives which are indicated in the manual separation are also present in both segmentations with the exception of a single dark primitive located at the upper right corner of the nucleus which is removed by the morphological algorithm. We also see that the watershed transformation algorithm separates the dark region along the right edge of the nucleus to one more dark primitive than our manual separation and the other segmentation did. While it is difficult to determine whether this is good or bad, we simply state that it indicates that the approach which uses the watershed transformation will result in slightly more primitives than the morphological approach.

The most prominent difference between the segmentations on the left and centre in figure 4.7 is that the morphological approach results in smaller primitives. This difference is mainly a result of the morphological opening used to separate overlapping primitives and remove small objects. Because the natural separation of primitives will in this aspect result in a segmentation similar to the one which uses the watershed transformation, we acknowledge the size of the primitives as a weakness with the morphological approach. We can in this context note that a potential problem that was mentioned in section 4.3.2 with using the gradient magnitude as a measure of segmentation fitness, that the strongest edge response may occur before the intensity valley has completely

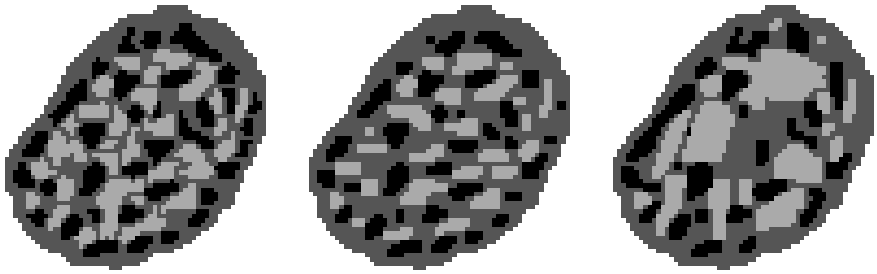

Figure 4.7: Left and centre) the segmentation of the representative cell image in figure 4.2 using the proposed segmentation method with the watershed transformation algorithm and with the morphological algorithm, respectively, right) corresponding segmentation using the segmentation method used in [49].

ascended to a normal level, does not seem to be an issue for our segmentation methods. Tests indicate that it would have been a problem if we allowed the window size or the relevant uncertainty parameters to attain much higher values; the resulting segmentations would in this case often only include the most prominent region of the few most prominent primitives.

It is also interesting to see how the proposed segmentation methods handle the three dark regions where it was difficult to find a good separation between multiple intensity valleys, see the right image in figure 4.3. In the manual inspection, we suspected that each of these regions were likely to consist of at least three dark primitives. Comparing with the segmentation result on the left and centre in figure 4.7, we see that each region has been quite reasonably separated into three or two dark primitives. Our segmentation methods thus seem to be handling all forms of overlapping dark primitives reasonably.

We also note that our two separation approaches handle the bright regions rather differently. The morphological approach is especially coarse for these regions (compare with the initial segmentation in figure 4.6); it splits and removes much of the bright regions. The watershed transformation approach is much more detailed and results in more bright primitives, but it may also suffer more severely from the overlapping problem with the monolayer imaging technique. With few indications of what is a proper segmentation of these regions, it would be foolish to attempt to make a conclusion as to which approach is the best here. Instead, we should evaluate both segmentation methods and on that basis attempt to make a conclusion of which approach is better for our classification problem.

The segmentation resulting from using the segmentation method used in [49] are shown on the right in figure 4.7. In comparison with the manual separation and the segmentations when using our segmentation methods, this segmentation is not that good. In particular, it lacks multiple dark primitives and also fails to separate overlapping dark primitives. It also clearly overestimates the size of most bright primitives, in fact, this overestimation is quite severe in the segmentation on the right in figure 4.7. This overestimation also implies that the number of bright primitives is less than what this number would have been in a more natural segmentation. This comparison does in total indicate that both our

segmentation methods may be significantly better than this method. We note that maybe the most important difference, the more accurate estimation of the dark primitives, is mainly a result of optimising k_d and k_b in our segmentation methods in comparison with the estimation of these values which is performed in this method (see section 3.2.6).

The segmentations resulting from using our segmentation methods on some other cell images are shown in the second and third column in figure 4.8. These segmentations substantiate the already stated difference. They both result in reasonable segmentation. In particular, the shape of the estimated primitives by the morphological approach seems to be more natural, but the approach also removes possibly valuable information.

The result of applying the segmentation method used in [49] are included for comparison in the fourth column in figure 4.8. From these segmentations we see a significant resemblance between our segmentation method with the morphological approach and the method used in [49], especially with respect to the estimation of the dark primitives. A closer inspection do however reveal some difference. In particular, the already stated claims are substantiated; our methods estimates the dark primitives slightly more accurately (save maybe the cell image in the fourth row) and the method used in [49] tends to overestimate of the size of many bright primitives (and thus indirectly underestimate the number of bright primitives). From an inspection of the segmentations of other cell images, we have verified that these are representative differences in general for our dataset. We also note that the other method completely fails to detect any dark regions in the first cell image. The inspection of the segmentations of the other cell images reveal that this is an exception, but it is far from rare. If we relax the claim to say that the other segmentation method sometimes fails to estimate the vast majority of the dark primitives, i.e. fails to label the pixels in these regions as the assumed true dark class, then this flaw with the other method is too frequent to be called an exception.

In conclusion, our proposed segmentation methods seems to reasonably segment the cell images in our dataset. In comparison with the segmentation method used in [49], we believe that our methods are generally slightly better and also do not fail to estimate the vast majority of the dark primitives in some cases (which the other method does). We thus expect that our segmentation methods are slightly better in general.

4.4 Contextual measurement

For a general dataset, there are many contextual measurements of interest. Traditionally, the frequency and orientation has been much used in texture analysis [65, p.6]. Such use can be motived by the study of Knutsson and Granlund [27, pp.1,8] which mentions evidence that the decomposition into spatial frequencies is an essential component of the human visual system and also shows that it serves well as a basis for texture discrimination. While we claimed in section 2.3 that our discrimination task is subvisual in general, a generalised use of the frequency and orientation may still be appropriate. In particular, Nielsen et al. [40, p.3] discovered a connection between carcinogenesis and frequent occurrence of the more condensed chromatin structures at the periphery of the nucleus, thus some measurements of the frequency of the primitives and

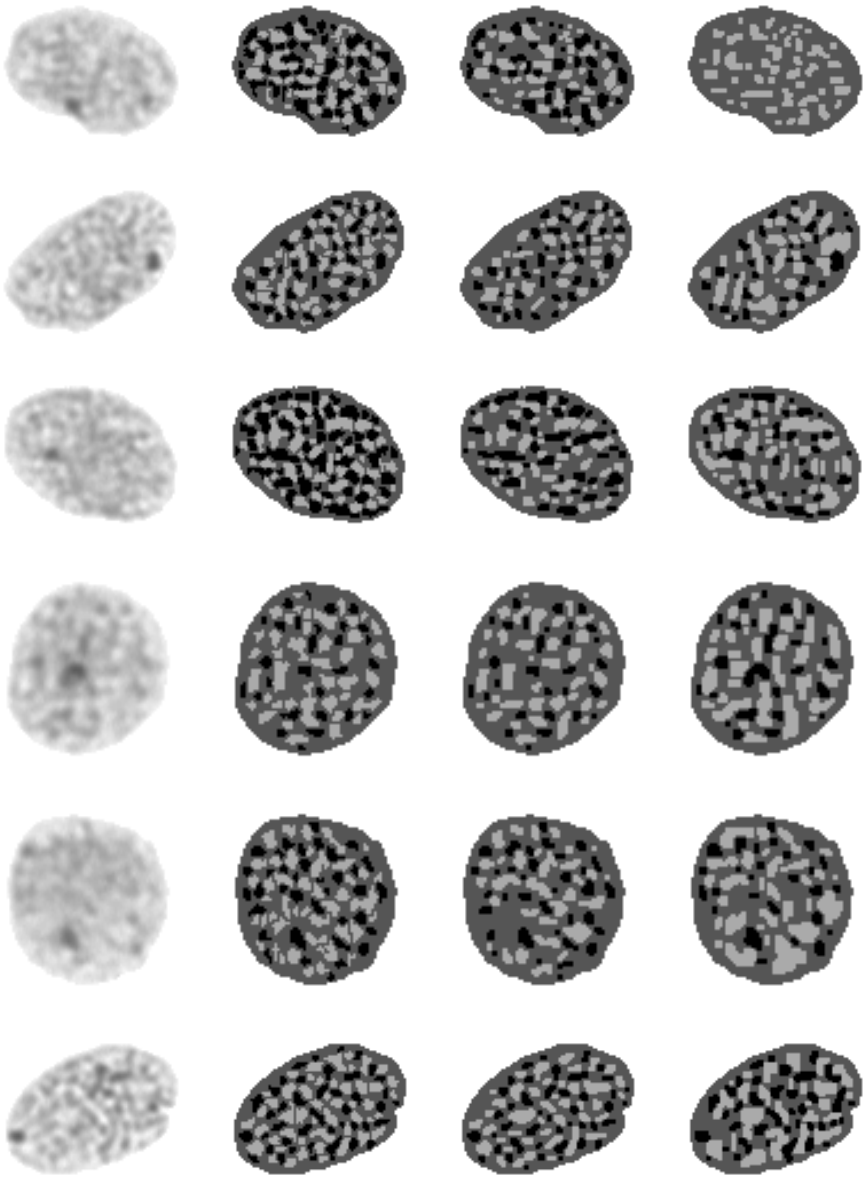

Figure 4.8: First column) cell images, second and third column) corresponding segmentation using the proposed segmentation method with the watershed transformation algorithm and with the morphological algorithm, respectively, fourth column) corresponding segmentation using the segmentation method used in [49].

their relative orientation may be appropriate contextual measurements.

For our dataset, detailed contextual measurements are likely to suffer significantly because of the monolayer imaging technique and low spatial resolution. We will because of this, and to restrict the extent of this thesis, limit ourself to a simple contextual measurement; the object size. This contextual measurement is motived by the study by Danielsen [8, p.40] which observed an increase in the size of condensed chromatin structures during carcinogenesis. The study also observed an increase in the number of structures, an increase which will be indirectly included in the CSDEMs of the object size because the maximum spatial entropy will then increases with the number of structures, thus the expected spatial entropy is also likely to increase with the number of structures.

Chapter 5

Features

This chapter will provide an overview of the features that will be evaluated in this study. As most features are already sufficiently discussed, we will not provide a discussion of these features in this chapter.

5.1 Cell features

The following five features will be referred to as the *cell features*:

- *Area*: Average cell area (average number of cell pixels in the cell images of a specific patient).
- *Compactness*: Average compactness using 8-connectivity and $\sqrt{2}$ diagonal distance (see definition in section 3.2.6).
- *Eccentricity*: Average eccentricity (see definition in section 3.2.6).
- *GreyLevelAverage*: Average of the average grey level of the cell pixels.
- *GreyLevelVariance*: Average of the variance of the grey level of the cell pixels.

These features are non-adaptive features of the nucleus that are independent of a segmentation. Note that this definition of the cell features is different from the definition in [49, p.44] in two ways; it assumes that the patient is the analytical unit and does not include the number of dark and bright objects.

5.2 NO-features

The following two features depend on a segmentation and describes the average number of estimated primitives within each primitive type. We will refer to these features as the *NO-features*, and these are:

- *NumberOfDarkObjects*: Average number of dark objects (the estimated dark primitives).
- *NumberOfBrightObjects*: Average number of bright objects (the estimated bright primitives).

Because these features depend on a segmentation, we will specify the used segmentation whenever these features are applied.

5.3 Adaptive texture features

The following list contains six choices for the property array. For each of these choices, we will always evaluate all four adaptive texture features in the set described in section 3.2.3. Typically, only the best of the four classification results will be provided, along with a specification of which adaptive texture feature this result corresponds to. Also, all these features includes a scaling of the cell images to fewer grey levels and the computation of an entropy. As mentioned, we will in this study always use linear scaling and the Shannon entropy with binary base for these choices.

The six choices for the property arrays, which results in 24 adaptive texture features, are:

- *GLEM*: The average of the GLEM computed from the cell images of a specific patient (see definition in section 3.2.5). The number of grey levels is set to 64, the window size is set to 9 and the number of quantification levels per integer entropy value is set to 10. The cell images are grouped according to the cell area and only the cell area groups $[2000, 2999]$, $[3000, 3999]$ and $[4000, 4999]$ are used, which is in correspondence with the findings in [48, p.94]. All these parameter choices[1] are the same as in the study by Nielsen et al. [46]. It is worth mentioning that the same study also shows that the choice of number of grey levels and the window size is insignificant for our dataset (evaluated choices were $G_S = \{16, 32, 64, \ldots, 1024\}$ and $w = \{3, 5, 7, \ldots, 31\}$).

- *GLEM4D*: The average of the 4D-GLEM computed from the cell images of a specific patient (see definition in section 3.2.5). The number of grey levels is set to 64, the window sizes are set to $\{3, 5, 7, 9, 11, 13\}$, the cell area groups are set to $\{[1, 999], [1000, 1999], \ldots, [9000, 9999], [10000, \infty)\}$ and the number of quantification levels per integer entropy value is set to 10.

- *CSDEMdark* and *CSDEMbright*: The average of the CSDEM of the object size of the dark and bright primitive type, respectively, computed from the cell images of a specific patient (see definition in section 4.1). The number of grey levels is set to 64 and the number of quantification levels per integer entropy value is set to 5 for both the grey level entropy and the spatial entropy. The set of containing the CSDEMdark- and the CSDEMbright-feature will be called the *CSDEM-features*.

- *CSDEMsumDark* and *CSDEMsumBright*: The average of the CSDEM sum histogram of the object size of the dark and bright primitive type, respectively, computed from the cell images of a specific patient. The sum histogram, which was proposed by Unser [68], is the sum of the two axes in the matrix, here the sum of the two entropy values of the CSDEM. The

[1] The chosen quantification was selected using a different approach in [46], but resulted in approximately the same quantification.

number of grey levels is set to 64 and the number of quantification levels per integer entropy value is set to 5 for both the grey level entropy and the spatial entropy. The set of containing the CSDEMsumDark- and the CSDEMsumBright-feature will be called the *CSDEMsum-features*.

As the CSDEM- and the CSDEMsum-features depend on a segmentation, we will for these features specify the used segmentation whenever they are applied.

Chapter 6

Classification and evaluation

In image analysis, the goal is either *supervised*, *unsupervised* or *reinforcement learning* [13, pp.16–17]. In *supervised learning*, the membership of each pattern is *a priori* known to belong to one of a specific set of categories called *classes*. The natural goal is then to correctly classify each new, unknown pattern to one of the specified classes, or, in some cases, conclude that this particular pattern could not be classified.

In *unsupervised learning*, which is sometimes called *clustering*, the patterns are not labelled with their category membership. Typically, the number of categories (*clusters*) is also unknown and will depend on the particular application. As an example, consider the scatter plot of the set of two-dimensional feature vectors in the unsupervised learning problem in figure 6.1. Even under the assumption that the feature vectors represents the underlying patterns well, it is not clear whether these patterns should be clustered into two or three clusters and the desired number of clusters may depend on the application.

Lastly, in *reinforcement learning* or *learning with a critic*, we do not *a priori* known the membership of each pattern, but each tentative category is responded, typically with either *correct* or *incorrect* for image analysis problems [13, p.17]. Such problems may be viewed as a hybrid between supervised and unsupervised learning because the category of each pattern is not known, though it in a way do exist.

As we for the problem under study in this thesis *a priori* know the category of each pattern, which is the outcome of each patient, we will in the following only study the case of supervised learning. The classification rule will thus be designed using both the class memberships and the values of a set of features that have been extracted from each pattern. To estimate the performance of any designed classifier, i.e. its ability to classify novel patterns, we need to apply the classification rule to a set of patterns. More precisely, for each pattern in a set of patterns, we will extract the values of the same set of features as those who where used to design the classification rule, estimate the class of this pattern by feeding its feature values into the classification rule and finally compare this estimate with the true class of this pattern. To make the estimated performance reliable, we should at least have an empty intersection between the patterns used to design and those used to evaluate the classifier, see section 6.6 for a more precise discussion. The dataset used to design the classification rule is called the *learning dataset* or the *training dataset*, while the dataset used to evaluate the

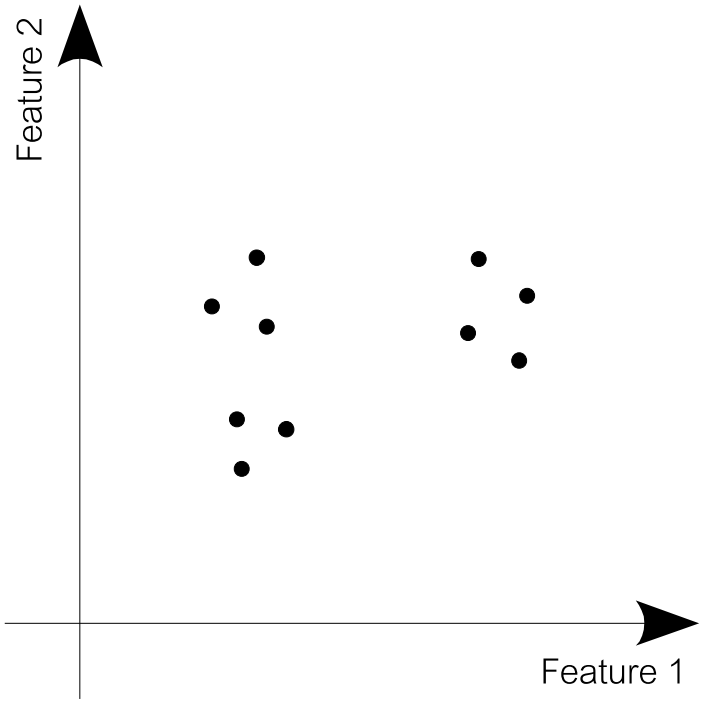

Figure 6.1: Scatter plot of the learning dataset in an unsupervised classification problem where the true number of clusters depends on the particular application.

performance of the classifier is called the *validation dataset* or the *test dataset*.

The chapter begins in section 6.1 with some general definitions about fundamental terminology and quantities that will be applied in the following discussions. Then the set of classifiers most frequently used in supervised learning problems are discussed in a general context in section 6.2. Other classifiers, e.g. those based on optimisation, are not discussed in order to limit the extent of this thesis. We instead continue with a general discussion of the problems with complex classifiers and many feature candidates in section 6.3, which both may cause *overfitting*. Two commonly used techniques to reducing the risk of overfitting, *dimension reduction* and *feature selection*, follows in section 6.4 and 6.5, respectively. Section 6.6 contains a discussion of the methods and some challenges associated with the evaluation of a classifier, including how the total number of patterns should be appropriately divided into a learning and a validation dataset. We will conclude the chapter in section 6.7 with a description of the classification procedures applied in this thesis.

Most of the content of this chapter is based on the textbook by Duda et al. [13] and the paper by Raudys and Jain [56], in particular section 6.1, 6.2, 6.4 and 6.6. The discussion found in this chapter is more detailed than the one found in these two sources and is also based on several other sources.

6.1 Definitions

In order to be able to provide a precise discussion of the methods and challenges with classification and evaluation, we will start off by defining some fundamental terminology and quantities.

Define $\Omega : \{\omega_i | i = 1, \ldots, c\} \to [0, 1]$ as the discrete random variable giving the true *a priori* probability for a novel pattern to belong to each of the c possible classes, $\omega_1 \ldots \omega_c$, e.g. each of the two prognosis classes. Also, uniquely index each element in the set of features with an integer in the interval $[1, \ldots, d]$, where d is the cardinality of the set of features. Let the set of feature values for a single pattern, e.g. for a single patient, be denoted as the d-dimensional column vector $\vec{x} = (x_1, \ldots, x_d)$, ordered according to the indexing of the feature set, and be known as the *feature vector* of that pattern. The set of all possible feature vectors is known as the *feature space* and is denoted as $\mathfrak{D}$. Moreover, let n_L be the number of d-dimensional vectors in the learning dataset, n_V the corresponding number in the validation dataset and $n := n_L + n_V$ be the total number of patterns. Lastly, define n_i as the number of feature vectors in class ω_i of the learning dataset, $\vec{x}_{i,1}, \ldots, \vec{x}_{i,n_i}$ as the feature vectors of these patterns, $\mathfrak{X}_\mathfrak{i} := \{\vec{x}_{i,1}, \ldots, \vec{x}_{i,n_i}\}$, $i = 1, \ldots, c$, as the set of these feature vectors and $\mathfrak{X} := \mathfrak{X}_\mathfrak{1} + \cdots + \mathfrak{X}_\mathfrak{c}$ as the collection of all feature vectors in the learning dataset.

Every classifier has a *decision rule*, which is the concrete rule the classifier bases its classification (or decision) on. The effect of the decision rule is to divide the feature space into c *decision regions*, $\mathfrak{R}_1, \ldots, \mathfrak{R}_c$, each decision region $\mathfrak{R}_i$ corresponds to the set of possible feature vectors for which the classifier in question classifies the corresponding pattern into class ω_i. The *decision boundary* of class ω_i is the piecewise continuous boundary which separates $\mathfrak{R}_i$ from all other decision regions. Note that there exists only a single decision boundary for the case of two classes. The *complexity of a classifier* is here defined as the number of independent parameters the classifier estimates. If a classifier's decision boundaries are of a specific polynomial degree, the complexity of the classifier is positively correlated with the product of this polynomial degree, the number of features and the number of classes in this classifier.

One of the most useful ways of representing the classifier is in terms of a set of *discriminant functions*, $g_i(\vec{x})$ for $i = 1, \ldots, c$ [13, p.29]. These functions may be viewed as the relative evidence that a specific pattern belongs to each possible class with respect to the corresponding classifier. A general decision rule can be expressed in terms of these functions as follows:

Decision rule 1 (General discriminant functions)**.** *For all $\vec{x} \in \mathfrak{D}$, classify the pattern to the class ω_i for which $g_i(\vec{x}) \geq g_j(\vec{x})$ for all $j = 1, \ldots, c$ (if there are multiple choices for ω_i, any will do).*

The decision boundaries of a classifier represented using discriminant functions are the set of feature vectors for which the classifier is satisfied with multiple choices for ω_i. More precisely, the decision boundaries are the set of feature vectors for which there exists at least two unique indexes, i and j with $i \neq j$, such that $g_i(\vec{x}) = g_j(\vec{x}) \geq g_k(\vec{x})$ for all $k = 1, \ldots, c$.

If the number of classes is only two, as is the case for cancer prognosis, the formulation can be simplified by defining $g(\vec{x}) := g_1(\vec{x}) - g_2(\vec{x})$. The classifier, which is sometimes called a *dichotomiser* in the case of only two classes [13, p.30], can now be expressed in terms of $g(\vec{x})$ as follows:

Decision rule 2 (Dichotomiser using discriminant functions)**.** *For all $\vec{x} \in \mathfrak{D}$, classify the pattern to ω_1 if (and only if) $g(\vec{x}) > 0$, otherwise classify the pattern to ω_2.*

It could be noted that any class could have been chosen at the decision boundary, which now can be simply defined as the set of all feature vectors for which $g(\vec{x}) = 0$, but we have in the rule above made the decision that these boundary points should be classified to ω_2.

While many classifiers can be represented in the terms of discriminant functions, the representation is not unique in the sense that there will always exists multiple sets of discriminant functions that classifies all patterns equally. To relate sets of discriminant functions which are equivalent with respect to classification, we will define a equivalence relation (see [22, pp.194,214]), denoted $\equiv$, on the set of discriminant functions that associates sets of discriminant functions which for all $\vec{x} \in \mathfrak{D}$ have either equal unique maximum of $g_i(\vec{x})$ or equal index sets which attains the common maximum (giving equal classification on the decision boundaries). It should be noted that this definition indirectly requires both equal feature spaces and equal number of discriminant functions (which is equal to the number of classes).

It is useful to note the following general equivalences:

Theorem 1 (General equivalences of discriminant functions)**.** *A set of discriminant functions are equivalent to the set obtained by adding, subtracting or multiplying all discriminant functions in the set with the same positive function that is independent of the class index i (but may depend on e.g. $\vec{x}$). Moreover, the set $g_i(\vec{x})$, $i = 1, \ldots, c$, is equivalent to the set $f(g_i(\vec{x}))$, $i = 1, \ldots, c$, when $f : \mathbb{R} \to \mathbb{R}$ is an arbitrary strictly monotonically increasing function. [13, p.30]*

When studying the performance of a set of features or a specific classifier, the most interesting quantities are those who gives a *probability of misclassification (PMC)*. Indeed, if the learning dataset was of unlimited size, there would only be two quantities worth mentioning: [56, p.253]

- *Asymptotic PMC*: P_∞^α - the PMC of a specific classifier α.
- *Optimal PMC*: $P_\infty := \max_\alpha P_\infty^\alpha$ - the best PMC of all (known and unknown) classifiers when using a specific set of features, i.e. the theoretically best achievable PMC for the given set of features (when extracted from the specific data source under study).

When using an unlimited number of learning patterns, the feature values and the corresponding true classes can in theory be used to find the true values of the parameters that the classifier depends on. Since we in practise have a limited number of learning patterns, the true parameters must typically be replaced by more or less accurate estimates, which may (and probably will) result in a designed classifier different from the one designed with infinite number of patterns. Moreover, since different datasets provide different feature values which in general result in different parameter estimates, the PMC of a designed classifier with a given number of patterns is a random variable. We therefore have the two following interesting PMCs in addition to the asymptotic and optimal PMC:

- *Conditional PMC*: $P_{n_L}^{\alpha}$ - the PMC of a specific classifier α designed using a specific number of patterns n_L (the number of patterns in each class is unspecified). This is a random variable.
- *Expected PMC*: $E(P_{n_L}^{\alpha})$ - the expectation of the conditional PMC. Notice that as n_L approaches infinity, the expected PMC approaches the asymptotic PMC for the given classifier α, in fact, one can define the asymptotic PMC in terms of this relationship [56, p.253]. By the *law of large numbers* (see [11, pp.297–298]), the conditional PMC will converge in probability to the expected PMC as n_L increases. From this we know that the conditional PMC will also approach the asymptotic PMC as n_L approaches infinity.

Because of the restricted number of patterns, we will in practise also only have a limited number of validation patterns. We can therefore in practise only obtain estimates of the four different PMCs described above. While the description of how to obtain some such estimates and a discussion of their appropriateness is left to section 6.6, we will here note that the distribution of any such estimator is also of interest. In particular, the expectation and the variance of this distribution are interesting characteristics of such estimators.

6.2 Bayesian decision theory

We will in this section assume that the set of features is selected, making the classification method the only remaining choice to have a specific classifier.

The fundamental assumption in Bayesian decision theory is that the set of feature vectors from a given class consists of independent realisations of a continuous random variable $\vec{X}|\Omega = \omega_i$ with a true, but usually unknown conditional pdf $f_{\vec{X}|\Omega=\omega_i}(\vec{x}|\Omega = \omega_i)$. Assuming the knowledge of the conditional pdfs, $f_{\vec{X}|\Omega=\omega_1}(\vec{x}|\Omega = \omega_1) \dots f_{\vec{X}|\Omega=\omega_c}(\vec{x}|\Omega = \omega_c)$, and the *a priori* probabilities, $p_\Omega(\omega_1) \dots p_\Omega(\omega_c)$, we can apply Bayes' formula to obtain the true *a posteriori* probability of each class given a pattern with feature vector $\vec{x}$:

$$p_{\Omega|\vec{X}=\vec{x}}(\omega_i|\vec{X} = \vec{x}) = \frac{f_{\vec{X}|\Omega=\omega_i}(\vec{x}|\Omega = \omega_i)p_\Omega(\omega_i)}{f_{\vec{X}}(\vec{x})} \tag{6.1}$$

where $f_{\vec{X}}(\vec{x})$ can be computed using the law of total probability:

$$f_X(\vec{x}) = \sum_{i=1}^{c} f_{\vec{X}|\Omega=\omega_i}(\vec{x}|\Omega = \omega_i)p_\Omega(\omega_i) \tag{6.2}$$

Under the assumption of the origin of the feature vectors, the true asymptotic PMC of a specific classifier can easily be derived as [13, p.22]:

$$P(\text{error}) = \int_{-\infty}^{\infty} P(\text{error} \wedge \vec{X} = \vec{x})d\vec{x} = \int_{-\infty}^{\infty} P(\text{error}|\vec{X} = \vec{x})f_{\vec{X}}(\vec{x})d\vec{x} \tag{6.3}$$

By equation (6.2) we see that, under the assumption of known conditional pdfs and *a priori* probabilities, $f_{\vec{X}}(\vec{x})$ is completely determined. However, the other factor in the integrand, $P(\text{error}|\vec{X} = \vec{x})$, depends on the choice of classification

method. The best possible method will for all $\vec{x} \in \mathfrak{D}$ choose the class with least PMC. The resulting classifier is known as *Bayes' classifier* and is by definition a classifier which obtains the optimal PMC given the set of features and under the assumption of the feature vectors' origin. The decision rule of this classifier can be expressed as:

Decision rule 3 (Bayes' decision rule)**.** *For all $\vec{x} \in \mathfrak{D}$, classify the pattern to the class ω_i for which $p_{\Omega|\vec{X}=\vec{x}}(\omega_i|\vec{X} = \vec{x}) \geq p_{\Omega|\vec{X}=\vec{x}}(\omega_j|\vec{X} = \vec{x})$ for all $j = 1, \ldots, c$ (if there are multiple choices for ω_i, any will do).*

or by using decision rule 1 or (2) with discriminant functions defined as:

$$\begin{aligned} g_i(\vec{x}) :=& p_{\Omega|\vec{X}=\vec{x}}(\omega_i|\vec{X} = \vec{x}) \overset{(6.1)}{=} \frac{f_{\vec{X}|\Omega=\omega_i}(\vec{x}|\Omega = \omega_i)p_\Omega(\omega_i)}{f_{\vec{X}}(\vec{x})} \\ \equiv& f_{\vec{X}|\Omega=\omega_i}(\vec{x}|\Omega = \omega_i)p_\Omega(\omega_i) \end{aligned} \tag{6.4}$$

where the last transition follows from theorem 1 and the fact that $(f_{\vec{X}}(\vec{x}))^{-1}$ is positive and independent of the class index i.

The asymptotic PMC of Bayes' classifier will in the following be denoted by $P_{\mathfrak{B}}$ and can be computed by inserting Bayes' decision rule into equation (6.3):

$$P_{\mathfrak{B}} = 1 - \int_{-\infty}^{\infty} \max_{i=1,\ldots,c} \{p_{\Omega|\vec{X}=\vec{x}}(\omega_i|\vec{X} = \vec{x})\} f_{\vec{X}}(\vec{x}) d\vec{x} \tag{6.5}$$

In practise, both the conditional pdfs and the *a priori* probabilities are unknown. In many applications, it is reasonable to assume that the probabilities of occurrence of each class is the same in the learning dataset as in novel patterns, thus allowing the *a priori* probabilities to be easily estimated as the corresponding proportions in the learning dataset. Obtaining a proper estimate of the conditional pdfs are however worse and can be tackled in different ways. Broadly divided, there are two different approaches, *parametric classification* and *nonparametric classification*.

6.2.1 Parametric classification

In parametric classification, we assume that all conditional pdfs belongs to the same distribution class, but possibly with different true values of its parameters. The problem of estimating each conditional pdf is then reduced to the problem of estimating the parameters of each distribution. The method of estimation is of course arbitrary in general, but the common practise is to use the *maximum likelihood estimator (MLE)*. The MLE is recommended by most statisticians, at least for large datasets, due to some attractive properties. In particular, the *invariance principle* of the MLE states that given the MLEs of a number of parameters, the MLE of any function of these parameters is simply obtained by replacing the parameters with its MLEs in the function expression. In general, the computation of a composed estimator must be computed from scratch, but this property makes such computation trivial when using MLEs. The most important theoretical property is however related to the large sample behaviour of the MLE, which roughly states that all MLEs are approximately unbiased estimators attaining (nearly) the minimum possible variance of all unbiased estimators. [11, pp.346,351–352]

In practise, the assumed distribution of the conditional pdfs are almost without exception the normal distribution (when parametric classification is used). Moreover, the covariance matrix of the normal distribution is assumed to be invertible, but this is only a minor assumption as a singularity of the covariance matrix indicates a redundancy of some features [13, p.34].

There are several arguments supporting the assumption of normality of the conditional pdfs, or at least supporting that this assumption is in general more appropriate than assuming any other distribution class of all conditional pdfs. An essential argument follows from the central limit theorem, in particular that the sum (or average) of many small, independent random sources of noise will converge to a normal distribution as the number of sources increase, which makes the normal distribution the appropriate model if the conditional pdfs of $\vec{X}|\Omega = \omega_i$ may be viewed as the result of a typical or prototype vector $\vec{\mu}_i$ corrupted by continuously valued, random noise [13, pp.31,33]. Another interesting property is that the normal distribution has the maximum entropy of all distributions having the same mean and variance, i.e. the normal distribution has the theoretically maximum informational content and uncertainty with respect to a given mean and variance [13, pp.32–33]. Finally, the classifiers produced when assuming normality of the conditional pdfs are to some degree simplistic and therefore likely to generalise acceptably. In fact, we will later see that under addition assumptions the classifier includes the best possible linear separation with respect to an intuitively reasonable criterion function, see section 6.4.1.

The assumption of normality can be stated as $\vec{X}|\Omega = \omega_i \sim N(\vec{\mu}_i, \Sigma_i)$ for $i = 1, \ldots, c$, where $\mu_i \in \mathbb{R}^d$ is the expectation and $\Sigma_i \in \mathbb{R}^{d,d}$ is the covariance matrix. The Bayesian decision rule can then be obtained by using the discriminant functions:

$$
\begin{aligned}
g_i(\vec{x}) &\overset{(6.4),1}{\equiv} \ln(f_{\vec{X}|\Omega=\omega_i}(\vec{x}|\Omega = \omega_i)) + \ln(p_\Omega(\omega_i)) \\
&\overset{\text{normality}}{=} \ln(\frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T\Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)}) + \ln(p_\Omega(\omega_i)) \\
&= -\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T\Sigma_i^{-1}(\vec{x}-\vec{\mu}_i) - \frac{d\ln(2\pi)}{2} - \frac{\ln|\Sigma_i|}{2} + \ln(p_\Omega(\omega_i)) \\
&\overset{1}{\equiv} -\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T\Sigma_i^{-1}(\vec{x}-\vec{\mu}_i) - \frac{\ln|\Sigma_i|}{2} + \ln(p_\Omega(\omega_i)) \qquad (6.6) \\
&= -\frac{1}{2}\vec{x}^T\Sigma_i^{-1}\vec{x} + \vec{\mu}_i^T\Sigma_i^{-1}\vec{x} - \frac{1}{2}\vec{\mu}_i^T\Sigma_i^{-1}\vec{\mu}_i - \frac{\ln|\Sigma_i|}{2} + \ln(p_\Omega(\omega_i)) \qquad (6.7)
\end{aligned}
$$

where the last equality follows from that the covariance matrix is symmetric by definition.

Since the covariance matrix is symmetric, it contains 'only' $0.5d(d+1)$ unique parameters, and the expectation contains d values, thus a total of $0.5d(d+3)$ independent parameters are available for each class. If the number of features is high, then the number of independent parameters will be large. This can in turn make the designed classifier overfitted to the learning dataset, a phenomenon which will receive more attention later in section 6.3, but for now it is sufficient that we know it may be beneficial to restrict the number of independent parameters. In the case of normal conditional pdfs, this can be done by assuming values or relationships among the $0.5d(d+3)$ parameters. There are two such assumptions which are commonly mentioned, both restricting the number of independent covariances.

Case 1: Independent features with equal variances, $\Sigma_i = \sigma^2 I$

In this simple case, all features are assumed to be statistically independent and to have the same common variance σ^2. Thus there is in this case only a single independent covariance parameter, making the total number of independent parameters $cd + 1$. The discriminant functions can in this case be written as:

$$\begin{aligned} g_i(\vec{x}) &\overset{(6.6)}{=} -\frac{1}{2}(\vec{x} - \vec{\mu}_i)^T \Sigma_i^{-1}(\vec{x} - \vec{\mu}_i) - \frac{\ln|\Sigma_i|}{2} + \ln(p_\Omega(\omega_i)) \\ &= -\frac{\|\vec{x} - \vec{\mu}_i\|_2^2}{2\sigma^2} - 2\ln\sigma + \ln(p_\Omega(\omega_i)) \\ &\overset{1}{\equiv} -\frac{\|\vec{x} - \vec{\mu}_i\|_2^2}{2\sigma^2} + \ln(p_\Omega(\omega_i)) \end{aligned} \tag{6.8}$$

In the case of equal *a priori* probabilities, it is easy to see that a classifier based on the discriminant functions in equation (6.8) will simply classify each feature vector to the class belonging to the nearest expectation, as the discriminant function of this class will provide the largest value of the discriminant functions because it attains the least possible distance $\|\vec{x} - \vec{\mu}_i\|_2$, $i = 1, \ldots, c$, for the specific feature vector. Such a classifier is known as the *minimum distance classifier* and the distance is measured using the Euclidean norm.

In the case of a dichotomiser and by applying theorem 1, it is easy to further simplify the discriminant functions in equation (6.8) to $g(\vec{x}) = \vec{w}^T(\vec{x} - \vec{x}_0)$ where:

$$\vec{w} = \vec{\mu}_1 - \vec{\mu}_2 \tag{6.9}$$

$$\vec{x}_0 = \frac{\vec{\mu}_1 + \vec{\mu}_2}{2} - \frac{\sigma^2(\vec{\mu}_1 - \vec{\mu}_2)}{\|\vec{\mu}_1 - \vec{\mu}_2\|_2^2} \ln\left(\frac{p_\Omega(\omega_1)}{p_\Omega(\omega_2)}\right) \tag{6.10}$$

The resulting decision boundary is thus a hyperplane orthogonal to the line between the expectations, as given in equation (6.9). Furthermore, equation (6.10) shows that the location of this hyperplane is precisely the middle point of the expectations if the *a priori* probabilities are equal, an observation that can also be derived independently as we known the classifier is equivalent to the minimum distance classifier in this case. If there is a deviation in the *a priori* probabilities, then the second term in equation (6.10) is a nonzero constant times the difference vector between the expectations, so the hyperplane will be translated along the line between the expectations. These observations are visualised in figure 6.2 and 6.3 for the case of one and two features, respectively.

Case 2: Equal covariance matrices, $\Sigma_i = \Sigma$

A slightly more complex case is when all features are assumed to have the same common covariance matrix Σ. As a general covariance matrix contains $0.5d(d+1)$ independent parameters, the total number of independent parameters is $cd + 0.5d(d+1) = 0.5d(2c + d + 1)$ in this case. The discriminant functions can now be written as:

$$\begin{aligned} g_i(\vec{x}) &\overset{(6.6)}{=} -\frac{1}{2}(\vec{x} - \vec{\mu}_i)^T \Sigma_i^{-1}(\vec{x} - \vec{\mu}_i) - \frac{\ln|\Sigma_i|}{2} + \ln(p_\Omega(\omega_i)) \\ &\overset{1}{\equiv} -\frac{1}{2}(\vec{x} - \vec{\mu}_i)^T \Sigma^{-1}(\vec{x} - \vec{\mu}_i) + \ln(p_\Omega(\omega_i)) \end{aligned} \tag{6.11}$$

$$\overset{1}{\equiv} \vec{\mu}_i^T \Sigma^{-1}\vec{x} - \frac{1}{2}\vec{\mu}_i^T \Sigma^{-1}\vec{\mu}_i + \ln(p_\Omega(\omega_i)) \tag{6.12}$$

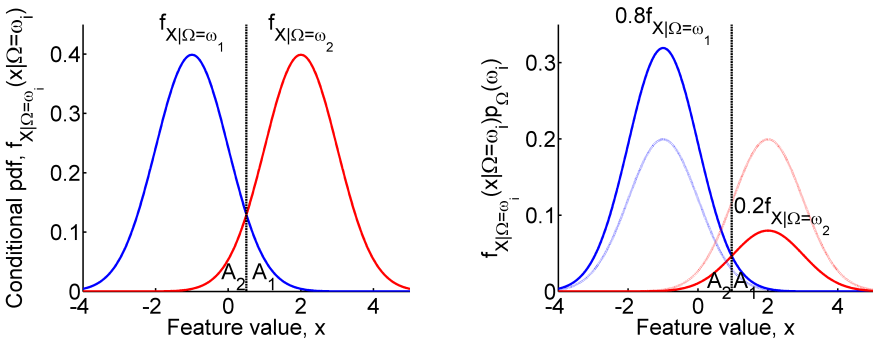

Figure 6.2: Two normal distributions with equal variances. Under the assumption of normality and equal variances, this is an idealised case when the number of features is one and the number of classes is two. The dashed, black line is the decision boundary of the Bayes decision rule under the assumption of (left) equal a priori *probabilities or (right)* $p_\Omega(\omega_1) = 0.8$ *and* $p_\Omega(\omega_2) = 0.2$*, and the area of* A_1 *and* A_2 *is the probability of misclassification when the true class is* ω_1 *and* ω_2*, respectively.*

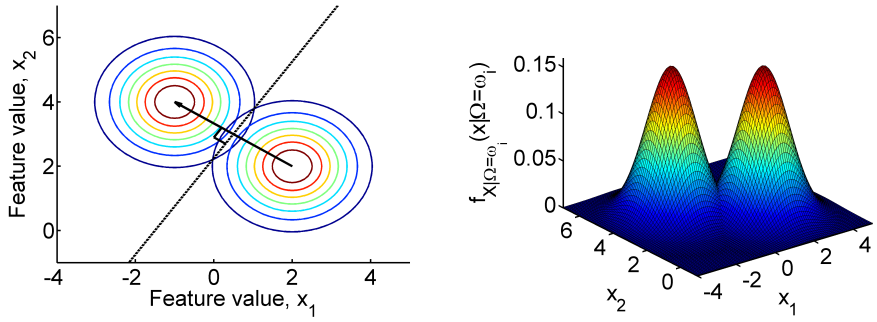

Figure 6.3: Two bivariate normal distributions with independent components with equal variances. Under the assumption of normality and independent features with equal variances, this is an idealised case when the number of features is two and the number of classes is two. The dashed, black line is the decision boundary of the Bayes decision rule under the assumption of equal a priori *probabilities. The solid, black vector is* $\vec{w} = \vec{\mu}_1 - \vec{\mu}_2$*, which also is the normal vector of the decision boundary.*

The formulation in equation (6.11) strongly resembles the formulation of in equation (6.8), a set of discriminant functions that under the assumption of equal *a priori* probabilities defines a minimum distance classifier. In fact, under the assumption of equal *a priori* probabilities, equation (6.11) also defines a classifier that decides the class with minimum distance between its expectation and the feature vector, as the minimum distance classifier, only that the distance is now measured in terms of the norm defined by the matrix Σ^{-1}.

In the case of a dichotomiser, it is easy to show that $g(\vec{x}) = \vec{w}^T(\vec{x} - \vec{x}_0)$

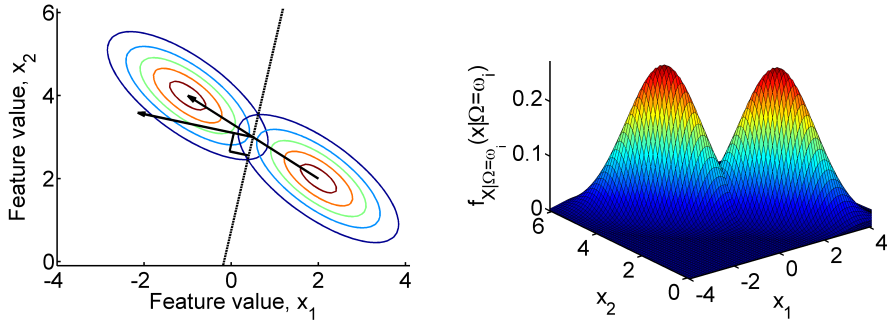

Figure 6.4: Two bivariate normal distributions with equal covariances. Under the assumption of normality and equal covariances, this is an idealised case when the number of features is two and the number of classes is two. The dashed, black line is the decision boundary of the Bayes decision rule under the assumption of equal a priori *probabilities. Though the decision boundary passes through the average of the expectations, $\vec{\mu}_1 - \vec{\mu}_2$, this vector is not the normal vector of the decision boundary as indicated by the solid, blank vectors.*

where:

$$\vec{w} = \Sigma^{-1}(\vec{\mu}_1 - \vec{\mu}_2) \tag{6.13}$$

$$\vec{x}_0 = \frac{\vec{\mu}_1 + \vec{\mu}_2}{2} - \frac{\vec{\mu}_1 - \vec{\mu}_2}{(\vec{\mu}_1 - \vec{\mu}_2)^T \Sigma^{-1} (\vec{\mu}_1 - \vec{\mu}_2)} \ln\left(\frac{p_\Omega(\omega_1)}{p_\Omega(\omega_2)}\right) \tag{6.14}$$

The resulting decision boundary is thus again a hyperplane, but the normal vector of this hyperplane, as given in equation (6.13), is generally not orientated as the difference vector between the expectations (as it was in case 1). The middle point of the expectations is however still a part of the hyperplane if we assume equal *a priori* probabilities, as we see from equation (6.14), which also shows us that the translation caused by unequal *a priori* probabilities is along the line between the expectations as in the previous case. These observations are visualised in figure 6.4 for the case of two features.

Case 3: No further assumptions, Σ_i is arbitrary

In this last case we do not assume any values or relationships on the parameters. We have already mentioned that we then will have $0.5d(d+3)$ independent parameters for each class, thus a total of $0.5cd(d+3)$ independent parameters. The discriminant functions of this case were given in equation (6.7). The resulting decision boundaries are general piecewise hyperquadrics, or a single general hyperquadric for dichotomisers. In fact, given any hyperquadratic, there exists always two normal distributions whose decision boundaries using the Bayes decision rule is that hyperquadric [13, p.42]. Some of the many possible forms of the decision boundaries are visualised in figure 6.5 for the case of two features.

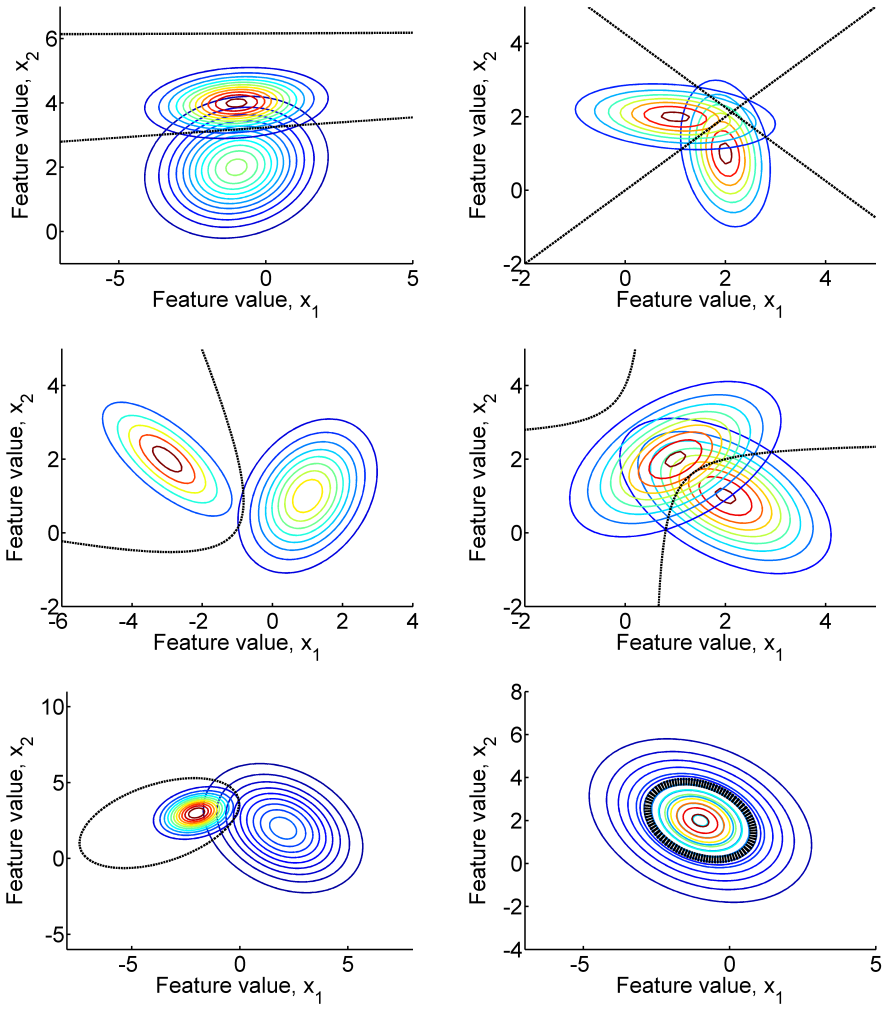

Figure 6.5: Two bivariate normal distributions. Under the assumption of normality, this is an idealised case when the number of features is two and the number of classes is two. The dashed, black curves are the decision boundaries of the Bayes decision rule under the assumption of equal a priori *probabilities, which are quadrics for the case of two features.*

Discussion

While all derived classifiers attain the optimal PMC under the stated assumptions and with knowledge about the true value of parameters, problems arise when any of these conditions are not met. The problem of not knowing the true value of the parameters, which is the typical case in practise, is related to the a general phenomenon known as *overfitting*, a problem that will receive more attention in section 6.3. For now we shall only note that the problem increases with the number of independent parameters. A violation of the stated

assumptions, in particular an incorrect assumption of the conditional pdfs (the assumption of the feature vectors' origin is mild if the patterns can be said to be 'independent'), may further decrease the performance of the classifier.

It has been noted by many authors that the linear discriminant functions obtained in case 2 above is robust to discrepancies from the normality assumption [56, p.253]. This can be partly explained by its strong correlation with the Fisher's linear discriminant, a method described in section 6.4.1 that optimises the linear separation between the classes with respect to an intuitively reasonable criterion function. As this method does not rely on the normality assumption, nor any other assumptions about the parametric form of the conditional pdfs for that sake, it seems reasonable that importance of the normality assumption in a classifier of case 2 above is minor.

In contrast, the hyperquadratic decision boundary obtained when using arbitrary covariance matrices, significantly suffers from a violation of the normality assumption. Another critique directed toward this classifier is that its performance degenerates when the number of patterns in each class in the learning dataset differs, in fact, a learning dataset with equal number of patterns of each class may have a lower expected PMC than a similar learning dataset where the number of patterns of only one class has increased. A generalisation accommodating for this negative effect has been proposed, see [56, p.253–254] for details. [56, p.253]

6.2.2 Nonparametric classification

We now turn our attention to a different approach of estimating the true, but unknown conditional pdfs. This time we will not assume a specific distribution class of all pdfs, we will instead let the data describe the distributions. While this approach may seem favourable in all circumstances, it is in practise not that simple, because the arbitrary estimation of the conditional pdfs requires a much larger learning dataset to be properly estimated than is the case when estimating some relatively few independent parameters of an assumed distribution. This problem is related to the complexity of the classifier and will be discussed later in section 6.3.

The positive side of this nonparametric approach compared to the parametric counterpart is obvious. We omit the possibly (and maybe even probably) flawed assumption of a specific distribution class of all conditional pdfs and in effect allow the estimation of distributions closer to the true pdfs. In particular, the nonparametric classifiers allow different distributions of both the marginal pdfs of a conditional pdf and the different conditional pdfs, in addition to allowing unknown distributions of the conditional pdfs.

We will in the following describe two common nonparametric classifiers, the *Parzen window classifier* and the *k-nearest neighbour classifier*. They both share the same fundamental technique of how to estimate a distribution of a continuous, random variable $\vec{X}$ from a set of independent realisations of the variable, $\vec{x}_1, \ldots \vec{x}_n$. In image analysis, there will in general be c distributions to estimate, $\vec{X}|\Omega = \omega_1, \ldots, \vec{X}|\Omega = \omega_c$, and the set of feature vector associated with each class is assumed to be the set of independent realisations from the corresponding conditional pdf.

Let $\mathfrak{R}$ be any region in $\mathfrak{D}$ and define $p_{\mathfrak{R}}$ as the probability of a realisation

occurring in that region, more precisely, define:

$$p_{\mathfrak{R}} := \int_{\mathfrak{R}} f_{\vec{X}}(\vec{x}) d\vec{x} \tag{6.15}$$

Define $B_{\mathfrak{R}}$ as the binomial variable of the number of occurrences in the region $\mathfrak{R}$ among the n realisation, $\vec{x}_1, \dots \vec{x}_n$. $B_{\mathfrak{R}} \sim Bin(n, p_{\mathfrak{R}})$ is then a random variable depending on the set of realisations. In particular we note that:

$$E(B_{\mathfrak{R}}) = np_{\mathfrak{R}} \tag{6.16}$$

Since a binomial variable is just the sum of multiple independent Bernoulli variables with parameter $p_{\mathfrak{R}}$, it follows from the law of large numbers that the average of the Bernoulli variables, $B_{\mathfrak{R}}/n$, will converge in probability to $p_{\mathfrak{R}}$. A reasonable estimator of $p_{\mathfrak{R}}$ is thus:

$$\widehat{P_{\mathfrak{R}}} = \frac{B_{\mathfrak{R}}}{n} \tag{6.17}$$

with the corresponding estimate:

$$\widehat{p_{\mathfrak{R}}} := \frac{k_{\mathfrak{R}}}{n} \tag{6.18}$$

where $k_{\mathfrak{R}}$ is the observed quantity of $B_{\mathfrak{R}}$ (the symbol $k_{\mathfrak{R}}$ is chosen instead of $b_{\mathfrak{R}}$ because of a prominent convention of using k in this context).

As the law of large numbers only applies when there is literally an infinite number of realisations inside the region $\mathfrak{R}$, we should comment whether this approximation is acceptable for small values of n. Under the assumption of a large dataset, this can be justified because a small number of realisations in the region will then strongly indicate a minor probability of occurrence in that region, i.e. a small value of $p_{\mathfrak{R}}$, and since the variance of a Bernoulli variable is $p_{\mathfrak{R}}(1 - p_{\mathfrak{R}})$, this means that a small value of n implies a small variance of the estimator in equation (6.17), which is $p_{\mathfrak{R}}(1 - p_{\mathfrak{R}})/n$. We thus need to assume that the dataset is large, if not, it is possible that the number of occurrence in a specific region is small even though the probability of occurrence in that region is relatively large.

We will further assume that $f_{\vec{X}}$ is continuous and $\mathfrak{R}$ is so small that $f_{\vec{X}}$ is nearly constant in the region. Under these assumptions, we have that:

$$\int_{\mathfrak{R}} f_{\vec{X}}(\vec{x}') d\vec{x}' \approx f_{\vec{X}}(\vec{x}) V_{\mathfrak{R}} \tag{6.19}$$

where $V_{\mathfrak{R}}$ is the hypervolume of $\mathfrak{R}$.

In summary, for a given point $\vec{x}$, we now have the following approximation under the stated assumptions:

$$f_{\vec{X}}(\vec{x}) \overset{(6.19), V_{\mathfrak{R}} > 0}{\approx} \frac{1}{V_{\mathfrak{R}}} \int_{\mathfrak{R}} f_{\vec{X}}(\vec{x}') d\vec{x}' \overset{(6.15)}{=} \frac{p_{\mathfrak{R}}}{V_{\mathfrak{R}}} \approx \frac{\widehat{p_{\mathfrak{R}}}}{V_{\mathfrak{R}}} \overset{(6.18)}{=} \frac{k_{\mathfrak{R}}/n}{V_{\mathfrak{R}}} \tag{6.20}$$

We can from this define an estimate of the distribution of $\vec{X}$ for every point $\vec{x}$ as following:

$$\widehat{f_{\vec{X}}}(\vec{x}) := \frac{k_{\mathfrak{R}(\vec{x})}/n}{V_{\mathfrak{R}(\vec{x})}} \tag{6.21}$$

where $\vec{x}$ is included as an argument of $\mathfrak{R}$ to indicate that the region is adapted to the point in question.

The only thing that remains in order to have an explicitly defined method using the estimate in equation (6.21) is how to wisely choose the regions $\mathfrak{R}(\vec{x})$. To assist our choice we can use three conditions that are necessary if $\widehat{f_{\vec{X}}}(\vec{x})$ is to converge to $f_{\vec{X}}(\vec{x})$: [13, p.163]

$$\lim_{n\to\infty} V_{\mathfrak{R}(\vec{x})} = 0 \tag{6.22}$$

$$\lim_{n\to\infty} k_{\mathfrak{R}(\vec{x})} = \infty \tag{6.23}$$

$$\lim_{n\to\infty} k_{\mathfrak{R}(\vec{x})}/n = 0 \tag{6.24}$$

In order to satisfy these condition the region will have to depend on the number of independent realisations. More precisely, the region must approach the empty set while the number of realisations within the region approaches infinity, but in a rate such that the number of realisations within the region is negligible with respect to the total number of realisations.

Parzen window classifier

In the Parzen window classifier, we define the hypervolume in equation (6.21) as a function of the number of realisations as follows:

$$V_{\mathfrak{R}(\vec{x})} := V_n \tag{6.25}$$

The dependence of the number of realisations follows from the discussed necessary conditions of convergence, see equations (6.22)-(6.24). The hypervolume V_n is further defined in terms of a *window width* or *smoothing parameter* h_n as follows:

$$V_n := h_n^d \tag{6.26}$$

where d is the dimension of the realisations, which is the number of features in our case.

Define a *window function* $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ as a measure of the distance to origin. The number of realisations within the region is defined in terms of this window function and the window width as follows:

$$k_{\mathfrak{R}(\vec{x})} := \sum_{i=1}^{n} \varphi\left(\frac{\vec{x} - \vec{x}_i}{h_n}\right) \tag{6.27}$$

By inserting equations (6.25) and (6.27) into equation (6.21), we obtain the following estimate of the distribution of $\vec{X}$:

$$\widehat{f_{\vec{X}}}(\vec{x}) = \frac{1}{nV_n} \sum_{i=1}^{n} \varphi\left(\frac{\vec{x} - \vec{x}_i}{h_n}\right) \tag{6.28}$$

In essence, for any point $\vec{x}$ this estimate is an interpolation of all realisations with weights proportional to the distance from $\vec{x}$ to each of the realisations. The distances are measured in terms of the window width, which thus determines the focus of the interpolation, and the transitions is determined by the window function.

We must restrict the definition of the window function to ensure a meaningful estimate in equation (6.28). Firstly, it is natural that the window function is an unimodal, symmetric function centred at the origin and decreasing along any direction away from origin. This requirement roughly states that realisation close to the point in question contributes more in the estimate than realisations further way, but is only of practical purpose. Secondly, we require the window function to be a pdf. This is a theoretical requirement that can be shown to be a sufficient condition for that the resulting estimate is a pdf [13, p.165].

With respect to the origin of the estimate in equation (6.21), which is based on a strictly defined region $\mathfrak{R}(\vec{x})$ with hypervolume $V_{\mathfrak{R}(\vec{x})}$ and number of realisations $k_{\mathfrak{R}(\vec{x})}$, we should let the window function be a hypercube centred at origin and with side h_n. The estimate in equation (6.28) would then be the ratio of the number of realisations within a length of $h_n/2$ from the argument point to the total number of realisations (scaled with the constant $V_n^{-1} = h_n^{-d}$). Such restriction of the window function is however not required for the practical use of this classification method. In fact, because the sharp transition of the hypercubical window function can be said to be artificial because the separation of contributing and ignored realisations is typically not crisp, we expect window functions with smoother transitions to perform better in general.

From a theoretical point of view, it is interesting to study necessary and sufficient conditions of how the window width needs to depend on the number of realisations in order to ensure convergence to the true, unknown distribution of $\vec{X}$. We will however instead take a practical approach to this choice and refer interested readers to [13, pp.166–168].

In practise, we are left with two choices, the distribution of the window function and the window width. With considerations of noise, statistical independence of the realisations and its variance, a generic normal window shape is justifiable without any other problem specific information [13, p.169]. The exact distribution could be given some attention. In particular, if the typical range of the realisations along different axes seem to be different, this should be taken into account by e.g. setting the standard deviation of the window function along each axis proportional to a measurement of the typical range of the realisations along that axis, e.g. the standard deviation of the realisations along that axis. However, in an experiment using 13 different types of window function, including the normal window shape, it was found that the PMC was nearly identical for every window function as long as the window width was chosen properly, indicating that the importance of the choice of window function is minor relative to the choice of window width [56, p.254].

The problem of choosing a window width is generally complicated. Indeed, there exists little theoretical justification of which window width is best in general [13, p.169]. A recommended approach is to estimate a PMC using several different window widths and choosing the window width attaining the minimum PMC. This procedure is practically acceptable as the valley of the expected PMC against the window width is wide for most practical problems, thus allowing relatively widely sampled window widths. The problem of estimating the PMC is thus the most troubling, as estimating a PMC using only the learning dataset increases the risk of overfitting (discussed in general in section 6.3), but using a separate *tuning dataset* will affect either the accuracy of the estimated PMC or the conditional and expected PMC of the resulting classifier, see section 6.6 for details. [56, p.254]

In practise, the performance of a Parzen window classifier is highly dependent on the number of realisations. In particular, the classifier will require many more realisations with an increasing dimension of the vectors, e.g. the number of features. Roughly speaking, if we assume that the required number of realisations is known to be a when the dimension is one, then the required number of realisations with d dimensions is a^d, under the assumption of equal resolution in each dimension. In other words, the required number of realisations grows exponentially with the dimension of the vectors. This problem is related to the problem of overfitting, which will be discussed in general in section 6.3, and is often referred to as the *curse of dimensionality* after Bellman [3, pp.94,197–198], though the problem in itself was known prior to Bellman's work. In the context of the Parzen window classifier, we can also recognise the problem directly because it is clear that increasing the dimension makes the space filled by a constant number of realisations n rapidly sparser and quickly the method will have to choose between an extremely large window width or a highly unreliable point estimate because of few - or none - closely located realisation, i.e. $k_{\mathfrak{R}(\vec{x})}$ is microscopic.

In summary, we will define the discriminant functions which may be inserted into decision rule 1 or 2 to obtain the Bayes' decision rule using the Parzen window classifier:

$$\begin{aligned} g_i(\vec{x}) &\overset{(6.4)}{=} f_{\vec{X}|\Omega=\omega_i}(\vec{x}|\Omega=\omega_i)p_\Omega(\omega_i) \\ &\approx \widehat{f_{\vec{X}|\Omega=\omega_i}}(\vec{x}|\Omega=\omega_i)p_\Omega(\omega_i) \\ &\overset{(6.28),1}{\equiv} \frac{p_\Omega(\omega_i)}{n_i}\sum_{j=1}^{n_i}\varphi\left(\frac{\vec{x}-\vec{x}_{i,j}}{h_n}\right) \\ &\overset{(6.27)}{=} \frac{k_{\mathfrak{R}(\vec{x}),i}p_\Omega(\omega_i)}{n_i} \end{aligned} \tag{6.29}$$

where $k_{\mathfrak{R}(\vec{x}),i}$ is the weighted contribution of class ω_i in the fuzzy region $\mathfrak{R}(\vec{x})$ as given in equation (6.27), the window function φ is an arbitrary chosen pdf, e.g. the multivariate normal distribution with an appropriately chosen covariance matrix, and the window width h_n is a problem specific constant that should be estimated, e.g. as the window width attaining the minimum estimated PMC when evaluated using the learning dataset. In particular, if we estimate the *a priori* probabilities as the corresponding proportions in the learning dataset, we obtain the following discriminant functions:

$$g_i(\vec{x}) \overset{(6.29)}{=} \frac{k_{\mathfrak{R}(\vec{x}),i}p_\Omega(\omega_i)}{n_i} = \frac{k_{\mathfrak{R}(\vec{x}),i}}{n_i}\frac{n_i}{n_L} \overset{1}{\equiv} k_{\mathfrak{R}(\vec{x}),i} \tag{6.30}$$

and the decision rule will simply be to decide the class with maximum weighted contribution in the fuzzy region $\mathfrak{R}(\vec{x})$.

k-nearest neighbour classifier

Let $k_i(\vec{x}, k_n)$ be the number of realisations of class ω_i of the k_n nearest neighbours of $\vec{x}$ measured with respect to some norm, e.g. the Euclidean norm. The k-nearest neighbour (kNN) classifier can then be defined as follows:

Decision rule 4 (kNN decision rule)**.** *For all $\vec{x} \in \mathfrak{D}$, classify the pattern to the class ω_i for which $k_i(\vec{x}, k_n) \geq k_j(\vec{x}, k_n)$ for all $j = 1, \ldots, c$ (if there are multiple choices for ω_i, any will do).*

Roughly speaking, the kNN classifier decides the most frequent class among the k_n nearest neighbours of the argument point.

The kNN decision rule can be related to the nonparametric distribution estimate by stating that equation (6.21) indicates that the following is an appropriate estimate of the joint distribution of $\vec{X}, \Omega$:

$$\widehat{f_{\vec{X},\Omega}}(\vec{x}, \omega_i) := \frac{k_i(\vec{x}, k_n)/n_i}{V_{\Re(\vec{x})}} \tag{6.31}$$

Using this estimate we find an estimate of the a posteriori probabilities:

$$\widehat{f_{\vec{X}|\Omega=\omega_i}}(\vec{x}|\Omega = \omega_i) := \frac{\widehat{f_{\vec{X},\Omega}}(\vec{x}, \omega_i)}{\sum_{j=1}^{c} \widehat{f_{\vec{X},\Omega}}(\vec{x}, \omega_j)} \overset{(6.31)}{=} \frac{k_i(\vec{x}, k_n)/n_i}{\sum_{j=1}^{c} k_j(\vec{x}, k_n)/n_j} \tag{6.32}$$

which results in the following discriminant functions:

$$\begin{aligned} g_i(\vec{x}) &\overset{(6.4)}{=} f_{\vec{X}|\Omega=\omega_i}(\vec{x}|\Omega = \omega_i) p_\Omega(\omega_i) \\ &\approx \widehat{f_{\vec{X}|\Omega=\omega_i}}(\vec{x}|\Omega = \omega_i) p_\Omega(\omega_i) \\ &\overset{(6.32),1}{\equiv} \frac{k_i(\vec{x}, k_n) p_\Omega(\omega_i)}{n_i} \end{aligned} \tag{6.33}$$

If we estimate the *a priori* probabilities as the corresponding proportions in the learning dataset, we obtain the following discriminant function:

$$g_i(\vec{x}) \overset{(6.33)}{=} \frac{k_i(\vec{x}, k_n) p_\Omega(\omega_i)}{n_i} = \frac{k_i(\vec{x}, k_n)}{n_i} \frac{n_i}{n_L} \overset{1}{\equiv} k_i(\vec{x}, k_n) \tag{6.34}$$

which is precisely decision rule 4 when inserting the discriminant functions into decision rule 1.

The kNN classifier has a clear practical advantage over the Parzen window classifier; it dynamically determines the region, which is specified by the window width in the Parzen window method, based on the density of the realisations surrounding the point in question. This seems favourable as we want the region to be small (focused) whenever possible, as this provides an accurate estimate, but we need several realisations to ensure a reliable estimate. As the density of the realisations are in all practical cases bound to vary with the location of $\vec{x}$, a fixed region can not expected to be optimal from all locations of $\vec{x}$, so the kNN classifier has practical advantages over the Parzen window classifier.

A devastating flaw from a theoretical point of view should be commented in this context. As at least one realisation would have to be included in at least one of the pdfs in both equation (6.31) and equation (6.32) for any point $\vec{x}$, at least one of the pdfs in both these estimates will have to diverge to infinity for any finite number of realisations n. This is rather embarrassing as it shows that the resulting estimates are far from pdfs, as pdfs always integrate to one. As commented, from a practical point of view this is only favourable as we will always obtain some estimate of the probability in each point, even when it is microscopic.

Though the dynamical determination of the region allows the kNN classifier to more efficiently use a limited number of realisations, it will in general also create a complex decision boundary. In fact, the complexity of the kNN classifier is in general extremely high, but it will depend on the chosen value of k_n - a higher value will result in a less complex classifier.

When applying the kNN classifier, we would have to determine the number of required realisation in the region, k_n. This problem is analogous with the problem of setting the window width in the Parzen window classifier; performance depends significantly on the value of k_n, but there is in general no way of reasonably assigning this value. Again, a recommended approach is to let the value be determined by evaluating multiple values and choosing the one obtaining the minimum estimated PMC, e.g. by using the learning dataset for the evaluations. [56, p.255]

Another practical concern is the choice of norm, which is analogous with choice of contour lines for the window function the Parzen window method. As commented, a difference in the typical range of the realisations along different axes should be compensated for. If possible, an alternative is to standardise each feature with respect to some *a priori* knowledge about their distribution, but this knowledge could also be incorporated in the choice of norm or contour lines. It has been reported that the choice does affect the performance, but the importance of this choice is highly dependent on the features under study [56, p.255].

Distribution estimation based on the kNN

The reader may have spotted that the presentation of the kNN classifier was not directly derived from the estimated distribution in equation (6.21). The reason is that the classifier based on this distribution estimate using a constant number of realisations within the region will in general *not* be identical to the kNN classifier.

When estimating a distribution based on the kNN, we define the number of realisations within the region in equation (6.21) as a function of the number of realisations as follows:

$$k_{\Re(\vec{x})} := k_n \tag{6.35}$$

As in the Parzen window method, the dependence of the number of realisations follows from the discussed necessary conditions of convergence of $\widehat{f_{\vec{X}}}(\vec{x})$ in equation (6.21) to the true, unknown distribution of $\vec{X}$, see equations (6.22)-(6.24). In this case however, it can be shown that the following two conditions are both necessary and sufficient for the convergence: [13, p.175]

$$\lim_{n\to\infty} k_n = \infty \tag{6.36}$$

$$\lim_{n\to\infty} k_n/n = 0 \tag{6.37}$$

These conditions can easily be satisfied, e.g. by using $k_n = n^\alpha$ for any $\alpha \in (0, 1)$ or $\log k_n$, but, as with the kNN classifier, a recommended approach in image analysis problems is to estimate k_n based the entire learning dataset.

For any given set of data with n realisations, we define or compute the required number of realisations k_n. Then, for any point $\vec{x}$ we find the k_n nearest realisation with respect to some norm, e.g. the Euclidean norm, and define the

region $V_{\mathfrak{R}(\vec{x})}$ as all points equally close as or closer than the farthest realisation among the k_n with respect to the same norm. Thus we have all quantities necessary to define the discriminant functions which may be inserted into decision rule 1 or 2 to obtain the Bayes' decision rule using the kNN distribution estimate:

$$\begin{aligned} g_i(\vec{x}) &\overset{(6.4)}{=} f_{\vec{X}|\Omega=\omega_i}(\vec{x}|\Omega=\omega_i)p_\Omega(\omega_i) \\ &\approx \widehat{f_{\vec{X}|\Omega=\omega_i}}(\vec{x}|\Omega=\omega_i)p_\Omega(\omega_i) \\ &\overset{(6.21),(6.35)}{=} \frac{k_n p_\Omega(\omega_i)}{n_i V_{\mathfrak{R}(\vec{x}),i}} \\ &\overset{1}{\equiv} \frac{p_\Omega(\omega_i)}{n_i V_{\mathfrak{R}(\vec{x}),i}} \end{aligned} \tag{6.38}$$

where $V_{\mathfrak{R}(\vec{x}),i}$ is the just described region of class ω_i containing k_n realisations. In particular, if we estimate the *a priori* probabilities as the corresponding proportions in the learning dataset, we obtain the following discriminant function:

$$g_i(\vec{x}) \overset{(6.38)}{=} \frac{p_\Omega(\omega_i)}{n_i V_{\mathfrak{R}(\vec{x}),i}} = \frac{1}{n_i V_{\mathfrak{R}(\vec{x}),i}} \frac{n_i}{n_L} \overset{1}{\equiv} V^{-1}_{\mathfrak{R}(\vec{x}),i} \tag{6.39}$$

and the decision rule will simply be to decide the class with minimum distance to the k_n'th nearest neighbour measured in terms of some norm.

It is not difficult to find cases in which the decision rule based on the discriminant function in equation (6.38) or (6.39) decides a different class than the kNN decision rule 4. The main difference of these classifiers is that the kNN distribution estimation classifier will evaluate c times more realisations than the kNN classifier for the same value of k_n. If we accommodate for this difference, then we can expect the classifiers to perform similarly in all practical situations, though there will still exists cases where the classifiers decides differently.

Nearest neighbour classifier

The *nearest neighbour (NN)* classifier is a special case of the kNN classifier with $k_n := 1$. The decision rule of this classifier is:

Decision rule 5 (NN decision rule)**.** *For all $\vec{x} \in \mathfrak{D}$, classify the pattern to the class ω_i for which $\min_{k=1,\ldots,n_i} \|\vec{x}_{i,k} - \vec{x}\| \leq \min_{k=1,\ldots,n_j} \|\vec{x}_{j,k} - \vec{x}\|$ for all $j = 1, \ldots, c$ (if there are multiple choices for ω_i, any will do).*

This decision rule can be trivially stated as: *classify each point to the class of the nearest realisation with respect to some norm.*

This choice of k_n will as discussed result in the most complex possible kNN classifier. As the number of realisations within the region is constant, its corresponding distribution estimate will not converge to the desired distribution even with an infinite number of realisations, see the necessary condition in equation (6.36). It is on the other hand intuitive and simple, and it can furthermore be shown that its asymptotic PMC, P_∞^{NN}, is in the interval $[P_{\mathfrak{B}}, P_{\mathfrak{B}}(2 - P_{\mathfrak{B}}c/(c-1))] \subset [P_{\mathfrak{B}}, 2P_{\mathfrak{B}}]$, where $P_{\mathfrak{B}}$ is as always the asymptotic PMC of Bayes' classifier, so the asymptotic performance is reasonable [13, p.182].

6.3 Overfitting

If we were facing the idealised case with an infinite number of learning patterns in each class, it would be unwise to limit the classifier complexity, e.g. by limiting the number of features or by assuming a specific distribution class for all conditional pdfs. Indeed, it is well known that an increase in the number of features will never increase (and will decrease if the added feature is at least slightly meaningful) the true optimal PMC; it is theoretically impossible as the best classifier would in the worst case only ignore the newly added features. In fact, if we keep adding just random features, we must by pure chance add some that are at least slightly meaningful, thus, if we add enough, we will always obtain an optimal PMC of zero. Similarly, if the best classifier in a specific case was obtained by using e.g. normal conditional pdfs with equal covariance matrices, then the designed classifier with arbitrary covariance matrices would have used equal covariance matrices and the nonparametric classifiers with a parameter chosen to guarantee convergence will result in the same normal distributions with equal covariance matrices, which both indicates that assuming more restrictive distribution classes in the design of the classifier can only have a negative (increasing) effect on the optimal PMC. In fact, if we have an unlimited number of learning patterns, we would always have chosen a nonparametric classification method with an appropriately chosen parameter to guarantee convergence, or even a more general classification method if the origin of the feature vectors is questionable, and fed it with as many features as we could.

The problem with a high classifier complexity first becomes evident when we design our classifiers based on a finite number of learning patterns. Since the classifier is then based on a limited number of feature vectors, we must estimate its required quantities, e.g. the independent parameters of a specific distribution class or the entire distributions. Such estimates are typically based on the feature vectors, which makes the accuracy of the estimates highly correlated with the number of feature vectors. Thus the number of learning patterns and the 'allowed' complexity of the classifier is also highly correlated. If the complexity is greater than the number of patterns 'allows', then the classifier will *generalise* poorly, i.e. the classifier's ability to correctly classify novel patterns is much worse than its ability to classify the patterns in the learning dataset, and we say that the classifier is *overfitted* to its learning dataset [59, p.92].

The connection between the number of patterns and the 'allowed' complexity can be understood by assuming that there is an upper limit of the allowed total error due to the estimation; if the total error exceeds this values, then the expected PMC increases. When using a small learning dataset to design the classifier, each estimate is (on average) highly inaccurate, thus only a few estimates can be obtained before the maximum allowed total error is reached. If we on the other hand design the classifier based on a large learning dataset, each estimate is (on average) rather accurate, thus many estimates can be obtained before the maximum allowed total error is reached.

Of course, the effect of inaccurate estimates are in truth gradually increasing the expected PMC. The idea is however correct, in particular, more learning patterns will allow a higher classifier complexity without suffering from being overfitted. This is also recognised in the literature where some has shown that the expected PMC is mainly determined by the ratio between the number of learning patterns to the number of features [55, p.667] and other states that this

ratio should be larger for more complex classification methods [45, p.135]. There are however also other concerns affecting how complex a classifier can be before it significantly increases the expected PMC. In particular, the true distribution of the conditional pdfs and the effectiveness of the features are relevant [56, p.259].

Let us summarise. As the complexity of the classifier increases, the estimations become gradually more unreliable which in turn affects the expected PMC negatively. On the other hand, the increased complexity typically have a positive effect on the expected PMC, e.g. due to the inclusion of new relevant features or by relaxing incorrect assumptions of the conditional pdfs. The total effect of the expected PMC is what is of interest, but this depends on how the complexity of the classifier has increased.

In an idealised case with two classes and a single discrete feature, Hughes [24, pp.56–59,62–63] showed how the expected PMC and the asymptotic PMC of Bayes' classifier depends on the number of possible values of the discrete feature when no *a priori* information of the conditional pdfs is present. Because it is possible to reasonably quantify continuous features and because multiple discrete features can be compressed into a single discrete feature without loosing any information, the number of possible values of a discrete feature is a reasonable measure of the classifier complexity caused by the features. Figure 6.6 visualises the dependency for the case of equal *a priori* probabilities. This figure shows that the optimal classifier complexity increases with the number of learning patterns. Even more interesting is that all expected PMC curves are strictly unimodal, thus it should be possible to approximately find the optimal classifier complexity if we are able to reliably estimate the performance.

With the knowledge that more features and more general classification methods tend to increase the resulting complexity, and the indication in figure 6.6 that the optimal classifier complexity may be prominent enough to be reasonably estimated, we may attempt to gradually increase the classifiers complexity in the best way we can until we suspect that the expected PMC will decrease if we continue to increase the complexity. Such a procedure is highly dependent on accurate *a priori* information or on a reliable estimation of the expected PMC. By applying improper estimates, which are typically overoptimistic, one may be confused to believe that the classifier is performing well when it severely suffers from being overfitted. We will discuss this and related problems closer when we review evaluation in general in section 6.6.

To reduce the complexity of the classifier and thus the risk of overfitting, we can both use less features and apply a simpler classification method. Because it may be difficult to manually selection a good, but small set of features, much effort has been made to allow automatic reduction of the number of features from an initially larger set. Such methods can broadly be categories in two main classes; the *dimension reduction* methods which transforms a feature space into a feature space of lower dimension and the *feature selection* methods which attempts to select the best features from a larger set of features. We shall briefly describe both these method classes in the following two sections.

Before describing the feature reducing methods, a couple of notes should be made. Firstly, even though the methods can be useful to provide better generalisation, they may also result in a decreased performance and will in particular never decrease the asymptotic PMC. Secondly and most importantly, while the objective of reducing the number of features is to reduce the risk of overfitting,

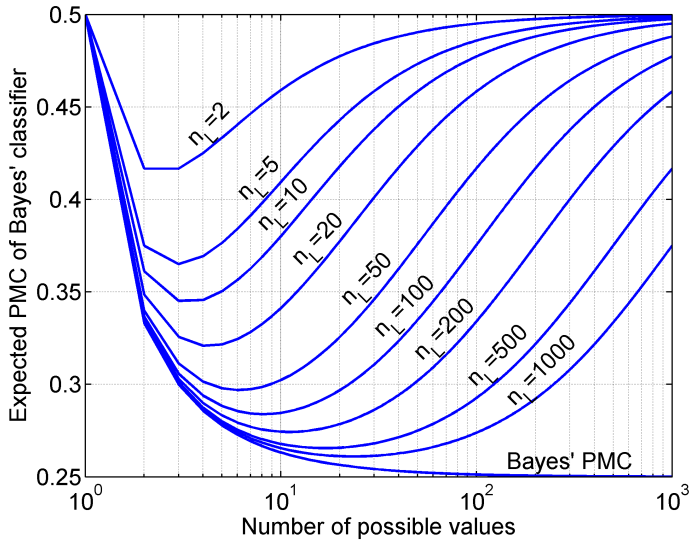

Figure 6.6: Visualisation of the dependency between the expected PMC of Bayes' classifier and the number of possible values of a discrete feature in an idealised case. The same dependency of the asymptotic PMC of Bayes' classifier, labelled Bayes' PMC*, is also included for reference.*

the use of a large initial set of features may also cause overfitting. In this case, the overfitting is not caused by a too high classifier complexity. Instead, this potential cause of the poor generalisation is that a large number of feature candidates increases the probability of obtaining features which by chance performs good or best on the specific learning or tuning dataset, but which is not the true optimal features of the initial feature set. For the case of feature selection methods, a study by Schulerud and Albregtsen [59, pp.93,97] has shown that this is especially critical if the number of learning or tuning patterns is low (less than 200). To reduce this cause of overfitting, one must reduce the number of feature candidates by non-statistical methods or, if possible, increase the number of learning or tuning patterns [59, p.97].

6.4 Dimension reduction

Dimension reduction is the general class of methods that transforms a feature space into a feature space of lower dimension. It deviates from feature selection in that the features forming the output feature space are generally combinations of the features forming the input feature space. The input feature space is often $\mathfrak{D}$, as formed by the set of all d features, but can also be any subset of this feature space, $\mathfrak{D}' \subseteq \mathfrak{D}$, as formed by a subset of the features.

Dimension reduction techniques that are based on linear transformation are practically attractive due to their computational and analytical tractability [13, p.114]. The two most commonly mentioned dimension reduction techniques that will be described in the following are both based on linear transformations.

6.4.1 Fisher's linear discriminant

Fisher's linear discriminant (FLD) is a linear transformation of a feature space $\mathfrak{D}'$ onto $\mathbb{R}$. Any such transformation may be expressed as different choices of $\vec{w}$ in the following equation:

$$y_i = \vec{w}^T \vec{x}_i \text{ for } i = 1, \ldots, n \tag{6.40}$$

This equation may be viewed as the projection of the feature vectors $\vec{x}_i$ for $i = 1, \ldots, n$ on the line that passes through origin and is parallel with the vector $\vec{w}$.

In the following we will assume that the number of classes is two as this is the case for the material under study in this thesis, but it should be noted that a generalisation of the method exists, called *multiple discriminant analysis (MDA)*, that allows any number of classes, see [13, pp.121–123].

FLD finds the optimal value of equation (6.40) with respect to maximising the following criterion function: [13, p.119]

$$J(\vec{w}) := \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \tag{6.41}$$

where $\tilde{m}_i$ and $\tilde{s}_i^2$ is the mean and scatter of the projected samples belonging to class ω_i, and the *scatter* is defined as:

$$\tilde{s}_i^2 := \sum_{y \in \mathfrak{Y}_i} (y - \tilde{m}_i)^2 \tag{6.42}$$

where $\mathfrak{Y}_i$ is the set of projection of each value in $\mathfrak{X}_i$, $i = 1, 2$.

Since $\tilde{s}_i^2$ is the sample variance estimate with the exception of the normalisation constant $(n_i - 1)^{-1}$, the denominator in equation (6.41) is an estimate of the pooled variance. In fact, because the standard pooled variance estimate in statistics weights the samples' variance estimates with a factor proportional to the inverse of the normalisation constant, see equation (3.8), $\tilde{s}_1^2 + \tilde{s}_2^2$ is precisely the pooled variance estimate recommended in statistical theory with the exception of the normalisation constant $(n_1 + n_2 - 2)^{-1}$. Thus, equation (6.41) is actually the squared Mahalanobis distance between the classes, see equation (3.6), where the common variance has been estimated using the standard pooled variance estimate with the exception of a normalisation constant which is irrelevant for the matter of optimisation.

From the derived relation above, we note that maximising the criterion function in equation (6.41) makes sense when the projections are close to symmetric and similarly distributed with the exception of possibly different expectation. It will then choose the vector $\vec{w}$ that attains the maximum expected intermediate difference with respect to the standard pooled variance estimate, commonly stated as attaining the maximum ratio of the *between class variance* to the *within class variance*. An essential property of $\vec{w}$ is that it is independent of its length because any scaling of $\vec{w}$ would cause the same squared factor to be added to both the numerator and the denominator.

It can be shown that the maximum of the criterion function in equation (6.41) satisfies the following equation [17, pp.179,181], [13, pp.119–120]:

$$S_W \vec{w} = \vec{m}_1 - \vec{m}_2 \tag{6.43}$$

where $\vec{m}_i$ is the sample mean of the samples belonging to class ω_i and the *within class scatter matrix* S_W is defined as the sum of the two *class scatter matrices*, which is defined as:

$$S_i := \sum_{\vec{x} \in \mathfrak{X}_i} (\vec{x} - \vec{m}_i)(\vec{x} - \vec{m}_i)^T \tag{6.44}$$

Equation (6.43) will normally result in a unique solution of $\vec{w}$ when the number of samples is greater than the number of initial features, i.e. $n > d$. In this case, we can write the solution of Fisher's criterion function explicitly as:

$$\vec{w} = S_W^{-1}(\vec{m}_1 - \vec{m}_2) \tag{6.45}$$

It is indeed interesting to note that this expression, and thus the optimal separation with respect to a reasonable criterion function, is identical to the separating hyperplane obtained in the parametric method when assuming normality and equal covariances, see equation (6.13), when we further assume that we use the standard or ML estimate of the covariance matrix. As already commented, this indicates that the classifier obtained by using this separation is not strongly dependent on the normality assumption, a hypothesis which is substantiated by practical experiments.

It is also interesting to note that the same separation is also obtained in an optimisation method call *minimum squared error (MSE)* under a reasonable further assumption of its parameter, but as we have chosen not to describe such optimisation methods in this thesis, we will only refer the interested reader to [13, pp.239–243] for details about the method and the proof of the relation.

6.4.2 Principal component analysis

Principal component analysis (PCA) or *Karhunen-Loève expansion* is the best known linear transformation used for dimension reduction and can be used to reduce the dimension of the feature space, d, to an arbitrary dimension d'. Let us define $X := (\vec{x}_1^T, \ldots, \vec{x}_n^T)$ as the matrix containing all feature vectors as rows and $H \in \mathbb{R}^{d,d'}$ as the matrix containing the eigenvectors of the d' largest eigenvalues of the covariance matrix as columns. The PCA method can then be defined as:

$$Y := XH \tag{6.46}$$

where $Y \in \mathbb{R}^{n,d'}$ will contain the resulting dimension reduced features as its rows. [26, p.12]

The idea of projecting the feature space onto the subspace spanned by the dominating eigenvalues should seem reasonable to those familiar with linear algebra. It can also be shown that this is the best d'-dimensional projection with respect to the sum of squared error of the collection of differences between the feature vectors and their global mean [13, pp.116–117]. In this sense, the method is optimal, but the downside is that we do not wish to represent the data optimally, we want to discriminate between the data of different classes. Since the described PCA method completely ignores the class information, it can be expected to have low discrimination value in general. As an example, a good representation of digital images of the letters O and Q are probably very similar, but may be useless to discriminate between the letters.

FLD optimised the separation between the classes (with respect to an intuitive criterion function) and thus attempted to maximise the discrimination

between the classes. The PCA method can borrow some inspiration from this approach. In particular, it is easy to show that the Fisher's criterion function can be written as $\vec{w}^T S_B \vec{w} / \vec{w}^T S_W \vec{w}$ [13, p.120], where S_W is the already defined within class scatter matrix and S_B is the *between class scatter matrix* which is defined as:

$$S_B = (\vec{m}_1 - \vec{m}_2)(\vec{m}_1 - \vec{m}_2)^T \tag{6.47}$$

We may thus replace the covariance matrix in the PCA method with $S_W^{-1} S_B$ with the justification that this matrix should include the discrimination value which is used by the FLD. Because we expect the discrimination value of the FLD to be reasonable, we can also expect reasonable separation when letting the PCA method extract the eigenvectors corresponding to the largest eigenvalues of the matrix $S_W^{-1} S_B$. It should be noted that other generalisation of the PCA method that also attempts to maximise the discrimination value of the dimension reduced feature space is possible. [26, pp.12–13]

6.5 Feature selection

In feature selection, we attempt to select the subset of features from a larger set of features that results in the best performing classifier, e.g. attains the lowest possible expected PMC among a set of reasonable classification methods. This sounds easy enough, but simple combinatoric reveals that there exists $\sum_{i=1}^{d} d!/(i!(d-i)!)$ possible subsets of d features, a number which is practically prohibitive even for moderate values of d [45, p.119]. One could thus hope that there exist some intermediate relationships among the subsets that could be exploited in order to efficiently find the best subset. Unfortunately, no such relationship exist for arbitrary features[1] and the only way to guarantee the discovery of the best subset is a complete inspection of all subsets [7, p.657]. If this is practically undesirable, then a generally suboptimal procedure may be applied. Many such procedures have appeared in the literature, but we will only superficially discuss the general problem before we direct our focus to one particular criterion function which is related to entropy [56, p.259].

As we because of a limited number of learning patterns are not guaranteed a decreased expected PMC by adding features, even when the features are meaningful, it is natural to select the best features first. It has however long been recognised that the subset of k features attaining the lowest expected PMC are in general not the k individually best features [15, p.669; 56, p.259]. This is partly due to correlations between features; if e.g. the object radius is the best feature, then both the major and minor axis are likely to also be good features. The essence of the problem is however retained even for independent features [6, p.116]. Thus we must pay attention to the cooperation of features in general.

A natural question to ask is how many features should optimally be included. The answer depends on the number of learning patterns, the classification method and the ability of the features to separate classes. In practise we will not be able to correctly compare the true performance of each feature and even less when comparing sets of features, thus the ordering of the features

[1] *Branch and bound search* is guaranteed to find the optimal subset of features if the criterion function is monotone. While only a fraction of all feature subsets are evaluated in this method, the worst case performance is still exponential. In addition, most criterion functions do not satisfy the monotonicity property. [26, pp.15–16]

are a fourth highly relevant factor, maybe even the most important one, which affects the optimal number of features. [56, p.259]

In practise, a general idea of feature selection is based on gradually increasing and/or decreasing the number of features to find the best forming subset of the initial set of features. In each step, the algorithm would use any of some dozens of criterion functions to attempt to find the optimal set of features with the specific cardinality [56, p.259]. The estimated optimal set found in each step will typically depend on the last estimated optimal set due to computational tractability.

The *selection method* is the search strategy applied in the feature selection method. The easiest approach with respect to implementation and computation is a *sequential search*, which is a search that at each step either adds [69] or subtracts [37] the required number of features (typically one) to or from the previous feature set. However, this will in general provide poor results, which is a natural consequence of the discussed feature cooperation. An algorithm that combines the approaches is the *plus l - take away r* strategy. This algorithm has been generalised to allow the number of included and eliminated features, l and r, change in each step, a technique known as *floating search* [53], which has been further refined to a technique known as *adaptive floating search* [62]. Naturally, these generalised approaches will required more computation than the straight sequential searches, but the computational burden is typically manageable and far less than for the exhaustive search. [26, p.15]

The choice of criterion function is important [45, p.119]. A natural and commonly used choice is any estimate of the expected PMC [45, p.119]. Obtaining a proper estimate of the expected PMC is however not easy and will in general require the use of a tuning dataset, a set independent of both the learning and validation dataset, which in turn will either affect the expected PMC (as discussed in section 6.3) or accuracy of the classifiers estimated PMC, see section 6.6 for details. Thus many other criterion functions have also been proposed [56, p.259].

To restrict the extent of this thesis, we will only briefly look into a couple of other criterion functions that are related to already discussed quantities. First out is to maximise an estimate of the Mahalanobis distance between the classes, see e.g. the general definition for the case of two classes in equation (3.3). As the compared classes will here be assumed to have a common covariance matrix and be measured in terms of (the inverse of) this matrix, this criterion function is reasonable only when the conditional pdfs can be said to be approximately symmetric and equal with the exception of a possibly different expectation. In this case, it follows from the discussion about FLD (see section 6.4.1) that this is a reasonable criterion function to maximise when we use an estimate of the covariance matrix that is proportional to the within class scatter matrix, e.g. the standard or ML estimate of the covariance matrix.

Another criterion function is based on *mutual information*, which for two continuous random variables X and Y is defined as [51, p.1226]:

$$I(X,Y) := \int_Y \int_X f_{X,Y}(x,y) \log \frac{f_{X,Y}(x,y)}{f_X(x) f_Y(y)} dxdy \tag{6.48}$$

where the base of the logarithm is arbitrary and e.g. $\int_X (\dots) dx$ denotes the integration with respect to x over all possible values of the random variable X.

We can easily generalise this definition to include a vector of $k \geq 2$ continuous random variables, $\vec{X} := (X_1, \ldots, X_k)$:

$$I(\vec{X}) := \int_{\vec{X}} f_{\vec{X}}(\vec{x}) \log \frac{f_{\vec{X}}(\vec{x})}{f_{X_1}(x_1) \cdot \cdots \cdot f_{X_k}(x_k)} d\vec{x} \tag{6.49}$$

where $\int_{\vec{X}}(\ldots)d\vec{x}$ denotes the multiple integral of all the continuous random variables in $\vec{x}$ over all possible values of the random variable $\vec{X}$. The mutual information of *discrete* random variables is defined by replacing the integrals in the above equations with sums.

The mutual information is related to the Shannon entropy. Let us show the relation for the case of a continuous random variable $\vec{X}$ of dimension two or greater (the mutual information is undefined in the case of a single dimension). The Shannon entropy of $\vec{X}$ is defined as:

$$H(\vec{X}) := \int_{\vec{X}} f_{\vec{X}}(\vec{x}) \log(f_{\vec{X}}(\vec{x}))d\vec{x} \tag{6.50}$$

Because the dimension of $\vec{X}$ is at least two, it can be defined as $\vec{X} := (\vec{Y}, \vec{Z})$ where $\vec{Y}$ and $\vec{Z}$ are continuous random variables, each with dimension of at least one. Starting with the general definition in equation (6.49), we obtain the relation between the mutual information and the Shannon entropy as follows:

$$\begin{aligned} I(\vec{X}) = I(\vec{Y}, \vec{Z}) &= \int_{\vec{Z}} \int_{\vec{Y}} f_{\vec{Y},\vec{Z}}(\vec{y}, \vec{z})(\log(f_{\vec{Y},\vec{Z}}(\vec{y}, \vec{z})) - \log(f_{\vec{Y}}(\vec{y}) f_{\vec{Z}}(\vec{z})))d\vec{y}d\vec{z} \\ &= -H(\vec{Y}, \vec{Z}) - \int_{\vec{Z}} \int_{\vec{Y}} f_{\vec{Y},\vec{Z}}(\vec{y}, \vec{z})(\log(f_{\vec{Y}}(\vec{y})) + \log(f_{\vec{Z}}(\vec{z})))d\vec{y}d\vec{z} \\ &= -H(\vec{Y}, \vec{Z}) - \int_{\vec{Y}} f_{\vec{Y}}(\vec{y}) \log(f_{\vec{Y}}(\vec{y}))d\vec{y} - \int_{\vec{Z}} f_{\vec{Z}}(\vec{z}) \log(f_{\vec{Z}}(\vec{z})))d\vec{z} \\ &= H(\vec{Y}) + H(\vec{Z}) - H(\vec{Y}, \vec{Z}) \end{aligned} \tag{6.51}$$

Define $\vec{S}_k$ as the random variable of k specific features selected from the initial feature set and let $\vec{s}_k$ be a realisation of these features, i.e. a subvector of a possible feature vector. By inserting this vector and the discrete random variable giving the true *a priori* probabilities, Ω, into the definition of mutual information, we obtain [51, p.1227]:

$$I(\vec{S}_k, \Omega) := \sum_{i=0}^{c} \int_{\vec{S}_k} f_{\vec{S}_k,\Omega_i}(\vec{s}_k, \omega_i) \log \frac{f_{\vec{S}_k,\Omega_i}(\vec{s}_k, \omega_i)}{f_{\vec{S}_k}(\vec{s}_k) p_{\Omega_i}(\omega_i)} d\vec{s}_k \tag{6.52}$$

The function $I(\vec{S}_k, \Omega)$ can be directly used as a criterion function in feature selection that should be attempted to be maximised, a scheme called *maximal dependency (max-dependency)*. However, accurately estimating $f_{\vec{S}_k}(\vec{s}_k)$ and $f_{\vec{S}_k,\Omega}(\vec{s}_k, \omega)$ is troubling when using a limited number of learning patterns, both because the feature (sub)space becomes sparse and, when assuming multivariate normal distribution, the covariance matrix becomes ill-conditioned (see [33, pp.101–104] for a general discussion of ill-conditioning). A popular estimation of this scheme is called *maximal relevance (max-relevance)*, in which the joint

mutation information is estimated simply as the mean of the individual mutual informations, both also with respect to the classes, i.e.: [51, pp.1226–1227]

$$\hat{I}(\vec{S}_k, \Omega) := D(\vec{S}_k, \Omega) := \frac{1}{k} \sum_{X_i \in \vec{S}_k} I(X_i, \Omega) \tag{6.53}$$

However, as this estimate does not take the cooperation of different features into account, the discussed phenomenon of correlated selected features will be a problem when using this criterion function. To accommodate for this, Ding and Peng [12] defined the *minimum redundancy (min-redundancy)* of any two features as:

$$R(\vec{S}_k) := \frac{1}{k^2} \sum_{X_i, X_j \in \vec{S}_k} I(X_i, X_j) \tag{6.54}$$

and proposed the *minimal-redundancy-maximal-relevance (mRMR)* criterion function, which is any combination of D and R, e.g. $D - R$. If using the simple forward sequential search, often called *sequential forward selection (SFS)*, and adding only a single feature at each step, it can be shown that this criterion function yields the same result as the max-dependency criterion function [51, p.1228].

While there have been proposed many selection methods and criteria of feature effectiveness, it is unclear whether one specific strategy performs consistently better than most others [56, p.260]. Whichever strategy is applied, one must be aware of the overfitting problem that was discussed in section 6.3. In particular, one should only include potentially relevant features in the initial feature set. Also, one should pay attention to the location of the peak where the true expected PMC begin to decrease with increasing classifier complexity. Indeed, it has been reported that this is important and even more so than e.g. determine the optimal number of features to be selected (which we mentioned is not easy to estimate anyway and will in particular depend on the specific features). Moreover, the peak of the expected PMC against the number of features is typically nearly flat, thus an accurate determination of the optimal number of features is likely to be of minor importance relative to locating the peak in the expected PMC against the classifier complexity. [56, p.259]

6.6 Evaluation

We have thus far in this chapter discussed how a learning dataset should be applied to design a reasonable classifier. Designing the classifier is however not enough, we also need to be able to properly evaluate the performance of the classifier. In this context, all mentioned PMC are of interest, but more specific quantities that e.g. depend on a particular class are also interesting. We will in this section limit our attention to the estimation of PMCs, but note that similar comments, especially about the distinction between conditional, expected, asymptotic and optimal estimates and the uncertainty of the estimators, could be applied to most performance estimates of the classifier. In section 6.7.1 we will look into some other commonly used performance estimates when we discuss how we are going to report the classification result in this study.

The conditional PMC is the total performance of a particular classifier and can be said to be the single most essential quantity if the classifier should be

used in practise. The expected PMC is the total performance we can expect of a similarly designed classifier, but with another equally sized learning dataset. This quantity is thus likely to be the most informative quantity for others with a similar learning dataset, because it reports the total performance they can expect when applying our method on their dataset. The asymptotic PMC is the total performance of the same classifier when designed using an unlimited number of learning patterns and can be seen as the optimal total performance of our method. Finally, the optimal PMC is the optimal total performance of our features with infinite learning patterns and an arbitrary classification method. This PMC can be used both to measure the discrimination value of our features and to relate all the three other total performances to the theoretically optimum.

Obtaining any estimate of a PMC is however not easy. There exists to date four main approaches [56, p.257]:

- *Resubstitution*: An optimistically biased estimate of the asymptotic PMC is obtained by using the learning dataset to estimate the performance of the classifier. It could also be viewed as a very optimistic estimate of the conditional PMC (with $n_L = n$) as it only evaluates the classifier using a single validation dataset of finite size. This approach allows the classifier to use all available patterns for both learning and evaluation, but there are no statistical independence between the learning dataset and the validation dataset. It can be shown both analytically and experimentally that the bias is approximately equal to the difference between the expected and asymptotic PMC for the parametric classifiers that assumes normality and independent features with equal variances (case 1) or equal covariance matrices (case 2). Thus, given an estimate of the expected PMC, we can obtain a more reliable estimate of the asymptotic PMC by averaging this estimate and the corresponding resubstitution estimate. [56, p.258]

- *K-fold cross-validation*: The entire dataset is partitioned in K (approximately) equal subsets. For each $k = 1, \ldots, K$, the classifiers is designed using the $K - 1$ other subset and evaluated on the last, k'th subsets. The average of the K evaluations are used as an estimate of the expected PMC with $n_L = n - K$ [45, p.135; 21, p.241]. An optional variation is the *repeated K-fold cross-validation* where the process is repeated for different partitionings and the average (of the averages) is given as the results [25, p.1961]. Another option is called *stratified K-fold cross-validation*, here the proportions in each of the K subsets (approximately) reflect the corresponding proportions in the complete dataset. The specific choice of $K = n$ is popular [56, p.257] and is known as *leave-one-out cross-validation* [31, p.4]. [4, p.375]

- *Bootstrap*: A general method based on statistical bootstrapping that estimates the expected PMC. The patterns are bootstrapped B times. In *bootstrapping with replacement*, the size of each bootstrap is n, and these n patterns are assigned as the learning dataset. Because we are sampling *with* replacement and are only acquiring the same number of samples as the total number of patterns, each learning dataset is likely to contain multiple copies of some patterns and none of other. For each learning dataset, the patterns that are not included in this dataset is set to be the validation dataset (with only a single copy of each pattern) [21, p.251]. On

average, the probability that any particular pattern is contained in any particular learning dataset is about 63.2 %, thus the validation dataset will on average contain 36.8 % of the entire dataset [21, p.251]. In *bootstrapping without replacement*, each bootstrap separates the entire dataset into a learning dataset of a specified size n_L and a validation dataset (of size $n_v = n - n_L$).

For both bootstrap methods, B classifiers are designed based on each bootstrapped learning dataset. An estimate of the expected PMC (with about $0.632n$ or n_L learning patterns for sampling with or without replacement, respectively) is obtained by averaging the misclassification rate for each pattern. This estimate is sometimes called the *leave-one-out bootstrap estimate* [21, p.251]. Another estimate of the same expected PMC is obtained by averaging the misclassification rate of each bootstrap. A small value of B will make the estimates unreliable, but a value in range of 25-200 has been reported to seem quite adequate [14, p.317] and current literature also uses such values, e.g. [4, p.377] and [21, p.249] indicates the used of 100 bootstraps.

As with cross-validation, the bootstraps may optionally be stratified. An alternative option is to let the class proportions of each bootstrapped learning dataset be (approximately) equal; we will define this option as *evened*. The bootstraps can also be *balanced* in the meaning that each pattern appears the same number of times in the learning dataset [4, p.375]. This corresponds to appearing B or Bn_L/n times in the learning dataset for sampling with and without replacement, respectively. We lastly define the *evened and balanced* bootstraps as evened bootstrapping where each pattern of any particular class appears the same number of times in the learning dataset. The standard definition of balanced is only satisfied by such bootstraps if the proportions in the complete dataset are equal, but this definition does in general extend the meaning of balancing under the restriction of evening.

- *Holdout-validation*: The dataset is separated prior to analysis, either randomly or manually without knowledge of the true classes and while allowing similar variations in each subset, into a learning dataset of size n_L and a validation dataset (of size $n_V = n - n_L$) with no intersecting patterns. The classifier is designed based on the learning dataset and its PMC on the validation dataset is used as an estimate of the conditional PMC (with n_L learning patterns). While this method can be said to not efficiently use the dataset, it provides the only rigorous yet practical estimate of a PMC [25, p.1965].

Let us now attempt to discuss the different evaluation methods. It should first be noted that the estimates will depend on several factors. For starters, it will depend on the classification method. This is easily illustrated by considering the NN classifier with the resubstitution method used for evaluation. Obviously, the estimated PMC will be zero even for completely meaningless features. This actually indicates a general problem with the resubstitution method; its optimistic bias will in general increase with the classifier complexity. In fact, its estimate will for most classification methods rather quickly converge to zero when adding even random features while the true conditional and expected PMC

can converge to randomness, i.e. $100(1 - 1/c)\%$. Other dependencies are also important for estimates and these include, but are not limited to, the particular dataset under study, the true PMC values and the complexity of the classifier.

Two basic ground rules for evaluating the performance of a classifier has been suggested by Schulerud et al. [60, p.77]. The first is the use of an acceptable experimental design, which for image analysis can be understood as separate learning and validation dataset [60, p.77]. In this sense, the resubstitution method falls through. The cross-validation and bootstrap[2] methods will however fulfil this requirement as long as any method that depends on the learning dataset, e.g. the adaptively chosen weight array in an adaptive texture feature, the feature dimension reduction, the feature selection or the adaptive learning of the classifiers parameters like for instance the estimated covariance matrices, the window width or k, is applied for each subset or bootstrap, respectively, and not e.g. once using the entire dataset [21, pp.241,245–246]. The requirement is obviously fulfilled for the holdout-validation method.

The second ground rule is a proper use of the statistical methods [60, p.77]. Typically, the statistical methods require independence within both the learning and the validation dataset. In this sense, the bootstrap method with replacement falls through because the learning dataset may contain multiple samples of the same pattern. For the case of nuclear image analysis, this also requires the patient to be the analytical unit due to dependences of the cells, see section 2.3.2.

Using these two ground rules, all the mentioned methods are appropriate except the resubstitution method and the bootstrap method with replacement. For the resubstitution and the leave-one-out cross-validation method where features were selected prior to partitioning the dataset (thus violating the first ground rule of independent learning and validation dataset), it has also been shown that an increased number of features results in a higher difference between the true performance and its estimate [60, p.76]. However, these methods are not completely ruled out, in particular, a study shows that an estimate that weights the resubstitution estimate with the leave-one-out bootstrap estimate when sampling with replacements (known as the *0.632 estimator*, see [21, p.251] for details) provides the best overall performance in comparison with the resubstitution method, several cross-validation methods and another bootstrap method [4, p.377].

Criticism has also been directed toward the other evaluation methods. For the holdout-validation method, it has been argued that it uses the dataset inefficiently and that it is in practise difficult to obtain a reliable data dependent variance estimate of this estimator. For the cross-validation and bootstrap methods in general, while it has been highlighted that these methods provide a practically unbiased estimate of the expected PMC [56, p.258], others have shown that its variance is large [25, pp.1961–1963], especially for the cross-validation method [4, pp.378-379] and also in comparison with the variance of the holdout-validation estimate [59, pp.95–97], whenever the number of patterns are in the order of hundreds. It could be noted that none of these critical studies used a bootstrap method without replacement, but other factors may be more severe for the accuracy of the bootstrap estimates, like for instance the expected PMC,

[2]In general, a bootstrap method does not need to fulfil this requirement, but using our definition of the bootstrap method with and without replacement, this requirement is fulfilled.

the feature distributions (which includes the true optimal PMC when we assume that such distributions exists) and the complexity of the classifier.

Investigations have shown that the variance of an estimator of a PMC, $\hat{P}_\eta$, where η indicates the estimation method, is in general of order [56, p.258]:

$$\mathrm{Var}(\hat{P}_\eta) = \frac{E(\hat{P}_\eta)(1 - E(\hat{P}_\eta))}{n_V} \tag{6.55}$$

In practise, we typically insert the concrete point estimate of the PMC for the expectation and then obtain the following estimate [56, pp.261–262]:

$$s^2_{\hat{P}_\eta} := \frac{\hat{p_\eta}(1 - \hat{p_\eta})}{n_V} \tag{6.56}$$

As this is a completely general variance estimate depending only on the point estimate and the number of validation patterns, and in particular on neither the evaluation method nor the classifier complexity, it can not be expected to be accurate in general.

For the cross-validation and bootstrap methods, an estimate of the estimator's variance can be obtained as the variance of the estimated PMC for each subset or bootstrap, respectively. It has been reported that there are many pitfalls with this estimate, in particular that it can be dominated by the variations caused by a small number of validation patterns, dependency between classifiers due to overlapping learning patterns, dependency between evaluations due to overlapping validation patterns and dependencies of the learning and validation patterns of different subsets or bootstraps [70, p.3; 25, p.1963]. A method that addresses the issues with a small number of validation patterns and dependencies between different subsets or bootstraps while estimating the estimator's variance has been proposed, but not compared to the standard cross-validation and bootstrap estimates [70, p.3]. It can be noted that this method begins with dividing the entire dataset in two to avoid the dependencies between subsets or bootstraps [70, p.6], which on the downside results in a less efficient use of the dataset. The method may also be applicable to estimate the accuracy of a holdout-validation estimate, though its variance estimate will be of the expected PMC and use less number of learning patterns than the holdout-validation estimate does.

6.6.1 Partitioning the dataset

We have indicated in section 6.3 and above that both the number of learning patterns and the number of validation patterns are essential to obtain the best possible classifier and estimated performance, respectively. We have also argued that no intersecting patterns in the two datasets should be a ground rule in image analysis. Because we in practise typically only have a limited number of patterns available, following this rule will result in a trade-off problem between a decent classifier and its estimated performance. We will in this section discuss how to reasonably partition the entire dataset with respect to this trade-off problem.

It has been shown [56, p.256] that under the assumption of two classes with normal conditional pdfs, the increase of expected PMC due to finite learning

Table 6.1: Contribution caused by the estimation of particular groups of independent parameters. [56, p.257]

i	θ_i	♯param	Independent parameters to be estimated
1	1	1	The *a priori* probabilities, $f_\Omega(\omega_1)$ and $f_\Omega(\omega_2)$
2	$\frac{\delta^2}{2} + d$	2	The means, μ_1 and μ_2
5	$\frac{\frac{\delta^4}{8} + \frac{d\delta^2}{4}}{1 - \frac{d}{n_L}}$	$\frac{d(d+1)}{2}$	The common covariance matrix, Σ
6	$\frac{2\left(\frac{\delta^4}{8} + \frac{d(d+\delta^2)}{4}\right)}{1 - \frac{2d}{n_L}}$	$d(d+1)$	The covariance matrices, Σ_1 and Σ_2

patterns is:

$$\Delta^\alpha_{n_L} := E(P^\alpha_{n_L}) - P^\alpha_\infty = \frac{f_Z(\delta/2)}{n_L \delta} \sum_{i \in \mathfrak{C}_\alpha} \theta_i \tag{6.57}$$

where α denotes the classification method, δ is the Mahalanobis distance between the classes (see equation (3.4)), $Z \sim N(0,1)$ is the random variable of the standard normal distribution, $\mathfrak{C}_\alpha$ is a set of indexes specifying groups of independent parameters and θ_i is the increased contribution on the difference caused by the estimation of a particular group of independent parameters, see table 6.1. This formula is interesting in several ways. In particular, it clearly shows the dependency on the number of learning patterns and also the effect of increased classifier complexity and the combined discrimination value of the features. We note that the formula depends neither on the values of the *a priori* probabilities nor on the relative discrimination value of each feature.

In equation (6.55), we reproduced a formula for the variance of any estimator of a PMC. This formula can thus provide us with an estimate of the variance of the expected PMC estimator, either by performing classifications iteratively until the optimal partitioning ratio converges, i.e. by inserting each classifier's estimated expected PMC into equation (6.56) and find an updated estimate of the optimal partitioning ratio, or by simply assuming the true value of the expected PMC.

By combining a variance estimate of an expected PMC estimator with the increase in expected PMC caused by a limited number of learning patterns, we obtain a criterion function that we can use to estimate a reasonable partitioning of the dataset. In correspondence with Nielsen et al. [45, p.136], we will assume that the cost associated with increased expected PMC is equal to the cost associated with increased variance of the estimator. We then obtain the following criterion function:

$$J(r) := \Delta^\alpha_{n_L} + \mathrm{Var}(\hat{P}_\eta) = \frac{f_Z(\delta/2)}{\delta n r} \sum_{i \in \mathfrak{C}_\alpha} \theta_i + \frac{E(\widehat{P^\alpha_{nr,\eta}})(1 - E(\widehat{P^\alpha_{nr,\eta}}))}{n(1-r)} \tag{6.58}$$

which we wish to minimise. $r := n_L/n$, the ratio of the number of learning patterns to the total number of patterns, is here chosen as the free variable, but either n_L or n_V could have been used instead.

Let us analytically minimise the criterion function in equation (6.58) for the simple case of assuming normally distributed conditional pdfs with known covariance matrices. This is similar to the assumption of independent features with equal variances (case 1), but such a distribution class is not assumed here because the article presenting the terms in table 6.1 did unfortunately not include the term associated for a single common variance [56, p.257], however, the resulting partitioning with known covariance matrices can be expected to be representative for the case of a single estimated variance too. In any case, we will under the assumption have three independent parameters; one from the *a priori* probabilities and two from the class means. Inserting the corresponding terms from table 6.1 into equation (6.58), we obtain the following criterion function:

$$J(r) = \frac{f_Z(\delta/2)}{\delta n r}\left(1 + \frac{\delta^2}{2} + d\right) + \frac{E(\widehat{P^{\alpha}_{nr,\eta}})(1 - E(\widehat{P^{\alpha}_{nr,\eta}}))}{n(1-r)} \tag{6.59}$$

By differentiating and setting equal zero, we obtain:

$$\begin{gathered}
J'(r) = -\frac{f_Z(\delta/2)}{\delta n r^2}\left(1 + \frac{\delta^2}{2} + d\right) - \frac{E(\widehat{P^{\alpha}_{nr,\eta}})(1 - E(\widehat{P^{\alpha}_{nr,\eta}}))}{n(1-r)^2}(-1) = 0 \\
\Downarrow \\
\frac{E(\widehat{P^{\alpha}_{nr,\eta}})(1 - E(\widehat{P^{\alpha}_{nr,\eta}}))}{(1-r)^2} = \frac{f_Z(\delta/2)}{\delta r^2}\left(1 + \frac{\delta^2}{2} + d\right) \\
\Downarrow \\
E(\widehat{P^{\alpha}_{nr,\eta}})(1 - E(\widehat{P^{\alpha}_{nr,\eta}}))\delta r^2 = f_Z(\delta/2)\left(1 + \frac{\delta^2}{2} + d\right)(1-r)^2 \\
\Updownarrow r \in [0,1], \text{ both square root terms are non-negative} \\
r\sqrt{E(\widehat{P^{\alpha}_{nr,\eta}})(1 - E(\widehat{P^{\alpha}_{nr,\eta}}))\delta} = (1-r)\sqrt{f_Z(\delta/2)\left(1 + \frac{\delta^2}{2} + d\right)} \\
\Updownarrow f_Z(\delta/2) > 0 \text{ when } \delta \text{ is finite} \\
r = \frac{\sqrt{f_Z(\delta/2)\left(1 + \frac{\delta^2}{2} + d\right)}}{\sqrt{f_Z(\delta/2)\left(1 + \frac{\delta^2}{2} + d\right)} + \sqrt{E(\widehat{P^{\alpha}_{nr,\eta}})(1 - E(\widehat{P^{\alpha}_{nr,\eta}}))\delta}}
\end{gathered} \tag{6.60}$$

We can make several interesting comments about this result. Firstly, we note that the ratio is independent of the particular number of patterns (when ignoring its indirect effect on the estimator $\widehat{P^{\alpha}_{nr,\eta}}$). Secondly, we note that as the number of features increases, the optimal partitioning (with respect to the used criterion function) is eventually to use the entire dataset as learning dataset. As we can expect that the variance of the expected PMC estimator is in general extremely high when only a microscopic proportion of the patterns is used for evaluation, this indicates the general negative impact of the classifier complexity on the expected PMC and that this is relevant for even the most simple classification methods.

Equation (6.60) also reveals that the Mahalanobis distance between the classes is highly relevant for the optimal partitioning (with respect to the used

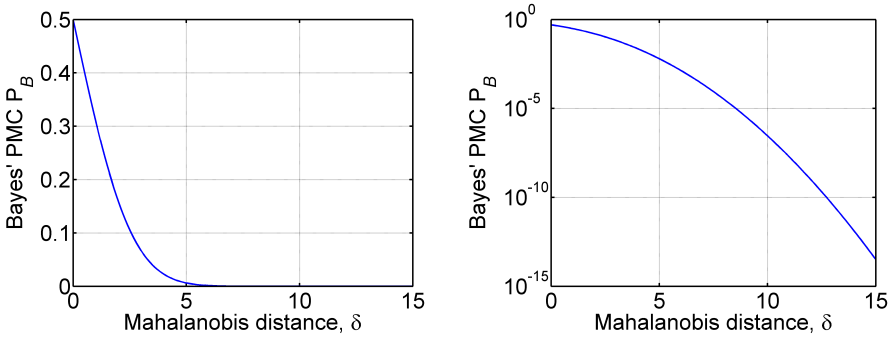

Figure 6.7: The relation between the Mahalanobis distance between the classes and the asymptotic PMC of Bayes' classifier for the case of two equally probable classes with normal conditional pdfs with covariance matrices equal to $I_d/(2\pi)$. In this case, the PMC is independent of the number of features.

criterion function); the optimal partitioning approaches $r = 0$, i.e. to use all patterns for evaluation, as the Mahalanobis distance between the classes increases. However, as the Mahalanobis distance between the classes is a measurement of the combined discrimination value of all features, this is only natural because it is relatively easy to construct a good decision rule when the classes are well separated, so we should be more concerned with obtaining a reliable estimate of the expected PMC. Figure 6.7 shows the relation between the Mahalanobis distance between the classes and the asymptotic PMC of Bayes' classifier for the case of two equally probable classes with normal conditional pdfs (as equation (6.57) also assumes) with covariance matrices equal to $I_d/(2\pi)$.

The result in equation (6.60) and the results of applying equation (6.58) when not assuming known covariance matrices are illustrated in figures 6.8 and 6.9. These illustrations enforces the already stated asymptotic behaviours of the optimal partitioning when increasing the number of features and the Mahalanobis distance between the classes. They however also provide more information.

In light of the relation in figure 6.7, we see from figure 6.8 that the classes must be very well separated before the optimal partitioning decreases below 0.5. We also note the importance of the complexity of the classification methods, even in this case with only four features; higher complexity makes it more important to have many learning patterns because there are relatively many independent parameters that must be estimated. Less expected PMC has the same effect, but this must be viewed in light of a constant Mahalanobis distance between the classes, thus this indicates that the difference between the expected PMC and the asymptotic PMC becomes more significant as the expected PMC decreases, which results in an increased need for learning patterns. We also note that if we increase the number of patterns, most of the new patterns should be assigned as validation patterns and thus the optimal partitioning ratio decreases. Finally, it is interesting to see that the optimal partitioning ratio converges with respect to the Mahalanobis distance between the classes, though it does not happen before the asymptotic PMC of Bayes' classifier is nearly 10^{-10}; at this point it is far more important to reliably estimate the expected PMC than use

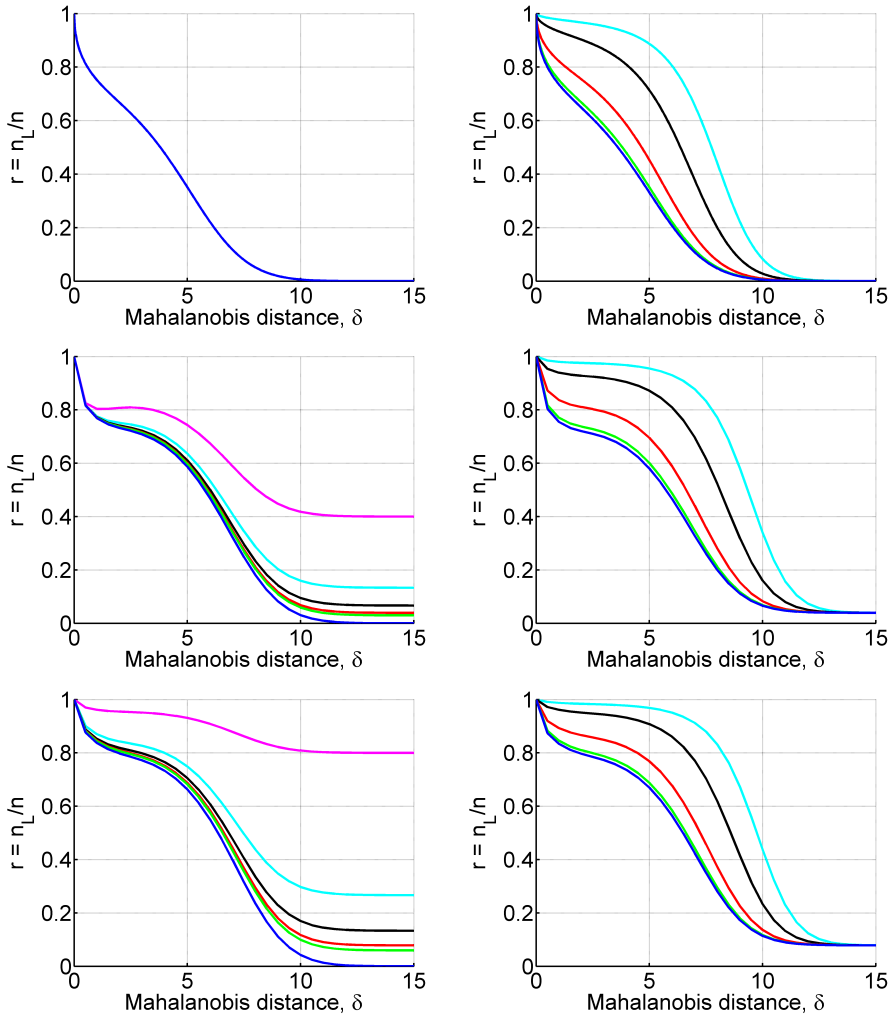

Figure 6.8: Results of minimising the criterion function in equation (6.58) *for different values of the Mahalanobis distance between the classes when there are four features. All conditional pdfs are assumed normally distributed and with; top row) known covariance matrices, middle row) common covariance matrix, and bottom row) arbitrary covariance matrices. In the left column the expected PMC is 0.3 and the total number of patterns are 10 (magenta curve), 30 (cyan curve), 60 (black curve), 102 (red curve), 134 (green curve) and 1000000 (blue curve). The values 102 and 134 included because this is the number of patients in our dataset when excluding and including the tetraploid and polyploid cases, respectively. In the right column the total number of patterns are 102 and the expected PMC is 0.001 (cyan curve), 0.01 (black curve), 0.1 (red curve), 0.3 (green curve) and 0.5 (blue curve).*

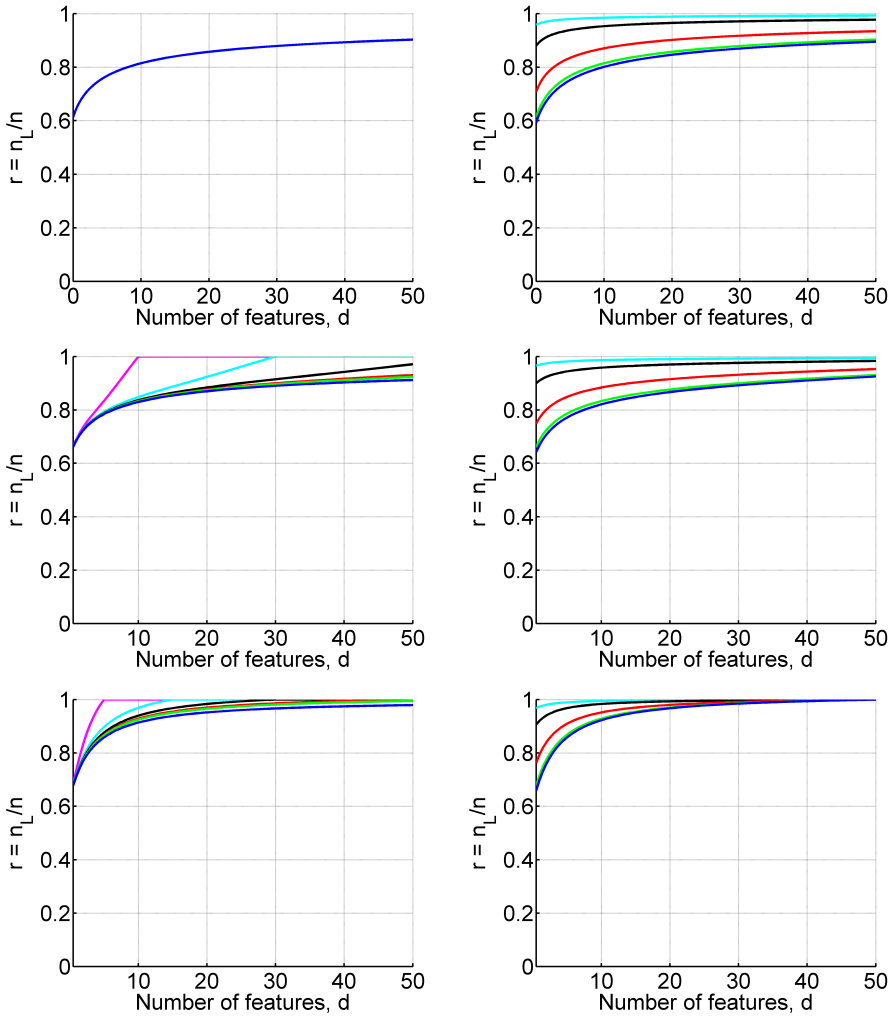

Figure 6.9: Results of minimising the criterion function in equation (6.58) *for different number of features when the Mahalanobis distance between the classes is one. All conditional pdfs are assumed normally distributed and with; top row) known covariance matrices, middle row) common covariance matrix, and bottom row) arbitrary covariance matrices. In the left column the expected PMC is 0.3 and the total number of patterns are 10 (magenta curve), 30 (cyan curve), 60 (black curve), 102 (red curve), 134 (green curve) and 1000000 (blue curve). In the right column the total number of patterns are 102 and the expected PMC is 0.001 (cyan curve), 0.01 (black curve), 0.1 (red curve), 0.3 (green curve) and 0.5 (blue curve).*

many learning patterns to reliably estimate the independent parameters.

Figure 6.9 enforces the comments made on the expected PMC and the number of patterns. It also better illustrates the problem of overfitting, with respect to both the number of features and high complexity of the classification method, as this is indicated by the great need of many learning patterns. As a Mahalanobis distance between the classes of one, which these plots assumes, is representative for many of our feature sets, we note that the optimal partitioning in our case will be to assign most patterns as learning patterns.

Our illustrations only partially corresponds to the stated relation in [45, p.136] that a 50/50 split is optimal if the product of the number of features and the Mahalanobis distance between the classes is approximately 30. While this is a very good approximation in view of figure 6.8, it is far off in view of figure 6.9.

6.7 Classification and evaluation in this study

In this study, we will design a classifier that attempts to estimate the prognosis, i.e. attempts to predict the outcome, of patients suffering from early ovarian cancer. There are two possible prognosis, *bad prognosis* which refers to patients who relapsed ovarian cancer within ten years after surgery and *good prognosis* which are all other patients that survived the ten years without relapse (patients which died of other causes within ten years are excluded prior to obtaining our dataset). We argued in section 2.3.2 that the patients, not the images of some of its nuclei, is the appropriate analytical units. This choice is maintained and enforced with the second ground rule that was mentioned in section 6.6, which requires statistical independence within both the learning and the validation dataset.

All properly discussed classification methods will be applied for comparative reasons. While the use of multiple classification methods is justifiable in a learning phase to locate the best classifier and also get an impression of whether we are on the verge of overfitting (then complex classification methods will give significantly worse result than simpler), it should be noted that a single classification method (and a single set of features) should be selected prior to performing an evaluation on an independent validation dataset if the test are to be statistical valid. The classification methods with implementation specific details are:

- *NMSC* (nearest mean scaled classifier): The parametric classifier under the assumption of normality and independent features with equal variances, see equations (6.9) and (6.10). The expectations are estimated as the corresponding means and the common variance as the weighted average of the individual standard variance estimates weighted by the *a priori* probabilities. When the *a priori* probabilities are estimated using the corresponding proportions in the learning dataset, this weighted variance estimate is identical to the standard pooled variance estimate.

- *LDC* (linear discriminant classifier): The parametric classifier under the assumption of normality and common covariance matrix, see equations (6.13) and (6.14). The expectations are estimated as the corresponding means and the common covariance matrix as the weighted average of the

individual standard covariance matrix estimates weighted by the *a priori* probabilities. When the *a priori* probabilities are estimated using the corresponding proportions in the learning dataset, this weighted covariance matrix estimate is identical to the standard pooled covariance matrix estimate.

- *QDC* (quadratic discriminant classifier): The parametric classifier under the assumption of normality, see equation (6.7). The expectations are estimated as the corresponding means and the covariance matrices as the corresponding standard covariance matrix estimates.

- *ParzenC*: The Parzen window classifier, see equation (6.29). The window width is estimated as the value attaining the minimum (expected) PMC when estimated using the leave-one-out cross-validation method on the learning dataset. The window function is a multivariate normal distribution with diagonal covariance matrix where all diagonal elements are equal to the squared window width. Thus the features are indirectly assumed to be independent and have the same typical range.

- *kNNC*: The k-nearest neighbour classifier, see decision rule 4. k is estimated as the value attaining the minimum (expected) PMC when estimated using the leave-one-out cross-validation method on the learning dataset. The Euclidean norm is used to measure the distances, thus the features should in particular have the same typical range.

- *NNC*: The nearest neighbours classifier, see decision rule 5. The Euclidean norm is used to measure the distances, thus the features should in particular have the same typical range.

A couple of general details about the classification methods should also be noted:

- In all classification methods which use the *a priori* probabilities, i.e. the parametric classification methods and the Parzen window method, the *a priori* probabilities are estimated as the corresponding proportions in the learning dataset when nothing else is stated. However, it will follow from the subsequent discussion of the evaluation method that the number of patients within each class in the learning datasets will typically be equal for our evaluations, thus this choice will typically correspond to choosing equal *a priori* probabilities.

- The nonparametric classification methods and the classification method which assumes normality and independent features with equal variances (case 1), all assumes that the features have the same typical range. To ensure this, we will linearly scale all the feature values prior to designing these classifiers to make the standard variance estimate of all sets of feature values in the learning dataset equal. The matter in which this is done is unimportant for the classification result, but we will in this study scale them to a fixed scalar in order to obtain equal variances for all bootstraps and for all feature sets. This will in particular allow us to compare the chosen window width in the Parzen window classifier for all bootstraps for different feature sets. Technically, we will perform the scaling, both on

the learning patterns prior to designing the classifier and on the validation patterns prior to evaluating the performance, by multiplying each feature value with the inverse of the standard estimate of its standard deviation computed using the values of the same feature for all current learning patterns. This will make the standard variance estimate of all sets of feature values in the learning dataset equal to one.

We can straight away note that the two classification methods which assume either common variance or common covariance matrix, will always give the same classification results when using only a single feature. This is because each feature distribution will only have a single variance in these cases and both methods will assume that the two variances, one for each of the two classes, are equal. We could also comment that we expect the simpler, parametric classification methods to perform best for even a small number of features because of the challenges with our dataset, see section 2.3, and especially because of the possibility of some incorrectly recorded patient outcomes, see section 2.3.3.

For evaluation we would have liked to use the holdout-validation method. However, true statistical estimates are only obtained when using an independent validation dataset, and this independence is violated if any part of our method is influence by previous studies which includes some of the same validation patterns. We have already seen that our dataset has been used in previous studies, and it is questionable to claim that the development of our methods are not influenced by these studies. Therefore, not even the holdout-validation method would provide us with true statistical estimates when using this dataset.

As it is tempting to be able to provide a data-dependent estimate of the uncertainty in our estimate, and because we really wish to say something about our method (thus expected quantities) and not only our specific classifier (the conditional quantities), we will use a bootstrap method without replacement with $B = 500$ to estimate the performance of our classifiers. Such a use of the iterative evaluation methods (cross-validation and bootstrapping) in the learning phase has also been recommended in literature, but we need to be aware of that this increases the risk of overfitting [60, p.69]. However, because our dataset can as mentioned be viewed as an extracted learning dataset from a larger dataset, the only risk associated with extensively using our dataset in this learning phase is that we may fool ourself into believing that the wrong method is the best, but we will get a result that shows this (if it is the case) when we evaluate on the previously unused holdout-dataset. We also emphasise that any method that depends on the learning dataset, e.g. the adaptively chosen weight array in the adaptive texture feature or the adaptive learning of the classifiers parameters, will be applied for each bootstrap to not increase the dependency between each learning and validation dataset.

The estimates of the uncertainties of the estimators could be given as 95 % two-sided confidence intervals (CIs). Such estimates would attempt to give the uncertainties of the expected quantities, which is the quantities of main interest here. However, these estimates are highly dependent on the number of bootstraps and will in particular converge in probability to zero as the number of bootstraps increases. To overcome this, and also not fool ourself into believing that our estimates are highly accurate when they are not, we will instead provide 95 % two-sided prediction intervals (PIs) of our estimators, which will be estimates of the uncertainties of the conditional quantities. We will not use an

estimated variance to produce these PIs because this unnecessarily assumes that our estimators are normally distributed. Instead, we will use the corresponding percentiles, the 2.5 % and 97.5 % percentile, of the empirical distribution of the estimates. However, we expect that these PIs are very similar to those who assume a normal distribution because the general rule of thumb for the applicability of the normality approximation is over 40 and we use 200 bootstraps [11, p.386].

Our last choice is how to partition the dataset. This is a delicate problem that was discussed in section 6.6.1. We will apply a simplification of these results in the following. In particular, we will use the criterion function in equation (6.58), but insert the assumption of normality with arbitrary covariance matrices for all classifiers, which correspond to the bottom row of the figures 6.8 and 6.9. This allows us both to do the same split for all classifiers and to make a somewhat reasonable partitioning for our nonparametric classifiers. To obtain an exact expression, the correct current values for the number of features and the number of patterns are used, in addition to a true expected PMC of 0.1 when the tetraploid and polyploid patterns are excluded and 0.2 if not (which are our optimistic goals for the expected PMCs), and an estimate of the Mahalanobis distance between the classes is based on the entire current dataset (and its feature distribution). Using the *entire* current dataset to estimate the Mahalanobis distance between the classes can be justified as it does not create any addition dependencies between or within different learning and validation dataset. When the Mahalanobis distance between the classes is estimated for adaptive features, then the adaptive feature values are computed using a weight array which is based on the entire current dataset, thus the estimated Mahalanobis distance between the classes used to compute the 'optimal' partitioning ratio is in general optimistically biased for the adaptive features.

We will not use the resulting partitioning directly. Firstly, because the number of patterns within each true class is uneven in our dataset and some classification methods, like the one assuming normality with arbitrary covariance matrices (case 3), degenerates when being designed on uneven datasets, we will construct each training dataset in a way that ensures that it contains the same number of patterns within each true class. Thus we will use an evened bootstrap method, where the evening will here be ensured by sampling the requested number of patterns for each class from the class specific datasets (rather than sampling the request number of patterns from the entire dataset). Secondly, we will impose a minimum accuracy in the design of the classifier and in its performance estimate by restricting the minimum number of patterns in each class in either dataset to 30 % of the total number of patterns in the same class. Unfortunately, because most feature sets will on our dataset have a low estimated Mahalanobis distance between the classes relative to the number of features, the estimated 'optimal' partitioning ratio will typically be so high that the required minimum accuracy in the performance estimate of the classifier will be reached. While it therefore is somewhat redundant to use an optimal partitioning ratio, it nevertheless justifies why we will typically only use 30 % of the patterns in the least probable class as validation patterns.

6.7.1 Reporting the classification result

Reporting the classification result can be done by using many different quantities or sets of quantities. When limiting our attention to reporting the result of a specific classifier with a predefined learning and validation dataset, the *confusion matrix* provides the most comprehensive of the classification result. In such matrices, one of the axes corresponds to the true class, while the other corresponds to the estimated class by the specific classifier, and each element gives the number of patterns with a specific true class that are classified as a specific estimated class.

In our case of two classes, the confusion matrix is only a 2x2-matrix and its strength of revealing how the classifier confuses classes is trivial; every misclassified pattern must be classified to the other class. Reporting the confusion matrix is thus equivalent to reporting the number of correctly classified and misclassified patterns within each of the two classes. Furthermore, with the knowledge of the number of patterns within each true class, we see that there are only two unique values in the confusion matrix. Many pairs of quantities could be used to describe these two unique values, and we will now describe two such pairs.

In correspondence with standard terminology, we let the relapse of ovarian cancer, i.e. bad prognosis, be referred to as a *positive* result of a specific patient. If a patient has cancer and the classification result is positive, then the patient is a *true positive (TP)*, of course, with respect to the specific classifier. If however the classification result is negative, but the patient has cancer, then the patient is a *false negative (FN)*. Similarly, if a patient does not have cancer and the classifier estimates this, the patient is a *true negative (TN)*, but if the classifier fails to identify this and its result is positive, then the patient is a *false positive (FP)* [45, p.123].

Using this terminology, we define the *sensitivity* of the classifier as TP/(TP+FN). This quantity indicates the ability of the classifier to correctly classify the bad prognosis patients. Similarly, *specificity* of the classifier is defined as TN/(TN+FP). This quantity indicates the ability of the classifier to correctly classify the good prognosis patients. While these performance quantities are indeed interesting, they are not individually interesting for arbitrary classification methods as it is easy to maximise either of them, e.g. by always deciding the corresponding class [45, p.123].

Much too often, the classification performance is given in a single value called the *correct classification rate (CCR)*. As the name indicates, this is the proportion of the patterns, or here patients, that is correctly classified and can for instance be computed as the ratio of the sum of the diagonal elements in the confusion matrix to the sum of all elements in the confusion matrix. For the case of two classes, this quantity can also be computed as a weighted average of the sensitivity and the specificity where the weights are the proportions of patterns in the true positive and the true negative class, respectively. Thus, this quantity emphasise the correct classification of the most frequently occurring class. This can be appropriate in some situations, but for our case we are equally interested in correctly classifying the bad prognosis patients as the good prognosis patients, thus this measurement is not appropriate because the bad prognosis is in clear minority in our dataset and this unevenness is enforced by the use of an *evened* bootstrap method. Facilitating this, a natural quantity

Table 6.2: A generic, complete classification result.

	Prognosis	Patients	Correctly classified	Misclassified	CCR
	Good	n_1	TN	FP	specificity
	Bad	n_2	TP	FN	sensitivity
Total:		n_L	TN+TP	FP+FN	CCR

CCR when assuming equal a priori *probabilities: (specificity+sensitivity)/2*

is to use the true average, instead of the weighted average, of the sensitivity and the specificity. It is easy to realise that this corresponds to the CCR when assuming equal *a priori* probabilities, and we will refer to this quantity as the *CCReq*. Since the CCReq and the CCR are two independent measurements of the confusion matrix, it is also easy to realise that this pair of quantities defines the confusion matrix when assuming knowledge of the number of patterns within each true class.

Table 6.2 shows the complete report of a generic classification result. The table also shows the relationship between the confusion matrix and the two discussed pairs of quantities that define this matrix under the assumption of known number of patterns within each true class. Because we are planning to use six different classification methods, making a complete report as in table 6.2 for each of the classification methods for each evaluated feature combination is somewhat extravagant. We will instead only use a more compact representation of the same information when we wish to present the complete classification result. This representation will include the CCReq, CCR, specificity and sensitivity of each classification method, and are thus both informative and defines all six confusion matrices.

Since we are planning to use a bootstrap validation method, we need to generalise the reporting of the classification results to the case of multiple bootstraps and not just a predefined learning and validation dataset. To obtain a point estimate of each expected quantity, we will use the average of the same quantity estimated for all bootstraps. This approach is similar to the expected PMC estimate obtained by averaging the PMC estimates of each bootstrap, which stands in contrast to the leave-one-out bootstrap estimate which also could be used for general quantities (and not just the PMC).

To obtain an estimate of the uncertainty of each quantity estimator, we will construct a 95 % two-sided PI of each of them. This will be done by using the corresponding percentiles, the 2.5 % and 97.5 % percentile, of the empirical distribution of each quantity estimate. Because we will use $B = 500$, the empirical distributions will contain five hundred values, which makes the 2.5 % and 97.5 % percentile the 13th smallest and largest quantity estimate, respectively. Note that a percentile of the empirical distribution of a quantity estimate is generally not equal to the same quantity estimate computed from the same percentile of the elements in the confusion matrix, thus we will of course obtain the empirical distribution for each desired quantity to compute the correct uncertainty estimate.

Table 6.3 shows the resulting generic classification result when using a spe-

Table 6.3: A generic classification result when using a specific classification method. The square brackets gives the estimated 95 % two-sided PI of the performance quantity in the right column. The value outside these brackets is the corresponding estimated expected performance.

CCReq	? % [? %, ? %]
CCR	? % [? %, ? %]
Specificity	? % [? %, ? %]
Sensitivity	? % [? %, ? %]

Using n_1 and n_2 learning patterns with good and bad prognosis, respectively.

cific classification method. It should be noted that the estimated expectation and PI of different classification results are not directly compared to determine significant difference as this would have given unrealistic results because the PI is not the estimated uncertainty of the estimated expectation. Instead we will apply a rough manual determination of significant difference; if the estimated expectations differs with less than 1 %, then the results are not significantly different, but they are significant different if the estimated expectations differs with more than 1.5 %. A difference in the interval $[1.0\%, 1.5\%]$ can be said to be questionably significant in general. This assertion of significance is inspired by the length of the estimated 95 % two-sided CI of the expected CCReq and CCR, which is about 1 % for good classifiers when using the mentioned evaluation method on our dataset. We note that a hypothesis test is not applied to determine significant difference as this would have been highly dependent on the number of bootstraps.

Chapter 7

Results and discussion

We will in this chapter present the classification results of the most promising features with respect to our dataset, in addition to the classification results of the proposed property arrays, the CSDEMs and the CSDEM sum histograms. The discussion of the results is interleaved to ease the reading, but it is still attempted to be separated from the factual results.

Throughout this chapter we will consider both the entire dataset of 134 patients and a subset containing only the 102 patients with either diploid or aneuploid histogram (using the ploidy classification described in section 3.1.3). The reason for selecting the subset is that it has been shown, and will also become evident in this chapter, that many properties of the patients with tetraploid or polyploid histogram are typically opposite to that of the patients with diploid or aneuploid histogram. From the description of the ploidy types in section 3.1, we see that IOD, which can be seen as a combination of the average area and grey level, is an example of such a property. This is because the cells that contribute to a diploid histogram will typically have a lower IOD than the cells that contribute to an aneuploid or tetraploid histogram, and the cells that contribute to a polyploid histogram typically have the highest IOD, but the true prognosis of a patient with diploid, tetraploid or polyploid histogram is typically good in our dataset, while bad for a patient with aneuploid histogram (see section 3.1.3).

We must note that one should be careful when selecting a subset to avoid drawing false conclusions. Firstly, the selection criteria can not make use of the true class of the patterns. Secondly, when only evaluating one of the subsets (and not also the complimentary subset), the conclusion is in general not valid for the entire dataset. A consequence of this is that any selection criterion must in this case have a concrete and reasonable interpretation. To obtain a conclusion which is valid with respect to the entire dataset, one should also perform the evaluation on the complimentary subset. For our dataset, there is unfortunately no point in performing such an evaluation as the number of patients with tetraploid or polyploid histogram and bad prognosis is only four. The conclusion of our evaluations on the subset is thus only valid for patients with diploid or aneuploid histograms, but we may postulate that the performance on the complimentary subset would have been similar if we had enough patients in this subset.

We will begin this chapter with a discussion of the segmentation methods by evaluating the cell features in combination with the NO-features for different segmentation methods. We will continue with considering the currently

Table 7.1: An overview of the discussions in this chapter and their locations.

Section	Discussion
7.1	Segmentation methods (using cell features + NO features)
7.2.1	GLEM-features Mahalanobis assumptions for the GLEM-features
7.2.2	GLEM4D-features
7.2.3	GLEM4D-features versus cell features + NO features
7.3	CSDEM-features
7.3.1	Mahalanobis assumptions for the CSDEM-features
7.3.2	CSDEM-features versus cell features + the NO-features CSDEM-features versus GLEM4D-features
7.4	CSDEMsum-features CSDEMsum-features versus CSDEM-features
7.4.1	Mahalanobis assumptions for the CSDEMsum-features
7.4.2	CSDEMsum-features versus cell features + NO-features
7.5.1	GLEM4D-features + cell features + NO-features
7.5.2	CSDEMsum-features + GLEM4D-features + cell features + NO-features
7.6	Classifier complexity and the choice of classification method
7.7.1	Choice of partitioning limit
7.7.2	Choice of the number of quantification levels per integer entropy
7.7.3	Using a stratified bootstrap method
7.7.4	Using the two different estimates of the common variance
7.8	Image analysis vs DNA ploidy analysis

most promising choice of property array for our dataset, the GLEM- and the GLEM4D-features. This will be followed by the evaluation and discussion of the proposed CSDEM- and CSDEMsum-features. The analysis of features will be concluded by an attempt of combining the best adaptive texture features with some of the cell features or NO-features in hope of further increasing the classification performance.

We will then discuss some related issues. First out is a discussion of the classifier complexity and the choice of classification method. This will be followed by a section where we look into what would have happened if we had made some other design choices. We will here specifically consider the choice of required minimum accuracy in the performance estimates of the classifier, the number of quantification levels per integer entropy in features based on spatial entropy, the use of stratified bootstrap instead of evened bootstrap and finally the effect of using the two different estimates of the common variance when estimating the Mahalanobis distance between the classes at each element in the design of the weight arrays. The chapter will be concluded by a general discussion of our findings. Table 7.1 shows an overview of the discussions and their location.

7.1 Segmentation methods

We will in this section attempt to determine which of the mentioned segmentation methods that is most appropriate for our classification purpose. Because the main difference in the mentioned segmentation methods is how they handle overlapping structures, the number of objects should be representative for discovering significant differences between them. However, we see from the classification results in table 7.2 that the cell features are highly significant features for our dataset. If we attempt to determine the best segmentation method based on evaluations of only the NO-features for different segmentation methods, we risk to consider the correlation with the cell features more than the information provided by the number of objects in itself. We will therefore compare the segmentation methods by combining the cell features with the NO-features for different segmentation methods.

We note that significant differences in the classification results of this section are expected to be in particular representative for the CSDEM- and CSDEMsum-features. This is not only because the NO-features are likely to be representative for the differences between segmentation methods, but also because of the likely positive correlation between the expected spatial entropy of the object size and the number of objects, a relation which was mentioned in section 4.4. Furthermore, if multiple segmentations of the same cell image have different number of objects, then this will also affect the object size, which in turn also affects the CSDEM- and CSDEMsum-features. We can therefore conclude that significant difference in the classification results of the NO-features are likely to be in particular representative for the CSDEM- and CSDEMsum-features.

We proposed two segmentation methods in section 4.3. Both where based on Niblack's method and the validation step of Yanowitz and Bruckstein's segmentation method [72, p.86] to obtain an initial segmentation. We thereafter suggested to remove all estimated bright primitives sufficiently close to the edge of the nucleus because we expect multiple falsely estimated bright primitives in this region. Because this procedure will also remove any estimate of true bright primitives on this region, we will here evaluate the segmentation methods resulting from both including and excluding this step. We finally proposed two different algorithms which both attempts to separate overlapping primitives and also removed small objects, one based on morphology and another based on the watershed transform. We will evaluate both these algorithms here, both when including and excluding the edge removal step prior to their application.

We will compare our four segmentation methods with the segmentation method used in [49] (see section 3.2.6). The classification results of using the combination of the cell features and the NO-features with this method is shown in table 7.3. In comparison with the classification results when using only the cell features in table 7.2, we see that both expected CCReqs have increased with a good percent, indicating that the NO-features are slightly prognostic relevant when using this segmentation method.

Table 7.4 shows the classification results of using the combination of the cell features and the NO-features with each of our proposed segmentation methods when evaluating on all 134 patients. These results indicates that the NO-features are slightly prognostic relevant also when using our segmentation methods. They do however not provide enough differences to determine which separation algorithm is best, nor do they indicate whether bright primitives near

Table 7.2: The classification results of the cell features when using the classification method which attained the best expected CCReq; LDC.

	All 134 patients	The 102 patients
CCReq	68.5 % [55.3 %, 79.9 %]	76.7 % [63.9 %, 88.1 %]
CCR	68.3 % [59.0 %, 76.9 %]	77.8 % [67.3 %, 88.5 %]
Specificity	68.2 % [56.1 %, 78.8 %]	78.5 % [63.4 %, 90.2 %]
Sensitivity	68.8 % [41.7 %, 91.7 %]	74.9 % [45.5 %, 100.0 %]

Using 28 (left) and 25 (right) learning patterns in each prognosis class.

Table 7.3: The classification results of the cell features and the NO-features with the segmentation method used in [49] when using the classification method which attained the best expected CCReq; NMSC.

	All 134 patients	The 102 patients
CCReq	70.0 % [59.1 %, 80.7 %]	77.8 % [66.0 %, 88.5 %]
CCR	71.2 % [61.5 %, 79.5 %]	80.8 % [69.2 %, 90.4 %]
Specificity	71.7 % [59.1 %, 81.8 %]	83.0 % [65.9 %, 95.1 %]
Sensitivity	68.4 % [41.7 %, 91.7 %]	72.6 % [45.5 %, 90.9 %]

Using 28 (left) and 25 (right) learning patterns in each prognosis class.

the edge of the nucleus should be removed or not. They are also not sufficiently different from the results in table 7.3 to determine the relation between our segmentation methods and the segmentation method used in [49].

Table 7.5 shows the corresponding classification results when evaluating on the 102 patients. We see in comparison with the results in table 7.3 that the best of our segmentation methods are now significantly better than the segmentation method used in [49]. However, also these results do to provide sufficiently difference to determine which of our segmentation methods is best.

We would like to note that the equally good classification results of our four segmentation methods, and also the segmentation method used in [49] when evaluating on all 134 patients, does not mean that similar classification results would be obtained by any slightly meaningful segmentation method. Indeed, if excluding the separation algorithm from our segmentation methods, the results of the same feature combination is significantly worse than the results in the tables 7.4 and 7.5, and they are furthermore not significantly different from the results with only the cell features (the best expected CCReq was 68.8 % when using all 134 patients and including the edge removal step and 77.6 % when using the 102 patients and excluding the edge removal step). Also, if we apply the morphological separation algorithm, but exclude the last step which performs an opening with the filled, flat 2x2-structure element, the results will be slightly worse when using all 134 patients (best expected CCReq was 69.8 %, obtained when including the edge removal step) and significantly worse when using the

Table 7.4: The classification results of the cell features and the NO-features with our segmentation methods when evaluating on all 134 patients and using the classification method which attained the best expected CCReq; NMSC. Edge removal *is used as the shorthand for the step which removes all estimated bright primitives sufficiently close to the edge of the nucleus.* Morphology *and* watershed *are used as the shorthands for the separation algorithms which are based on morphology and the watershed transform, respectively.*

	No edge removal Morphology	With edge removal Morphology
CCReq	70.2 % [57.2 %, 82.2 %]	70.1 % [57.2 %, 82.2 %]
CCR	69.9 % [59.0 %, 78.2 %]	70.2 % [59.0 %, 79.5 %]
Specificity	69.8 % [56.1 %, 80.3 %]	70.3 % [56.1 %, 81.8 %]
Sensitivity	70.7 % [41.7 %, 91.7 %]	69.9 % [41.7 %, 91.7 %]
	No edge removal Watershed	With edge removal Watershed
CCReq	70.4 % [58.0 %, 82.2 %]	70.0 % [56.8 %, 81.8 %]
CCR	70.4 % [59.0 %, 78.2 %]	69.9 % [57.7 %, 78.2 %]
Specificity	70.3 % [56.1 %, 80.3 %]	69.8 % [54.5 %, 80.3 %]
Sensitivity	70.4 % [41.7 %, 91.7 %]	70.3 % [41.7 %, 91.7 %]

Using 28 learning patterns in each prognosis class.

Table 7.5: The classification results of the cell features and the NO-features with our segmentation methods when evaluating on the 102 patients and using the classification method which attained the best expected CCReq; NMSC.

	No edge removal Morphology	With edge removal Morphology
CCReq	79.4 % [66.6 %, 90.6 %]	79.8 % [66.6 %, 90.6 %]
CCR	81.4 % [71.2 %, 90.4 %]	81.9 % [71.2 %, 90.4 %]
Specificity	82.9 % [68.3 %, 95.1 %]	83.4 % [68.3 %, 95.1 %]
Sensitivity	75.8 % [45.5 %, 100.0 %]	76.1 % [45.5 %, 100.0 %]
	No edge removal Watershed	With edge removal Watershed
CCReq	79.3 % [67.8 %, 91.5 %]	79.0 % [67.5 %, 89.4 %]
CCR	81.4 % [69.2 %, 90.4 %]	81.4 % [71.2 %, 90.4 %]
Specificity	82.9 % [63.4 %, 92.7 %]	83.1 % [65.9 %, 95.1 %]
Sensitivity	75.7 % [45.5 %, 100.0 %]	74.9 % [45.5 %, 100.0 %]

Using 25 learning patterns in each prognosis class.

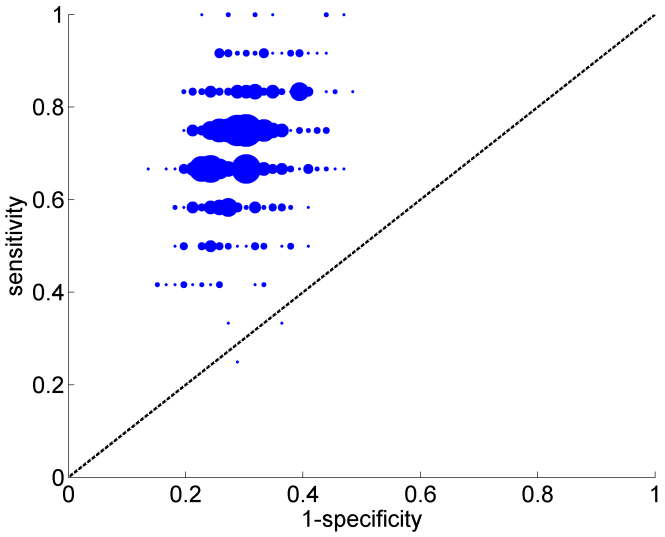

Figure 7.1: The ROC point cloud of the cell features and the NO-features when evaluating on all 134 patients and using the NMSC classification method. The NO-features are computed using our watershed segmentation method without the step which removes bright edge objects.

102 patients (best expected CCReq was 77.8 %, obtained when excluding the edge removal step). This is interesting because this last segmentation method can indeed be said to be reasonable as the alternation only removes the cleaning step of the morphological algorithm, a step which can be said to be coarse in combination with a separation technique that applies opening with a *linear* structure element. We therefore believe that our segmentation methods result in good classification results because they are reasonable and not by mere chance, though the results obviously also indicate that there are multiple paths to a good segmentation of our cell images.

Figures 7.1 and 7.2 show the *receiver operating characteristic (ROC)* point cloud of the classifiers which obtained the best expected CCReq with the combination of the cell feature and the NO-features for all 134 patients and the 102 patients, respectively. The ROC point clouds visualises the pair of specificity and sensitivity for each of the 500 bootstraps, thus giving an accurate visual impression of the uncertainty in the classification results. The diagonal line represents random guessing, which is the line where the average of the specificity and sensitivity is 0.5, and which also corresponds to a CCReq of 50 %. We must be alerted if multiple bootstraps cross this line because this would indicate that the true performance of the classifier may be random.

The two ROC point clouds indicates a large uncertainty in the classification performance. They also show that the uncertainty is larger for the classifier based on all 134 patients than for the classifier based on the 102 patients. Both these observations are also indicated by the PI of the corresponding classification results in tables 7.4 and 7.5. Despite the large uncertainty, the ROC point clouds

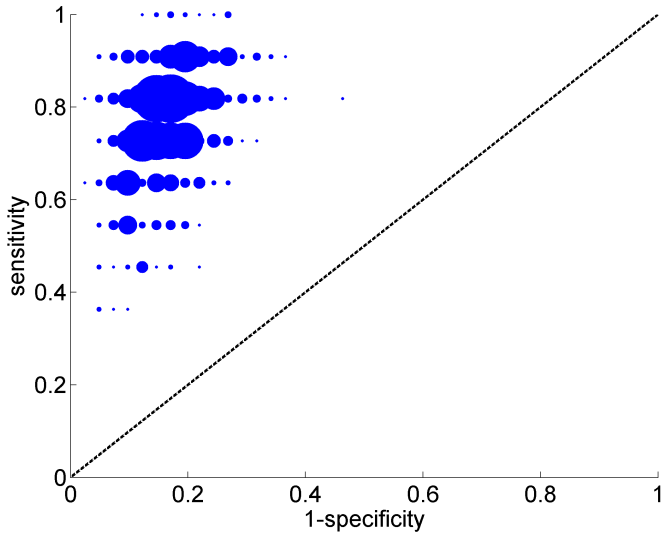

Figure 7.2: The ROC point cloud of the cell features and the NO-features when evaluating on the 102 patients and using the NMSC classification method. The NO-features are computed using our morphological segmentation method with the step which removes bright edge objects.

also indicate that both classifiers are meaningful and in particular significantly better than random guessing.

7.2 Grey level entropy matrices

The currently most promising choice of property array is based on the grey level entropy matrices. Each of these matrices captures much prognostic relevant information in a single feature, making them robust against the overfitting problem and therefore likely to generalise well. We will begin with discussing the GLEM-features, followed by the GLEM4D-features, and we conclude with a short comparison between the best feature based on grey level entropy matrices and the combination of cell features and the best NO-features.

7.2.1 The GLEM-features

The best classification results of the GLEM-features are shown in table 7.6. With respect to all 134 patients and the 102 patients and both with respect to the expected CCReq and the expected CCR, the best GLEM-feature among our four types of adaptive texture features is the negative GLEM-feature (closely followed by the difference GLEM-feature). This is interesting in itself, as it indicates that the positive part of the weight arrays[1] do not provide the GLEM-feature with new information. On the contrary, it confuses the feature because

[1]There are for each bootstrap three weight arrays in a GLEM-feature because the three used cell area groups are treated separately.

Table 7.6: The classification results of the negative GLEM-feature when using the classification method which attained the best expected CCReq; NMSC/LDC.

	All 134 patients	The 102 patients
CCReq	63.3 % [51.1 %, 75.4 %]	71.0 % [57.5 %, 83.6 %]
CCR	69.2 % [59.0 %, 76.9 %]	78.3 % [69.2 %, 86.5 %]
Specificity	71.9 % [57.6 %, 83.3 %]	83.7 % [70.7 %, 92.7 %]
Sensitivity	54.7 % [25.0 %, 83.3 %]	58.3 % [27.3 %, 81.8 %]

Using 28 (left) and 25 (right) learning patterns in each prognosis class.

it degrades its performance. There are two possible main reasons for this. The first is that the region which corresponds to the positive parts contains no information. The second is that the information in the positive parts is the same as in the negative part, but more prominent in the negative part. Because the lower limits of CCReq's PI of the positive GLEM-feature (for the best classification method) are 49.6 % and 53.5 % for all 134 patients and the 102 patients, respectively, we expect that the positive parts have just a little prognostic value information, but that this information is also present in the negative part.

In the discussion in section 3.2.4, we noted that it is essential to inspect and interpret the designed weight arrays in order to get a better understanding of what an adaptive texture features measure. Since we are using the bootstrap method for evaluation, we have multiple learning datasets and there are multiple weight arrays to interpret. We could overcome this problem by inspecting some of the weight arrays and plot a few which are representative, or plot the average of all weight arrays or plot the weight array designed using the entire dataset. We will apply the latter, but it must be noted that because such weight arrays have more scenes, its estimates will be more reliable than the weight arrays that will be used in the evaluation. On the other hand, these weight arrays will give a better understanding of where the discrimination value of the property arrays is high.

Figure 7.3 shows the designed weight arrays of the three area groups for the difference GLEM-feature when using the 102 patients. The grey surroundings are the elements where the weight arrays are zero (typically because of no occurrences), the darker lower region where the weight arrays are negative and the brighter upper region where they are positive. It is from these figures clear that the GLEM-features mainly measure the average grey level; lower grey level increases the probability of being bad prognosis. However, we also see that large grey level entropy indicates bad prognosis, even for high grey levels. Because the intensity changes in our cell images are gradual, large grey level entropy is correlated with large grey level variance. This observation is also verified by replacing the grey level entropy axis with the grey level variance in the same local window (9x9), resulting in the corresponding *grey level variance matrix (GLVM)* [73], which give insignificant different classification results with respect to both the CCReq and the CCR. Therefore, the GLEM-features can be seen as combined measurements of the average and variance in grey level.

The connection between the negative GLEM-feature and the average and

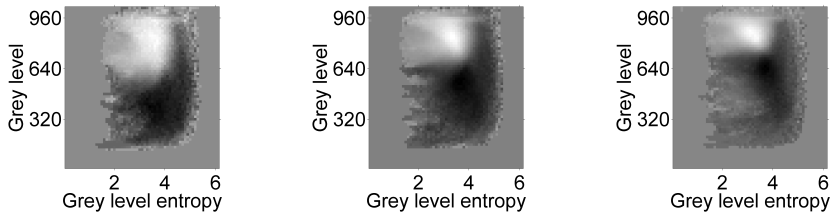

Figure 7.3: The designed weight arrays of the difference GLEM-feature when using the 102 patients. The arrays corresponds to the cell area group $[2000, 2999]$, $[3000, 3999]$ *and* $[4000, 4999]$ *from left to right. Each image is linearly scaled to fill the entire grey level range; the true range is from left to right:* $[-1.0, 0.85]$, $[-1.1, 1.1]$ *and* $[-1.4, 1.2]$.

variance in grey level are visualised by the scatter plots in figure 7.4 when using the 102 patients. For these scatter plots and all following scatter plots containing adaptive texture feature(s), the values of the adaptive texture feature(s) are computed using weight array(s) which are designed using the entire dataset that is visualised (the 102 patients in figure 7.4). This makes the visualised separation of the adaptive texture feature optimistically biased, but typically only slightly because the weight arrays will typically be well filled with occurrences due to our great concern for the overfitting problem. We emphasise that such computation of the values of adaptive texture features are *only* done to make plots and never during evaluation.

The corresponding weight arrays as in figure 7.3 when using all 134 patients shows the exact same pattern as this figure, but each element have about 20 % less estimated discrimination value and the negative part is nearly uniform (instead of peaked). These changes are as expected because patients with tetraploid or polyploid histograms are typically good prognosis in our dataset, but typically have the same local grey level characteristics as the patients with

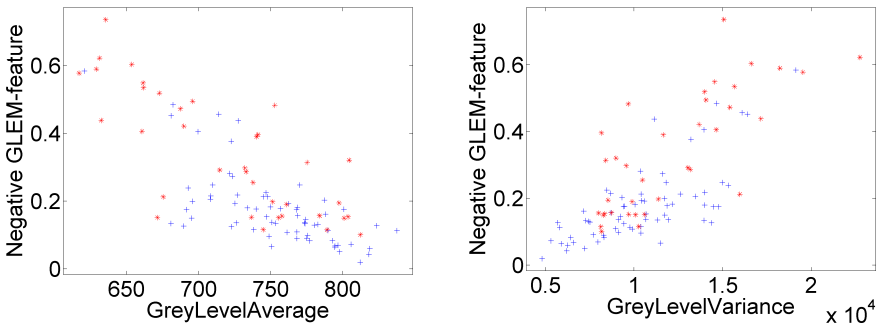

Figure 7.4: Scatter plot of the negative GLEM-feature against: left) the GreyLevelAverage-feature, right) the GreyLevelVariance-feature when using the 102 patients. The blue plus sign represents good prognosis and the red asterisk symbol represents bad prognosis.

aneuploid histograms. This is because all three ploidy histograms indicate that a significant proportion of the cell images have large IODs and thus also low grey level, see section 3.1). However, as the pattern of the weight arrays are similar, the discussion of what the GLEM-features measures is valid also for all 134 patients. Also, because the grey level average and variance are obviously affected similar by the inclusion of the patients with the tetraploid and polyploid histograms, the connection between the GLEM-feature and the average and variance in grey level is also still valid.

Assumptions of the estimated Mahalanobis distance between the classes

Because the weight arrays are designed using the estimated Mahalanobis distance between the classes, it is interesting to investigate whether the underlying assumptions are met. To test these assumptions, we will assume that the samples within each element in the collection of the property arrays of all 134 patients can be seen as independent. We will then test the normality assumption of each prognosis class using the Lilliefors goodness-of-fit test [32] at significance level 0.05. This is a generalisation of the Kolmogorov-Smirnov test for the case of normality when the expectation and variance are unknown [32, p.399]. The assumption of equal variances will be tested using the standard F-test [11, pp.515–519] at significance level 0.05 (the null hypothesis will of course be that the two variances are equal). Note that this test is strongly dependent on the normality assumption [11, p.519]. In particular, the standard F-test is more dependent on the normality assumption than the pooled two-sample t-test which the estimated Mahalanobis distance between two classes can be seen as the T-statistic of, if letting the null hypothesis be equal expectations [11, p.519]. However, because none of the tests would ideally be rejected (as the appropriateness of using the estimated Mahalanobis distance between the classes can only be guarantied in this case), we expect that the standard F-test performs acceptably as the distributions are at least approximately normal when none of the normality tests are rejected.

Figure 7.5 shows the result of testing the assumptions. We see from the images in the left and middle column that the normality assumptions are slightly questionable. In comparison with figure 7.3 we note that the assumptions are not rejected in the most discriminative elements. This is comforting, but only a natural consequence of the central limit theorem as these are also the elements with most occurrences[2]. The common variance assumption is slightly more frequently satisfied and also this assumption seems most appropriate in the more interesting elements. In total, we conclude that the underlying assumptions of the estimated Mahalanobis distance between the classes seem to be generally acceptable when using the GLEM-features.

7.2.2 The GLEM4D-features

The best classification results of the GLEM4D-features are shown in table 7.7. Again, the best among our four types of adaptive texture features is the negative adaptive texture feature (closely followed by the difference adaptive texture

[2]As mentioned in section 3.2.5, a direct application of the central limit theorem is illegal because of the dependencies between cells of the same patient, but we still expect a similar effect when greatly increasing the number of occurrences.

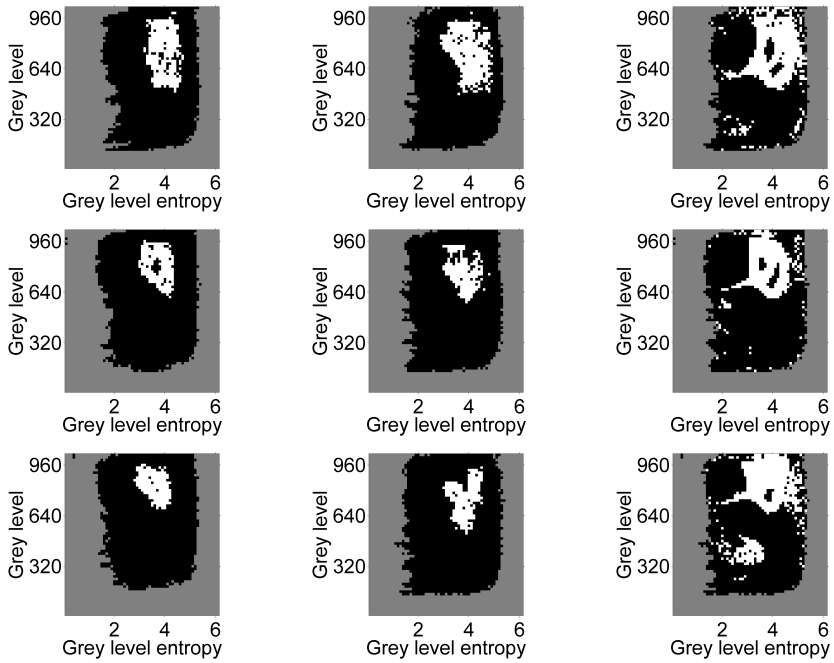

Figure 7.5: The assumption of: left column) normality in good prognosis, middle column) normality in bad prognosis, right column) equal variances of the difference GLEM-feature when using the 102 patients and the cell area group: upper row) $[2000, 2999]$*, middle row)* $[3000, 3999]$*, lower row)* $[4000, 4999]$*. The corresponding tests are rejected in black pixels and not rejected in white pixels, both at significance level 0.05. The grey pixels corresponds to elements where all relevant property arrays are zero.*

feature)[3]. In comparison with the result of the negative GLEM-features in table 7.6, we see the performance change when using all 134 patients is insignificant, but is clearly significant for the 102 patients.

It is difficult to inspect the designed weight array of the GLEM4D-features because of its four dimensions. We will therefore take a different approach to get an understanding of what the GLEM4D-features measures. While it has two common axes with the GLEM-features, the significantly improved performance tells us that the two added axes provide new or better information. A natural question is therefore whether both or only one of the axes are of prognostic relevance. It turns out that only the area group axis is relevant. Indeed, eval-

[3]With respect to the expected CCR. The expected CCReq of the difference GLEM4D-feature is 0.3 % than for the negative GLEM4D-feature when using all 134 patients, but because this is an insignificant amount and the difference in expected CCR was nearly 3 % in favour of the negative GLEM4D-feature, the negative GLEM4D-feature is considered to be the better among these two features. This conclusion can however be debated as we are most interested in the CCReq and the lower and upper limit of the PI is 1.9 % higher and lower, respectively, for the difference feature with respect to the negative feature, thus indicating that the difference GLEM4D-feature provides a more reliable measurement in terms of discriminating between the classes.

Table 7.7: The classification results of the negative GLEM4D-feature when using the classification method which attained the best expected CCReq; NMSC/LDC.

	All 134 patients	The 102 patients
CCReq	63.8 % [51.1 %, 76.5 %]	76.1 % [62.1 %, 89.4 %]
CCR	69.0 % [61.5 %, 75.6 %]	82.3 % [75.0 %, 90.4 %]
Specificity	71.3 % [60.6 %, 80.3 %]	86.8 % [75.6 %, 95.1 %]
Sensitivity	56.4 % [33.3 %, 83.3 %]	65.4 % [36.4 %, 90.9 %]

Using 28 (left) and 25 (right) learning patterns in each prognosis class.

uating the *GLEM3D*-features resulting from setting the window width in the GLEM4D-features to 9 gives expected CCReq and expected CCR which differs with less than 0.5 % (in absolute value) from corresponding performance estimates of the negative 4D-GLEM-features, both when using all 134 patients and the 102 patients (the best adaptive texture feature and classification method were again the negative adaptive texture feature and NMSC, respectively).

With respect to the classification results of the GLEM3D-features and the improved performance over the GLEM-features, it is natural that the area group axis provides new prognostic relevant information. Indeed, the scatter plot in figure 7.6 shows that this is the case. Because of this new prognostic relevant axis, we are not sure to which extent the connection between the GLEM-features and the grey level average and variance are inherited to the GLEM4D-features.

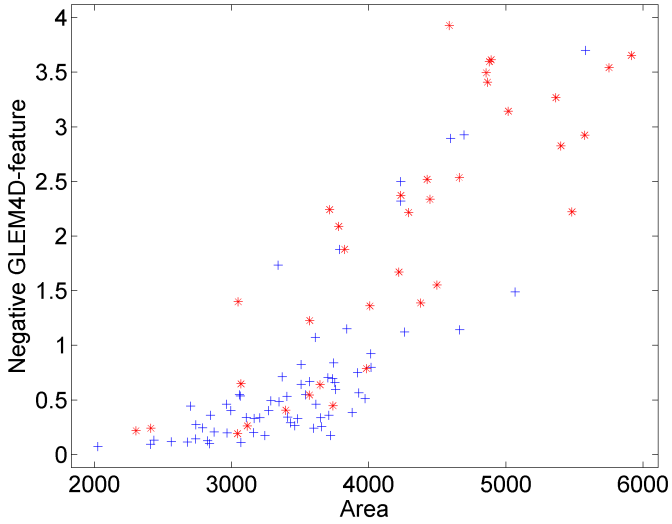

Figure 7.6: Scatter plot of the negative GLEM4D-feature against the Area-feature when using the 102 patients. The blue plus sign represents good prognosis and the red asterisk symbol represents bad prognosis.

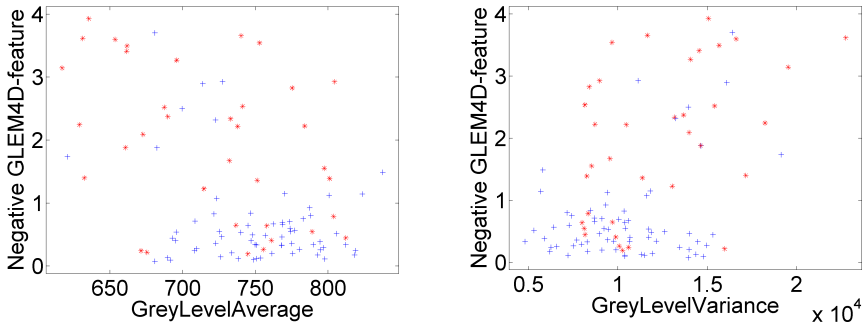

Figure 7.7: Scatter plot of the negative GLEM4D-feature against: left) the GreyLevelAverage-feature, right) the GreyLevelVariance-feature when using the 102 patients. The blue plus sign represents good prognosis and the red asterisk symbol represents bad prognosis.

We see from the scatter plots of the negative GLEM4D-feature against the grey level average and variance in figure 7.7 that this connection is greatly weakened, but still present. The presence of this connection is also indicated by both the greater separation along the negative GLEM4D-feature axis of figure 7.6 in comparison with the Area-feature axis and a comparison between the classification results of the negative GLEM3D-feature and the Area-feature (the CCReqs and CCRs of the negative GLEM3D-feature are about 2–6 % better than the corresponding performance estimates of the Area-feature for both all 134 patients and the 102 patients). We therefore conclude that the GLEM4D-features can be seen as combined measurements of the area and the grey level average and variance.

7.2.3 Comparison with the combination of the cell features and the NO-features

The performance of the best feature based on grey level entropy matrices, the negative GLEM4D-feature, is good. Indeed, in comparison with the cell features and the best NO-features, the negative GLEM4D-feature obtained equally good expected CCR with significantly better lower limit of the PI (the expected CCR and its PI of the negative GLEM4D-feature when using QDC and all 134 patients was 71.2 % [62.8 %, 78.2 %]). However, it provides significantly worse expected CCReq - over 6 % when using all 134 patients and over 3 % when using the 102 patients - and the corresponding PIs are less different or even insignificant when accommodating for the difference in expected CCReq. The net result is however still relatively good as the negative GLEM4D-feature is only a single feature, while the number of cell features and NO-features are five and two, respectively.

7.3 The CSDEM-features

In correspondence with section 7.1, we have four segmentation methods we wish to use to compute the CSDEM-features. We have evaluated the two CSDEM-features in each of these segmentation methods for each of the four adaptive texture feature types. The performance relation between the different adaptive texture features types is similar to the GLEM4D-features; the negative adaptive texture feature attains a significantly higher expected CCR than the other types with respect to both all 134 patients and the 102 patients, and also a significantly higher expected CCReq with respect to the 102 patients. In all cases, the difference adaptive texture feature is next to best, and this is also typically the best adaptive texture feature with respect to the expected CCReq and all 134 patients, but not significantly better (the maximum difference in expected CCReq is 0.8 % in favour of the difference adaptive texture feature, but the PIs corresponding to this maximum difference are equally unreliable, which stands in contrast to the observation with the GLEM4D-features and also with the CSDEM-features when using other segmentation methods).

The classification results of the negative CSDEM-features for all 134 patients and the 102 patients are shown in table 7.8 and 7.9, respectively. It is from these results clear that the classification performance of the CSDEM-features significantly decreases when including the step which removes all estimated bright primitives sufficiently close to the edge of the nucleus. While we noted in section 4.3.3 that most of the periphery of the nuclei is likely to be classified as bright primitives due to relatively small projection of each nucleus in this region (this relative difference is relevant because the local windows of the Niblack's method will include pixels further away from the edge of the nucleus), these re-

Table 7.8: The classification results of the negative CSDEM-features when evaluating on all 134 patients and using the classification method which attained the best expected CCReq; NMSC for the case with edge removal and watershed, LDC for the rest.

	No edge removal Morphology	With edge removal Morphology
CCReq	64.8 % [51.9 %, 76.5 %]	61.7 % [48.9 %, 73.5 %]
CCR	69.0 % [59.0 %, 75.6 %]	67.4 % [59.0 %, 74.4 %]
Specificity	70.9 % [59.1 %, 80.3 %]	69.9 % [59.1 %, 78.8 %]
Sensitivity	58.8 % [33.3 %, 83.3 %]	53.4 % [25.0 %, 83.3 %]
	No edge removal Watershed	With edge removal Watershed
CCReq	63.9 % [51.9 %, 75.8 %]	61.4 % [47.3 %, 73.5 %]
CCR	68.4 % [57.7 %, 75.6 %]	67.5 % [59.0 %, 74.4 %]
Specificity	70.4 % [56.1 %, 80.3 %]	70.2 % [57.6 %, 78.8 %]
Sensitivity	57.4 % [25.0 %, 83.3 %]	52.6 % [25.0 %, 75.0 %]

Using 28 learning patterns in each prognosis class.

Table 7.9: The classification results of the negative CSDEM-features when evaluating on the 102 patients and using the classification method which attained the best expected CCReq; LDC.

	No edge removal Morphology	With edge removal Morphology
CCReq	77.0 % [65.4 %, 88.1 %]	73.6 % [60.9 %, 86.0 %]
CCR	83.5 % [76.9 %, 90.4 %]	80.8 % [73.1 %, 88.5 %]
Specificity	88.3 % [80.5 %, 95.1 %]	86.0 % [75.6 %, 95.1 %]
Sensitivity	65.7 % [45.5 %, 90.9 %]	61.2 % [36.4 %, 90.9 %]
	No edge removal Watershed	With edge removal Watershed
CCReq	76.6 % [63.3 %, 89.4 %]	73.2 % [59.6 %, 85.1 %]
CCR	83.5 % [76.9 %, 90.4 %]	81.0 % [73.1 %, 88.5 %]
Specificity	88.6 % [80.5 %, 95.1 %]	86.7 % [75.6 %, 95.1 %]
Sensitivity	64.6 % [36.4 %, 90.9 %]	59.7 % [36.4 %, 90.9 %]

Using 25 learning patterns in each prognosis class.

sults strongly indicate that the applied algorithm for removing falsely estimated bright primitives in this region is too coarse. This is not entirely unexpected as the algorithm simply sets all pixels sufficiently close to the edge of the nucleus with bright class label to the grey class label, thus potentially also removing true estimated bright primitives in this region. With few indications of how to separate the true estimated bright primitives from the false ones in this region, we simply conclude that it seems better to leave all bright primitives in this region unchanged when using CSDEM-features.

The estimated performance of the morphological algorithm is slightly better than the one based on the watershed transform, but the differences in CCReq and CCR are generally too small to be called significant. However, the improved expected sensitivity (a good percent) with a corresponding greatly improved lower limit (nearly 10 %) when using the morphological algorithm without the edge removal step, indicates that the features based on this algorithm may be significantly better than the others with respect to correctly classifying the patients with bad prognosis.

Let us study the designed weight arrays to get a better understanding of the CSDEM-features. On the basis of the classification results in table 7.8 and 7.9, it seems reasonable to base this study on the CSDEM-features which use the morphological algorithm without the edge removal step. Figure 7.8 shows the designed weight arrays of the difference CSDEM-features when using both all 134 patients and the 102 patients. First of all, if we look at the axes individually, we see that both large grey level entropy[4] and large spatial entropy indicates bad prognosis while small entropies indicates good prognosis. This is the expected

[4]Note that this grey level entropy differs from the grey level entropy in the GLEM in two ways; it is global and it only includes the grey level of pixels with a specific class label (the GLEM grey level entropy included the grey level of all pixels within its local window).

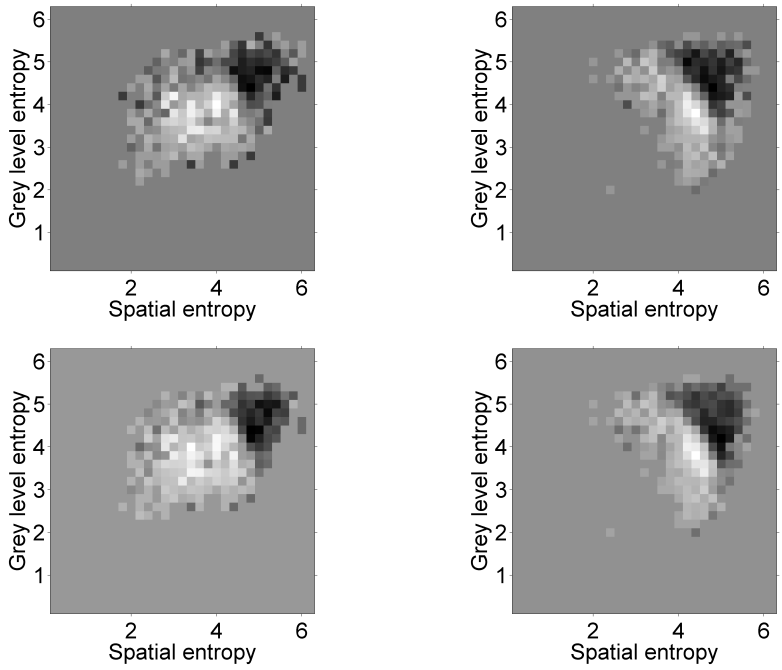

Figure 7.8: The designed weight arrays of the difference: left column) CSDEMdark-feature, right column) CSDEMbright-feature when using the morphological algorithm without the edge removal step and: upper row) all 134 patients, lower row) the 102 patients. Each image is linearly scaled to fill the entire grey level range; the true ranges in the upper row are $[-0.53, 0.53]$ *and* $[-0.72, 0.71]$*, and in the lower row are* $[-0.92, 0.62]$ *and* $[-1.2, 0.91]$*, both for the left and right image, respectively.*

behaviour in light of the similar behaviour for the grey level entropy axis in the GLEM-features and the observation of increased object sizes and number of objects during carcinogenesis (see [8, p.40] and section 4.4), respectively.

Even more interesting is the fact that the plots in figure 7.8 reveal that the effect of an increase in either entropy is nearly identical with respect to the estimated discrimination value for our dataset. Moreover, we see that as we move orthogonally away from the diagonal where the values of the two entropies are equal, the estimated discrimination value remains approximately equal, which indicates that the entropies may be inversely proportional. Also, the support of the weight arrays of the CSDEMdark-features is nearly rectangular, which indicates small correlation between the entropies. Thus we conclude that despite the entropies are attaining values in a similar range, they are not strongly correlated and may even be complimentary to some extent.

If we compare the designed weight arrays for the same segmentation class when using all 134 patients and when using the 102 patients, it is clear that the weight arrays of the 102 patients exhibit the essential prognostic information, while the weight arrays of all 134 patients contains the same information behind some 'noise'. By also considering the true range of the weight arrays, we see

that the corruption affects the discrimination value of the dark part, which is the only part used in the negative adaptive texture feature, far more severely than the bright part. As mentioned in the discussion of the GLEM-features in section 7.2.1, this is a natural consequence of the inclusion of patients which typically are of good prognosis, but have cell images with similar grey level characteristics as the bad prognosis patients. We could stretch this to claim that the bright region of the plots in figure 7.8 corresponds to the normal cell, which we mentioned in section 2.3.2 was likely to be present in even the most essential part of the tumour, while the dark region corresponds to all abnormal cells, which often indicate bad prognosis, especially when only considering patients with diploid and aneuploid histograms (see the used criteria for ploidy classification in section 3.1.3).

We also note that all weight arrays in figure 7.8 are relatively noisy in comparison with the smooth weight arrays of the difference GLEM-feature in figure 7.3). This is unfortunate, though far from unexpected due to the great reduction in number of occurrences in the CSDEM property arrays in comparison with property arrays like the GLEM (see section 4.2). In particular, the weight arrays of the CSDEMdark-features are noisier than the corresponding weight arrays of the CSDEMbright-features, and the weight arrays designed using all 134 patients are far more noisy than the corresponding weight arrays designed using the 102 patients. We therefore suspect that the CSDEM-features, save maybe the CSDEMbright-features when using the 102 patients, are subject to overfitting. We could reduce this problem by reducing the number of quantification levels per integer entropy, but we are already on a relatively low level as the weight arrays in figure 7.8 indicate that most occurrences fall in a region of about 10x10 pixels in the property arrays. Another better alternative is to use non-linear quantification, which will be one suggestion for further work in chapter 9. In light of the detected relation between the two entropies, we will instead reduce the dimension with one by using the CSDEMsum-features instead of the CSDEM-features. This will greatly reduce the risk of overfitting while preserving most or maybe even all relevant information of the CSDEM-features.

Before we study the CSDEMsum-features, let us consider the performance of the individual CSDEM-features. Figure 7.9 shows a scatter plot of the two features when using all 134 patients. This plot indicates that the CSDEM-features of the two segmentation classes are strongly correlated. In general, it is unclear whether including two strongly correlated features in the classifier is better than only using the best of them. For our case, we see that it is possible to draw a straight line in the scatter plot in figure 7.9 which separates slightly better between the classes than projecting to either axis, thus we suspect that including both features may be slightly better. We also see from the plot that the separation of the negative CSDEMbright-feature is slightly better than the negative CSDEMdark-feature. If also including that the corresponding weight arrays in the upper row of figure 7.8 indicate that the negative CSDEMdark-feature should be more overfitting than the negative CSDEMbright-feature, we expect that the classification performance of the negative CSDEMbright-feature is significantly better than the negative CSDEMdark-feature. This also casts doubt on the observed improved separation when including both features, as the slightly better separation may be caused by an overfitted CSDEMdark-feature.

Tables 7.10 and 7.11 show the best classification results of the CSDEMdark-features with respect to both datasets and both the expected CCReq and the

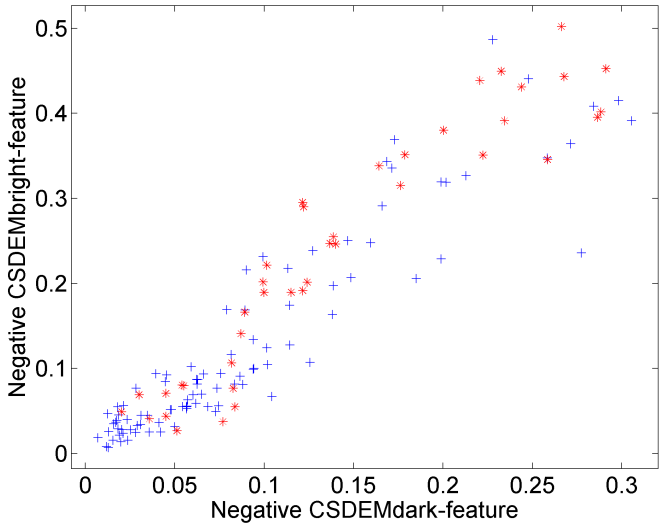

Figure 7.9: Scatter plot of the negative CSDEMbright-feature against the negative CSDEMdark-feature when using the morphological algorithm without the edge removal step and evaluating using all 134 patients. The blue plus sign represents good prognosis and the red asterisk symbol represents bad prognosis.

expected CCR; the negative CSDEMdark-feature. As expected, we see that excluding the CSDEMbright-feature significantly decreases all estimates of the classification performance. We however note that the negative CSDEMdark-feature is not bad in itself, indeed, it is comparable with the negative GLEM-feature (see table 7.6 in section 7.2.1). The features of the two segmentation methods which include the edge removal step are excluded from these tables because they were shown to perform significantly poorer for the CSDEM-features in general, though the best classification results with these features were not significantly poorer than the best classification results of the corresponding features when excluding the edge removal step (they did not perform significantly better either).

When we evaluate the CSDEMbright-features when using the four different segmentation methods, we are again back to the case where the classification performance significantly increases when excluding the edge removal step. However, the gain in expected CCReq when using the difference adaptive texture feature instead of the negative adaptive texture feature is now over a percent for the best classification results when using all 134 patients, thus we should be careful with claiming that there is no significant difference. This is enforced by the fact that it is now the positive adaptive texture features which attain the best expected CCReq when using all 134 patients; 67.5 % with NMSC when excluding the edge removal step and using the morphological algorithm. This is nearly 2 % over the corresponding negative adaptive texture feature, which is the negative CSDEMbright-feature that attains the best expected CCReq when using all 134 patients (65.8 % with NMSC). The best CSDEMbright-feature

Table 7.10: The classification results of the negative CSDEMdark-feature when evaluating on all 134 patients and using the classification method which attained the best expected CCReq; NMSC/LDC. The edge removal step is not included in either segmentation method.

	Morphology	Watershed
CCReq	59.8 % [47.0 %, 73.9 %]	60.5 % [45.5 %, 74.2 %]
CCR	65.7 % [53.8 %, 73.1 %]	66.0 % [53.8 %, 73.1 %]
Specificity	68.4 % [54.5 %, 78.8 %]	68.5 % [53.0 %, 78.8 %]
Sensitivity	51.1 % [25.0 %, 83.3 %]	52.5 % [25.0 %, 83.3 %]

Using 28 learning patterns in each prognosis class.

Table 7.11: The classification results of the negative CSDEMdark-feature when evaluating on the 102 patients and using the classification method which attained the best expected CCReq; NMSC/LDC. The edge removal step is not included in either segmentation method.

	Morphology	Watershed
CCReq	70.9 % [57.5 %, 83.6 %]	71.4 % [56.3 %, 84.8 %]
CCR	78.7 % [67.3 %, 86.5 %]	79.5 % [69.2 %, 88.5 %]
Specificity	84.5 % [68.3 %, 92.7 %]	85.4 % [73.2 %, 95.1 %]
Sensitivity	57.2 % [27.3 %, 81.8 %]	57.5 % [27.3 %, 81.8 %]

Using 25 learning patterns in each prognosis class.

with respect to the expected CCR is also the negative adaptive texture feature, but now using QDC and the separation algorithm based on the watershed transform.

Because of the significant differences in the classification results for different CSDEMbright-features, we have in tables 7.12 and 7.13 collected the best individual classification results of the CSDEMbright-features when using all 134 patients and the 102 patients, respectively, instead of selection a specific adaptive feature type and choosing the best classification method with respect to one performance estimate (which have been the expected CCReq in the previously presented classification results). Note that the results when using all 134 patients are significantly better than the results of using one CSDEM-feature from each segmentation class (see table 7.8), which indicates that the CSDEMdark-features, at least when designed using all 134 patients, are indeed overfitted and therefore result in decreased performance when included, rather than the improved performance we could have expected in light of the scatter plot of figure 7.9 alone.

An inspection of the different classification results of the CSDEM-features reveals that the classification results in table 7.12 and 7.13 are actually equally good or significantly better than all classification results of the CSDEM-features

Table 7.12: The best classification results of the CSDEMbright-features when evaluating on all 134 patients.

CCReq	67.5 % [56.1 %, 78.0 %] (pos., edge kept, morph., NMSC/LDC)
CCR	72.0 % [65.4 %, 79.5 %] (neg., edge kept, water., QDC)
Specificity	75.9 % [66.7 %, 84.8 %] (neg., edge kept, water., QDC)
Sensitivity	68.8 % [41.7 %, 91.7 %] (pos., edge kept, morph., QDC)

Using 28 learning patterns in each prognosis class.

Table 7.13: The best classification results of the CSDEMbright-features when evaluating on the 102 patients.

CCReq	77.2 % [65.4 %, 90.6 %] (neg., edge kept, water., NMSC/LDC)
CCR	83.8 % [76.9 %, 90.4 %] (neg., edge kept, water., NMSC/LDC)
Specificity	89.0 % [82.9 %, 95.1 %] (neg., edge kept, water., QDC)
Sensitivity	67.7 % [36.4 %, 90.9 %] (dif., edge kept, morph., NMSC/LDC)

Using 25 learning patterns in each prognosis class.

and the CSDEMdark-features[5]. These tables thus collect the best obtained classification results for all CSDEM-features, individually or paired, and can therefore also be used for comparison with the subsequent classification results. Note however that the results in table 7.12 and 7.13 are neither with a single feature nor using a single classification method, but the collection of the best results of multiple classifiers.

We mentioned that the best positive CSDEMbright-feature attained significantly better CCReq than the best negative CSDEMbright-feature when using all 134 patients for evaluation. When using the 102 patients, the best negative CSDEMbright-feature is significantly better than the best positive CSDEMbright-feature with respect to both CCReq and CCR. These differences are interesting. With respect to the mentioned claim that the dark and bright regions of the weight arrays correspond to the abnormal and normal cells, respectively, we can interpret this as to indicate that the abnormal cells are definitively best to differentiate between diploid and aneuploid histograms, but the normal cells are better to differentiate between all ploidy histograms. The reason for the latter may be that normal cells are slightly better estimated when evaluating using all 134 patients and that the presence of many abnormal cells implies a relative lack of normal cells and visa versa. The better estimation of normal cells is indicated by the weight array of the difference CSDEMbright-feature de-

[5]In fact, only a single classification result obtained a higher expected value. This one better result was obtained with the positive CSDEM-features when using the morphological algorithm without the edge removal step and using the 102 patients and the Parzen window classifier for evaluation. This classifier attained an expected sensitivity of 68.2 % with the PI [45.5 %, 90.9 %], so it can not be said to be significantly better than the corresponding result of table 7.13.

signed using all 134 patients, see the image in the upper left corner in figure 7.8, which shows that the bright region is slightly smoother than the dark region.

It is unfortunate that a single classifier can not be claimed to be best with respect to both the CCReq and the CCR when using all 134 patients, as it is when using only the 102 patients. Because the positive and negative part of the weight arrays are non-overlapping, we could hope that the positive and negative CSDEMbright-feature could be combined to obtained a single classifier which attains equally good or maybe even better result as the best individual CSDEMbright-feature. This is however not the case; the best combination is using the morphological algorithm without the edge removal step and the NMSC. This attains both the best expected CCReq and expected CCR of all six classification methods, but with an expected CCReq of 67.6 % and an expected CCR of 69.1 %, the performance is significantly poorer with respect to the CCR than the best individual CSDEMbright-feature (the corresponding PIs are also slightly poorer for the combined CSDEMbright-features). It could also be mentioned that combining these two CSDEMbright-features has significantly decreased effect for both the CCReq and the CCR in comparison with the best negative CSDEMbright-feature when evaluating using the 102 patients.

The effect of combining the negative and positive CSDEMbright-features is visualised by a scatter plot of the two CSDEMbright-features in figure 7.10. This plot clearly shows that this combination does not significantly increase the discrimination value in comparison with the positive CSDEMbright-feature. This plot also indicates that the positive and negative CSDEM-features are strongly correlated, which is natural in light of the discussed interpretation

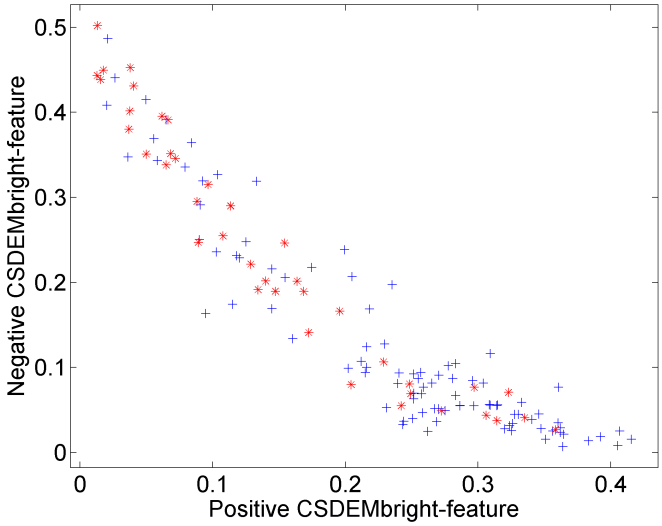

Figure 7.10: Scatter plot of the negative CSDEMbright-feature against the positive CSDEMbright-feature when using the morphological algorithm without the edge removal step and evaluating using all 134 patients. The blue plus sign represents good prognosis and the red asterisk symbol represents bad prognosis.

that the dark and bright regions of the designed weight arrays in figure 7.8 correspond to abnormal and normal cells, respectively, and that the presence of many abnormal cells implies a relative lack of normal cells and visa versa. Finally, the plot also indicates that the positive CSDEMbright-features is better to discriminate between the classes, as also indicated by the fact that it attains a significantly better CCReq, but that the negative CSDEMbright-features may attain better CCR by excluding more of the bad prognosis patients in favour of correctly classifying a larger number of the good prognosis patients.

7.3.1 Assumptions of the estimated Mahalanobis distance between the classes

Figure 7.11 shows the result of testing the underlying assumptions of the estimated Mahalanobis distance between the classes. Similarly to what we observed with the GLEM-feature in connection with figure 7.5, we see from the left and middle column that the normality assumptions are questionable, but not rejected (at significance level 0.05) in the most interesting element with respect to discriminating between the classes (compare with the right column in figure 7.8). However, we also see that the normality assumptions are far more questionable for this CSDEMbright-feature than for the GLEM-feature. Just as the central limit theorem explains that it is natural that the normality assumptions are most appropriate in the most interesting elements (because these also have most occurrences), this difference between the CSDEMbright-feature and the GLEM-feature is a natural consequence of the central limit theorem because of the relatively few occurrences in the CSDEM compared with the GLEM (as commented in section 4.2).

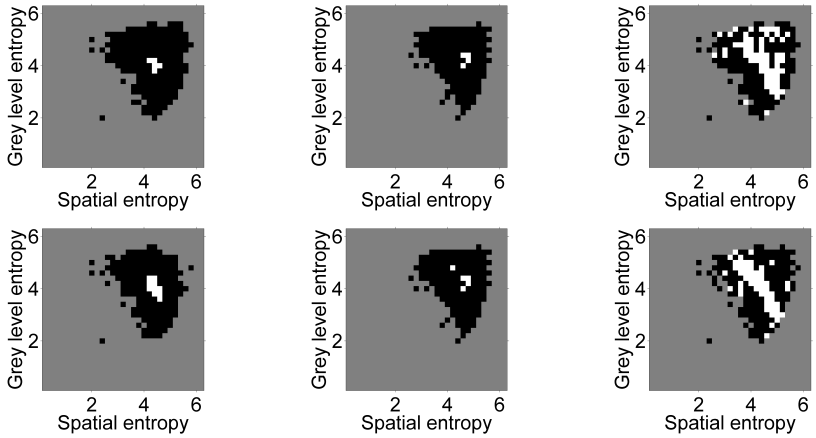

Figure 7.11: The assumption of: left column) normality in good prognosis, middle column) normality in bad prognosis, right column) equal variances of the difference CSDEMbright-feature when using: upper row) all 134 patients, lower row) the 102 patients. The corresponding tests are rejected in black pixels and not rejected in white pixels, both at significance level 0.05. The grey pixels corresponds to elements where all relevant property arrays are zero.

The common variance assumption is more frequently satisfied. In fact, the assumption of the CSDEMbright-feature seem to be approximately equally appropriate as for the GLEM-feature (compare with the right column in figure 7.5). We must however note that because the F-test for equal variances depends strongly on the normality assumption and this assumption is typically not satisfied for the CSDEMbright-feature, we should not blindly trust the conclusion of this test in the abnormal elements. Nevertheless, since the common variance assumption seems to be appropriate in the most interesting elements where the normality assumptions are most appropriate, the validity of the F-test can be expected to be acceptable in at least these elements.

We lastly note that all assumptions seem to be better suited when using the 102 patients in comparison with using all 134 patients. This can be explained by the mentioned observation that the inclusion of patients with tetraploid and polyploid histograms introduces 'noise' with respect to the designed CSDEM weight arrays.

In total, the underlying assumptions of the estimated Mahalanobis distance between the classes seem to be slightly questionable in general when using the CSDEM-features. We should therefore consider other methods for designing the weight arrays, a problem we will return to when we suggest further work in chapter 9. For now we are pleased with the good results of the CSDEM-features, and especially the CSDEMbright-feature, but note that the performance of these features may had been even better if a more appropriate method was used to design the weight arrays.

7.3.2 Comparison with the previously evaluated features

Comparing the negative CSDEMbright-feature when using the algorithm based on the watershed transform without the edge removal step with the best feature based on the grey level entropy matrices, the negative GLEM4D-feature, we note a 1.1 % and 1.5 % increase in expected CCReq and expected CCR, respectively, for the best classification method; the NMSC in both cases (compare table 7.13 with the right column of table 7.7). This increase is barely significant. However, when comparing the positive CSDEMbright-feature when using the morphological algorithm without the edge removal step with the best feature based on the grey level entropy matrices with respect to the expected CCReq when using all 134 patients, the positive GLEM4D-feature, the estimated expected CCReq increased from 64.1 % to 67.5 %, which is highly significant, when using the best classification method; the NMSC in both cases. We thus conclude that the best CSDEM-features, or just the best CSDEMbright-features, are in general significantly better than the best features based on grey level entropy matrices.

The best CSDEM-features are comparable with the combination of the cell features and the best NO-features. With respect to the expected CCReq, the best CSDEM-feature is nearly 3 % less than the combination of the cell features and the best NO-features both when using all 134 patients and when using the 102 patients (compare table 7.4 and 7.12 and table 7.5 and 7.13), thus significantly poorer, but far from the over 6 % difference that was noted when using the best GLEM4D-feature and all 134 patients. However, with respect to the expected CCR, the best CSDEM-features is slightly better than the combination of the cell features and the best NO-features; 1.6 % or 0.8 % when using all 134 patients for our best segmentation method and the segmentation

method using in [49], respectively, and one percent when using the 102 patients (the kNNC attained an expected CCR of 82.8 % with the combination of the cell features and the NO-features when using the morphology algorithm without the edge removal step). It is questionable to call these results significant, but we note that the best CSDEM-features seem to be at least equally good with respect to the CCR as the combination of the cell features and the NO-features.

In total, we note that the classification results of the CSDEM-features are indeed promising. For our dataset, it is the CSDEMbright-features which seem to capture the most relevant prognostic information. The best of these features, the positive CSDEMbright-feature when the CCReq of all 134 patients is of most interest, otherwise the negative CSDEMbright-feature, performs remarkably well. This statement is enforced by the fact that the best CSDEM-feature is only one feature, while the number in combination of the cell features and the best NO-features is seven. Also, because the best CSDEM-feature is only a single feature, we can expect that its generalised performance is good.

7.4 The CSDEMsum-features

We see from the weight arrays in figure 7.8 that it seems reasonable to use the projection onto the diagonal instead of the entire matrix. As noted in

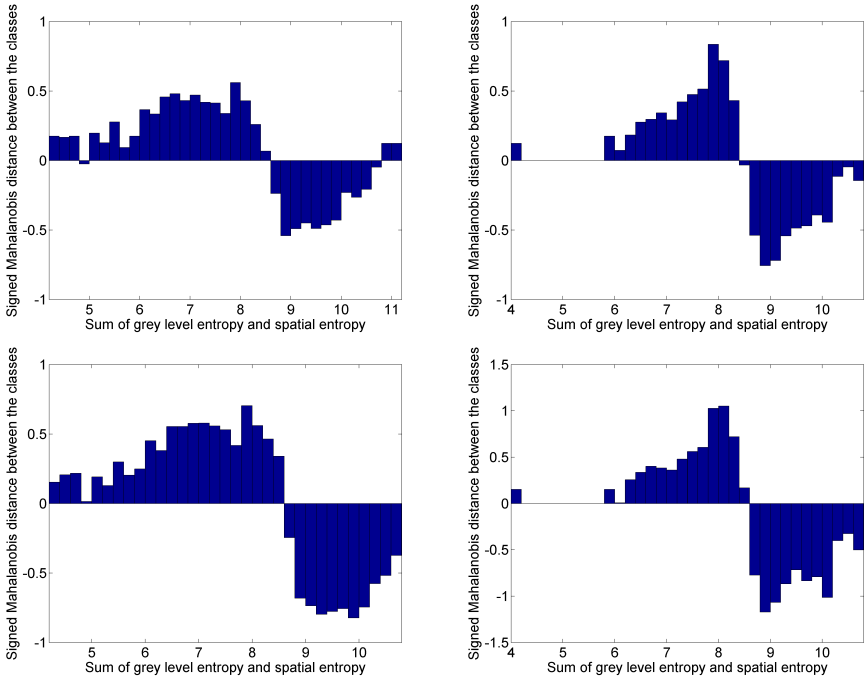

Figure 7.12: The designed weight arrays of the difference: left column) CSDEMsumDark-feature, right column) CSDEMsumBright-feature when using the morphological algorithm without the edge removal step and: upper row) all 134 patients, lower row) the 102 patients.

the discussion of this figure, using the diagonal projecting instead of the entire matrix will greatly reduces the risk of overfitting while likely preserving most prognostic information. This reduced risk is especially important due to the relatively few occurrences of the CSDEMs (see section 4.2), which resulted in the weight arrays in figure 7.8 that showed severe signs of being overfitted, especially for the CSDEMdark-feature and when using all 134 patients.

Before we use the features resulting from the diagonal projection of the CSDEM, the CSDEMsum-features, for classification, we should investigate if the problem with overfitting has been reduced. Figure 7.12 shows the designed weight arrays of the difference CSDEMsum-features when using the morphological algorithm without the edge removal step, just as figure 7.8 did for the CSDEM-features. These plots indeed look promising; the weight arrays of the difference CSDEMsumDark-feature are smooth, even when designed using all 134 patients (in particular in comparison with the same weight array of figure 7.8), and the weight arrays of the difference CSDEMsumBright-feature are even smoother than it was for the difference CSDEMbright-feature. Such smooth weight arrays indicate that the overfitting problem is small and therefore a good generalised performance of the corresponding features.

When using the CSDEMsum-features for classification, we observe a different relation between the four different adaptive texture features than the one seen for the previously evaluated adaptive texture features. The two competing adaptive texture features are now the negative and difference features. When using all 134 patients, the two adaptive texture features are insignificantly different with respect to the CCR, but the difference adaptive texture feature is significantly better with respect to the CCReq. Oppositely, the two adaptive texture features are insignificantly different with respect to the CCReq when using the 102 patients, but the negative adaptive texture feature is now significantly better with respect to the CCR. This indicates that the difference CSDEMsum-features is best when using all patients, but the negative CSDEMsum-features are best when using only the patients with diploid and aneuploid histograms.

Table 7.14 and 7.15 shows the classification results of the difference CSDEMsum-features using all 134 patients and the negative CSDEMsum-features using the 102 patients, respectively. The results of the features which use the segmentation methods that includes the removal of estimated bright primitives sufficiently close to the edge of the nucleus are again excludes, but tested and found to perform significantly poorer than the corresponding features that exclude this step. Comparing the evaluations of the 102 patients when using the negative CSDEMsum-features (see table 7.15) with the same results for the negative CSDEM-features (see table 7.9) or the best CSDEMbright-features (see table 7.13), we see that the performances are equally good. This is not unexpected as the weight array of the difference CSDEMbright-feature in figure 7.8 (bottom right corner) indicates that this feature is not severely subject to overfitting, thus decreasing the risk of overfitting (as the CSDEMsum-features do) has little effect on this performance. However, when comparing the evaluations of all 134 patients when using the difference CSDEMsum-features with the same results for the negative CSDEM-features (table 7.8), we see a highly significant increase in the CCReq; the corresponding expected CCReqs have on average increase of nearly 5 %. Also, the performance with respect to the expected CCReq has significantly increased in comparison with the same results of the best CSDEMbright-feature (see table 7.12) - which also is the best of all individ-

Table 7.14: The classification results of the difference CSDEMsum-features when evaluating on all 134 patients and using the classification method which attained the best expected CCReq; LDC. The edge removal step is not included in either segmentation method.

	Morphology	Watershed
CCReq	69.1 % [55.3 %, 80.3 %]	69.2 % [56.1 %, 79.2 %]
CCR	70.0 % [60.3 %, 78.2 %]	70.6 % [61.5 %, 78.2 %]
Specificity	70.4 % [59.1 %, 80.3 %]	71.2 % [60.6 %, 80.3 %]
Sensitivity	67.8 % [41.7 %, 91.7 %]	67.2 % [41.7 %, 91.7 %]

Using 28 learning patterns in each prognosis class.

Table 7.15: The classification results of the negative CSDEMsum-feature when evaluating on the 102 patients and using the classification method which attained the best expected CCReq; LDC. The edge removal step is not included in either segmentation method.

	Morphology	Watershed
CCReq	76.8 % [63.3 %, 88.1 %]	76.9 % [63.3 %, 89.4 %]
CCR	83.0 % [75.0 %, 90.4 %]	83.9 % [76.9 %, 90.4 %]
Specificity	87.6 % [80.5 %, 95.1 %]	89.0 % [82.9 %, 95.1 %]
Sensitivity	65.9 % [36.4 %, 90.9 %]	64.7 % [36.4 %, 90.9 %]

Using 25 learning patterns in each prognosis class.

ual or paired CSDEM-features. Thus we conclude that the CSDEMsum-features are in general significantly better than any individual or paired CSDEM-features with respect to the CCReq.

We will now study the individual CSDEMsum-features. Because the classification results in table 7.14 and 7.15 indicate that the features based on the watershed transform may be slightly better than the ones based on the morphological algorithm for the CSDEMsum-features, we will base this study on the features which use the watershed transform. Also, because the increased performance of the CSDEMsum-features in comparison with previous adaptive texture features is most evident when using all 134 patients, we will base the study on the difference CSDEMsum-features and typically all 134 patients.

Figure 7.13 shows the scatter plot of the CSDEMsum-features. Interestingly, this plot still indicates that the difference CSDEMsumDark-feature provides valuable prognostic information beyond the difference CSDEMsumBright-feature, as we also saw for the negative CSDEM-features in the scatter plot in figure 7.9, but we now do not expect that the feature of the dark primitive type is overfitted. We therefore expect that both individual CSDEMsum-features perform slightly to significantly worse than the corresponding paired CSDEMsum-features.

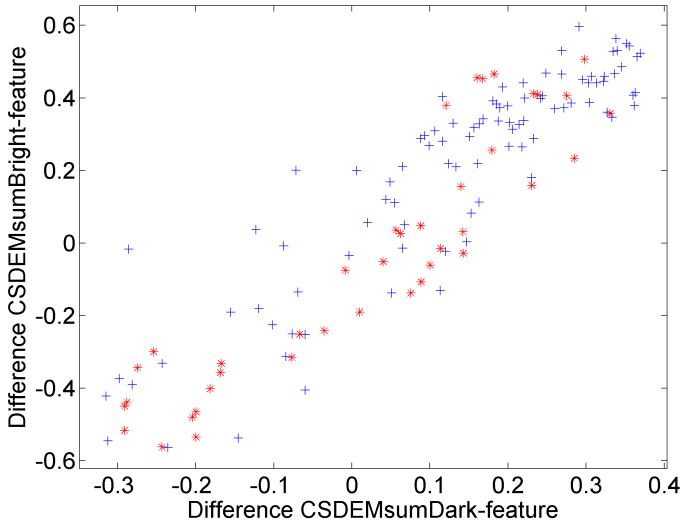

Figure 7.13: Scatter plot of the difference CSDEMsumBright-feature against the difference CSDEMsumDark-feature when using the algorithm based on the watershed transform and not including the edge removal step and evaluating using all 134 patients. The blue plus sign represents good prognosis and the red asterisk symbol represents bad prognosis.

The classification results of the individual CSDEMsum-features confirm this observation. The best result of the CSDEMsumDark-features when using all 134 patients is obtained by the negative CSDEMsumDark-feature with the watershed transformation based algorithm without the edge removal step, resulting in an expected CCReq of 61.2 % (with NMSC) and an expected CCR of 67.7 % (with QDC), closely followed by multiple other segmentation methods with and without the edge removal step. With respect to the 102 patients, the watershed transformation based algorithm *with* the edge removal step comes just on top, resulting in an expected CCReq of 70.4 % and an expected CCR of 77.9 %, both with NMSC and the negative adaptive texture feature. Thus the CSDEMsumDark-features performs significantly worse than the paired CSDEMsum-features.

For the CSDEMsumBright-features, we are again back to the case where the edge removal step significantly decreases the performance. With respect to all 134 patients, the best expected CCReq is 68.4 % with NMSC and the difference CSDEMsumBright-feature when using the morphological algorithm without the edge removal step, while the negative CSDEMsumBright-feature when using the watershed transformation based algorithm without the edge removal step obtains the best expected CCR with QDC; 72.2 %. This result for the expected CCR is not significantly better than corresponding result for the paired CSDEMsum-features, which attains 71.4 % with LDC. For the 102 patients, the negative CSDEMsumBright-feature when using the watershed transformation based algorithm without the edge removal step again obtains the best

expected CCReq and expected CCR; 75.9 % and 82.9 %, both with NMSC. In total, we see that the best CSDEMsumBright-features obtains slightly worse performance than the paired CSDEMsum-features, but not enough to be called significant. However, the best of the CSDEMsumBright-features is not significantly better than the best CSDEMbright-features (see table 7.12 and 7.13), but we have already concluded that the best paired CSDEMsum-features are in general significantly better than the same CSDEMbright-features with respect to the CCReq.

7.4.1 Assumptions of the estimated Mahalanobis distance between the classes

Figure 7.14 shows the result of testing the underlying assumptions of the estimated Mahalanobis distance between the classes. Just as we observed with the CSDEMbright-feature in connection with figure 7.11, most assumptions are rejected (at significance level 0.05), but they seem to be more appropriate at the most interesting elements with respect to discriminating between the classes. As also mentioned with the CSDEMbright-feature, this is only a natural consequence of the central limit theorem as these are the elements with most occurrences. The conclusion that can be drawn from figure 7.14 is also the same as with the CSDEM-features; though the performance of the CSDEMsum-features are indeed good, it may had been even better if a more appropriate method was used to design the weight arrays.

Giving the test results as the colour of the bars in the histogram of the designed weight arrays makes it easier to more precisely interpret the connection between discrimination value and assumption appropriateness. The plots indicate that it is more likely that the assumptions are rejected in the negative region than the positive region. Because it is likely that several normal cells exist within even the most essential part of the tumour, as mentioned in section 2.3.2, this is natural both because it is thus likely to be many occurrences in this positive region and because these elements are likely to have a relative stable estimated probability of occurrence (this estimate may oscillate between zero and some relatively large value in many other elements of the property array). The plots also show that the assumptions are better suited for the CSDEMsumDark-feature than the CSDEMsumBright-feature, which indicates that the property arrays of the CSDEMsumDark-feature are *relatively* more similar than the property arrays of the CSDEMsumBright-feature, a claim that is reasonable also in light of the significantly worse performance of the CSDEMsumDark-feature. Lastly we note that for these features it does not seem to be a significant difference in the appropriateness of the assumptions when designed using all 134 patients and using the 102 patients, which stand in contrast to the observation made in connection with figure 7.11. This may be explained by the fact that the CSDEMbright-feature showed slight signs of being overfitted when using all 134 patients, but not when using the 102 patients (see the right column of figure 7.8), while figure 7.14 indicates no overfitting in either weight array when using the CSDEM-features.

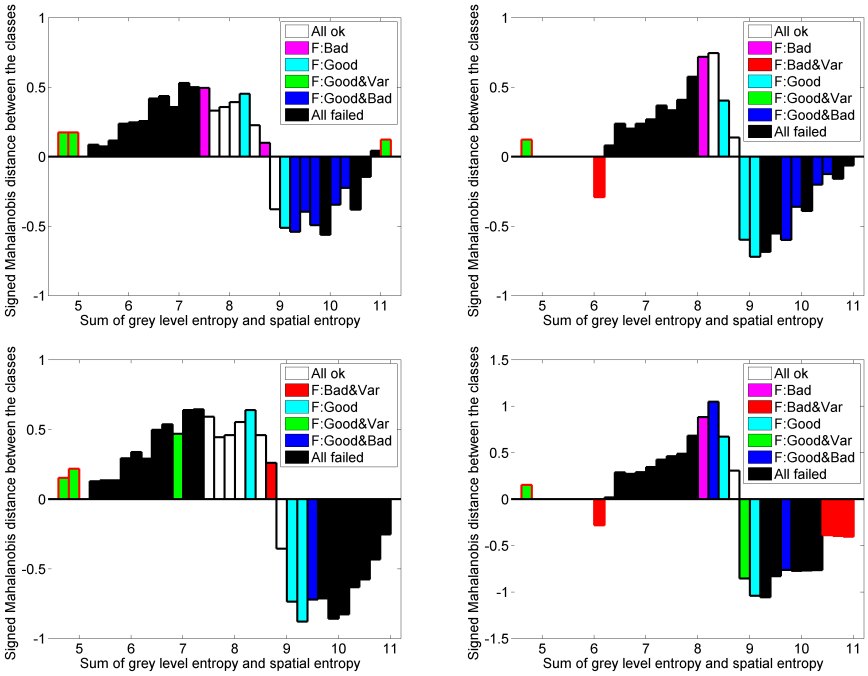

Figure 7.14: Hypothesis test at significance level 0.05 of the underlying assumptions of the estimated Mahalanobis distance between the classes, normality in each prognosis class and equal variances, of the difference: left column) CSDEMsumDark-feature, right column) CSDEMsumBright-feature when using the algorithm based on the watershed transform without the edge removal step and: upper row) all 134 patients, lower row) the 102 patients. For easier interpretation, the test results for each element in the relevant weight array are given by the colour of the face of the corresponding bar in the histogram of the weight array. The colour is coded in RGB according to the result of the test for normality in good prognosis, normality in bad prognosis and equal variances, respectively, and the presence of any colour indicates that the corresponding test is not rejected. As an example, a blue colour means that only the test for equal variances is not rejected; the legend of this colour is F:Good&Bad *to indicate that both normality tests are rejected. Any test where all relevant property arrays are zero is treated as not rejected, but the corresponding bar is marked with a red edge.*

7.4.2 Comparison with the combination of the cell features and the NO-features

We have already compared the CSDEMsum-features with the other choices of property arrays and concluded that the CSDEMsum-features in general perform significantly better than all other choices with respect to the CCReq. We have however not compared with the combination of the cell features and the best NO-features, which is the feature set that attains the best overall performance for our dataset in our evaluations.

Comparing the results when using all 134 patients, it is questionable to conclude that any particular feature set is significantly better. The combination of cell features and the best NO-features obtains expected CCReq and expected CCR of 70.4 % with NMSC, while the best CSDEMsum-features obtains 69.2 % and 70.6 %, respectively, with the difference feature and LDC, both using the algorithm based on the watershed transform and without the edge removal step. The expected CCR of the corresponding negative feature is 71.4 %, still with the same segmentation method and classification method. From these results we may suspect that the combination of the cell features and the best NO-features are slightly better than the CSDEMsum-features with respect to the CCReq, but we will not call this difference significant.

If evaluating using the 102 patients, the combination of cell features and the best NO-features can be said to be significantly better than the best CSDEMsum-features with respect to the CCReq. Tables 7.5 and 7.15 both show the best results with respect to the expected CCReq, and we see here that the difference is nearly 3 %, thus significant. As previously noted, the best expected CCR of the combination of cell features and NO-features were 82.8 % with kNNC and the morphological algorithm without the edge removal step, thus the best CSDEMsum-features may be slightly better with respect to CCR (see table 7.15), but we will not call this difference significant.

In total, the combination of the cell features and the best NO-features are equally good to classify all patients, but significantly better to classify the patients with diploid and aneuploid histograms with respect to the CCReq, both in comparison with the best CSDEMsum-features. We are pleased with this result, especially because the best CSDEMsum-feature are only two features, while the other feature set contains seven features. Furthermore, we have noted that the best CSDEMsumBright-feature is almost as good as the CSDEMsum-features, and in particular not significantly worse, thus the performance of the single best feature based on the CSDEM is nearly as good as the combination of the cell features and the best NO-features.

7.5 Combining features

Because of our great concern for the overfitting problem, we have no interest in forming a large set of more or less randomly chosen feature candidates and applying some technique to compute or select the best performing feature set for our dataset. In fact, we do not wish to use dimension reduction or feature selection at all, as the combination of such techniques and the bootstrap method results in the use of multiple feature sets and thus in a sense unrealistic classification results[6]. The result is that we must pick features by hand and even this should not be done too extensively, as we may cause overfitting also by manually selecting too many feature sets for evaluation. This latter problem is naturally dealt with by restricting the set of feature candidates to the relatively few features we have mentioned to this point; the cell features, the NO-features and the adaptive texture features.

[6]This problem is only present when performing the dimension reduction or feature selection for each bootstrap. However, performing such techniques prior to the use of any iterative evaluation method will introduce a significant (optimistic) bias [59, pp.95–97], thus this is not even an option for us.

We will further restrict the set of adaptive texture features to only include the most promising choice of property array within each of the two discussed types of property arrays; the GLEM4D-features and the CSDEMsum-features computed using either separation algorithm, but without the edge removal step. Table 7.16 shows the estimated Mahalanobis distance of each of these adaptive texture features for the adaptive texture feature type which attained the maximum estimated expectation of the four features in the set described in section 3.2.3, along with the same estimate of each cell feature, each NO-feature with each of the two relevant segmentation methods and six interesting feature sets. The individual estimates indicate that the Eccentricity-feature and the CSDEMsumBright-features are very good and should discriminate well be-

Table 7.16: The estimated Mahalanobis distance between the classes with an estimate of the 95 % two-sided PI for all relevant individual features and six interesting feature sets when using all 134 patients. The Mahalanobis distance between the classes is estimated using each bootstrapped validation dataset, and the expectation is estimated as the mean of the 500 bootstraps and the PI limits as the 2.5 % and 97.5 % percentiles of the empirical distribution. The table is sorted on the estimated expectation (secondly on the estimated lower PI limit).

Feature	Mahalanobis d.
Cell & NO-*features* (morph.)	1.33 [0.84, 2.00]
Cell & NO-*features* (water.)	1.32 [0.80, 1.94]
Cell *features*	1.20 [0.71, 1.80]
CSDEMsumBright (neg., water.) & Eccentricity	1.09 [0.61, 1.57]
CSDEMsumBright (dif., morph.) & Eccentricity	1.07 [0.58, 1.61]
GLEM4D (dif.) & Eccentricity	1.03 [0.55, 1.57]
Eccentricity	0.84 [0.35, 1.28]
CSDEMsumBright (dif., morph.)	0.79 [0.30, 1.29]
CSDEMsumBright (dif., water.)	0.77 [0.28, 1.28]
GLEM4D (dif.)	0.65 [0.17, 1.17]
Compactness	0.61 [0.31, 0.93]
CSDEMsumDark (neg., water.)	0.56 [0.12, 1.03]
GreyLevelAverage	0.53 [0.05, 1.05]
CSDEMsumDark (dif., morph.)	0.52 [0.07, 1.01]
Area	0.48 [0.04, 1.07]
NumberOfDarkObjects (morph.)	0.47 [0.05, 0.95]
NumberOfDarkObjects (water.)	0.47 [0.04, 1.02]
NumberOfBrightObjects (morph.)	0.43 [0.02, 1.04]
NumberOfBrightObjects (water.)	0.42 [0.02, 0.90]
GreyLevelVariance	0.41 [0.02, 0.93]

tween the classes. Also the GLEM4D-features, the Compactness-feature, the CSDEMsumDark-features and the GreyLevelAverage-feature seem reasonable, especially the Compactness-feature in light of its relatively high estimated lower PI limit. While these estimates are likely very representative for comparing individual features, it is as noted in section 6.5 more important to pay attention to the cooperation of features when attempting to find the optimal set of features.

When studying the adaptive texture features, we will further reduce our attention to the generally best performing GLEM4D-feature, the negative GLEM4D-feature, and to the insignificant worse performing subfeature of the CSDEMsum-features, the difference CSDEMsumBright-feature[7] when using the algorithm based on the watershed transform without the edge removal step. The classification results of the negative GLEM4D-feature are given in table 7.7. The best results of the difference CSDEMsumBright-feature when using all 134 patients is a CCReq of 68.3 % [57.6 %, 79.2 %] with NMSC and a CCR of 71.5 % [62.8 %, 79.5 %] with QDC, while the negative CSDEMsumBright-feature obtains a CCReq of 75.9 % [64.2 %, 88.5 %] and a CCR of 82.9 % [76.9 %, 90.4 %], both with the NMSC and when using the 102 patients.

We have seen that the adaptive texture features perform well, especially in comparison with the resulting low dimension of the feature space. However, the combination of the cell features and the best NO-features still performs equally well or, in most cases, significantly better. When attempting to combine features, it is thus naturally to begin with each of the relevant adaptive texture features and then consider the prognostic value of including some other adaptive texture feature, a cell feature or a NO-feature. We note that despite this angle of approach, it is not an aim to increase the performance of classifiers which includes each relevant adaptive texture feature, the aim is to obtain a classifier which performs significantly better than all previously evaluated classifiers (with respect to some performance estimate).

7.5.1 GLEM4D-features

From the relatively strong correlation between the negative GLEM4D-feature and the Area-feature which is evident for the scatter plot in figure 7.6 in section 7.2.1, there should be no reason to include the Area-feature with the GLEM4D-features. This is enforced by the fact that a straight line (or any other reasonably simple curve) can not be drawn in the scatter plot which significantly increases the separation between the classes. This is also true for the relation between the negative GLEM4D-feature and the GreyLevelAverage- and GreyLevelVariance-feature (see figure 7.7 in the same section), though both of these feature pairs are much less correlated. These observations have been verified to also be true for the corresponding scatter plots of all 134 patients, thus we will not include any of these three feature with the GLEM4D-features.

Figure 7.15 shows the scatter plots of the negative GLEM4D-feature with each of the two remaining cell features, the Compactness- and the Eccentricity-

[7]The negative CSDEMsumBright-feature performs generally slightly better than the difference feature when evaluating using the 102 patients. However, we will investigate all 134 patients when studying the adaptive texture features, thus we will and should use the difference CSDEMsumBright-feature for this purpose as it performs significantly better than the corresponding negative feature. We however note that when our feature candidates are located, we will evaluate using each adaptive feature type for each dataset.

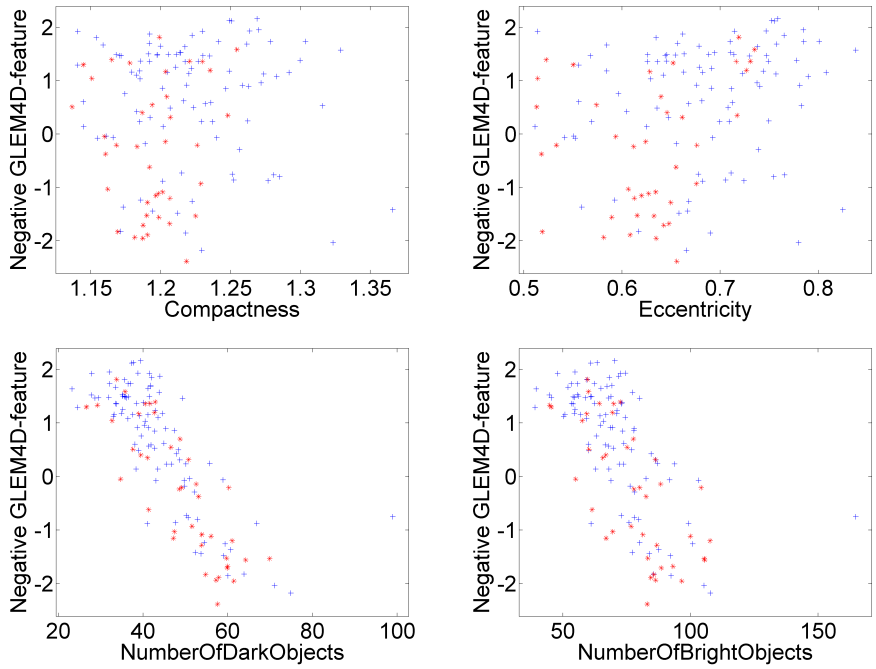

Figure 7.15: Scatter plot of the negative GLEM4D-feature against: first row) the Compactness- and the Eccentricity-feature, second row) the NumberOfDarkObjects- and the NumberOfBrightObjects-feature for the left and right plot, respectively, and when using all 134 patients. The NO-features are computed using our watershed segmentation method without the step which removes bright edge objects. The blue plus sign represents good prognosis and the red asterisk symbol represents bad prognosis.

feature, and each of the two NO-features for our watershed segmentation method without the edge removal step. It is clear from the scatter plots with the NO-features that these features are correlated with the negative GLEM4D-feature, and also that they do not provide any significant prognostic information beyond the negative GLEM4D-features. We therefore conclude that there is no reason to include these features with the negative GLEM4D-features.

The situation is far better for both geometrical features, i.e. the Compactness- and the Eccentricity-feature. Both these features seem to be independent of the negative GLEM4D-feature, and they also seem to be able to provide some new prognostic information beyond the negative GLEM4D-features. They are however not independent of each other. Their quadratical relation can be deduced from their definitions and is also indicated by their scatter plot for all 134 patients in figure 7.16. This scatter plot also indicates that we should only include the best of them. However, there is a minor tendency that for the same eccentricity, the compactness of patients with bad prognosis is slightly higher than for the patients with good prognosis, thus we will in the following also include evaluations of the feature sets which include both geometrical features.

For comparison, the classification results of using either or both geometrical

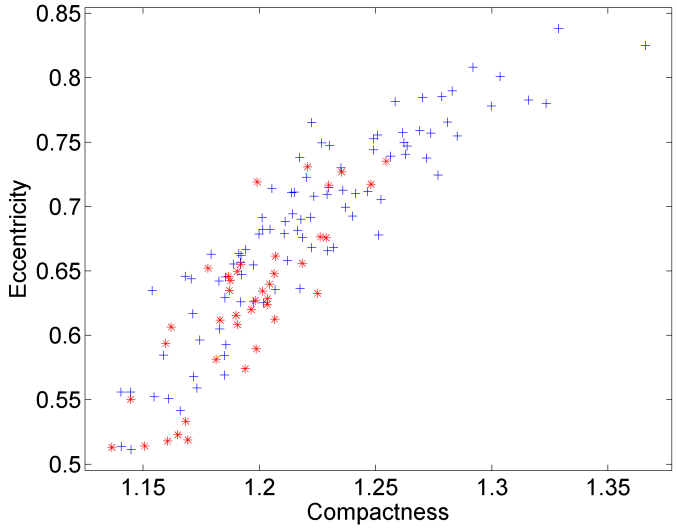

Figure 7.16: Scatter plot of the Eccentricity-feature against the Compactness-feature when using all 134 patients. The blue plus sign represents good prognosis and the red asterisk symbol represents bad prognosis.

features are shown in table 7.17. These results show that both features are good, but the Eccentricity-feature is the better. This observation can also be made from the scatter plot in figure 7.16. With respect to including both geometrical features or only the best of them, we note the decreased performance when combining the features and using all 134 patients in comparison with only using the Eccentricity-feature. We however see that the performance does not decrease when using the 102 patients, in fact, it significantly increases with respect to the CCR. We will therefore continue to include the results when including both geometrical features, even though we still suspect that these will typically be similar to the ones obtained when only including the best geometrical feature. It is lastly interesting to note that the Eccentricity-feature performs better with respect to the CCReq than the combination of all cell features and even significantly better if using all 134 patients. It is therefore uncertain whether the other cell features have true prognostic value beyond the Eccentricity-feature.

Based on the relatively good performance of the Eccentricity-feature, it is interesting to briefly investigate what it tells us. We see from the scatter plot in figure 7.16 that low eccentricity indicates bad prognosis. By using the definition of eccentricity, this tells us that the patients with bad prognosis typically have nuclei with more circular shape than the patients with good prognosis. As the eccentricity is measured by fitting the nucleus to an ellipse, it is interesting to see how good this fit is as this will indicate whether non-circular means not circular, but still elliptical or really just abnormally shaped. Because the nuclei are not likely to be shaped like a parallelepiped (this is also indicated by an inspection of the dataset), we expect that the goodness of fit can be seen by the solidity of the nuclei; a high solidity (near 1) will indicate a good fit, while a

Table 7.17: The classification results of the Compactness- and the Eccentricity-feature when using the classification method which attained the best expected CCReq; LDC. This classification method also attained the best expected CCR and best expected specificity in all cases.

	All 134 patients Compactness	The 102 patients Compactness
CCReq	65.9 % [52.7 %, 77.7 %]	69.4 % [57.4 %, 79.6 %]
CCR	60.7 % [50.0 %, 69.2 %]	65.3 % [53.8 %, 75.0 %]
Specificity	58.4 % [43.9 %, 71.2 %]	62.3 % [48.8 %, 73.2 %]
Sensitivity	73.3 % [41.7 %, 91.7 %]	76.4 % [54.5 %, 100.0 %]
	All 134 patients Eccentricity	The 102 patients Eccentricity
CCReq	70.5 % [58.3 %, 81.8 %]	77.1 % [64.7 %, 86.9 %]
CCR	69.0 % [59.0 %, 76.9 %]	75.8 % [65.4 %, 84.6 %]
Specificity	68.4 % [56.1 %, 80.3 %]	74.9 % [63.4 %, 85.4 %]
Sensitivity	72.6 % [41.7 %, 100.0 %]	79.3 % [54.5 %, 100.0 %]
	All 134 patients Comp.&Ecc.	The 102 patients Comp.&Ecc.
CCReq	69.3 % [55.7 %, 81.1 %]	77.1 % [64.7 %, 89.0 %]
CCR	68.8 % [59.0 %, 78.2 %]	78.1 % [67.3 %, 86.5 %]
Specificity	68.5 % [54.5 %, 80.3 %]	78.8 % [63.4 %, 90.2 %]
Sensitivity	70.1 % [41.7 %, 91.7 %]	75.4 % [45.5 %, 100.0 %]

Using 28 (left) and 25 (right) learning patterns in each prognosis class.

low solidity will indicate a bad fit. The scatter plot of the Eccentricity-feature against the average solidity within each patient is shown in figure 7.17. This indicates that the fit is actually really good, thus we can conclude both that the foundation of the Eccentricity-feature is valid and that the non-circular shapes which indicates good prognosis are reasonably well described by fitted ellipses.

The best GLEM4D-feature in combination with either or both geometrical features was the negative GLEM4D-feature when using either dataset and with respect to both the expected CCReq and expected CCR. The classification results of these combinations are shown in table 7.18. With respect to all 134 patients, we see that the inclusion of the best GLEM4D-feature caused an increase of the best expected CCReq with 0.1 %, which is clearly insignificant. The best combination with respect to the expected CCR, the combination of the negative GLEM4D-feature and the Eccentricity-feature, obtained an expected CCR of 71.3 %, which is a significant improvement from the corresponding best geometrical feature set; the Eccentricity-feature with its expected CCR of 69.0 %. While the total improvement of including the best GLEM4D-feature is not convincing with respect to all 134 patients, we note that the performance is indeed very good and in particular better than the performance of the combi-

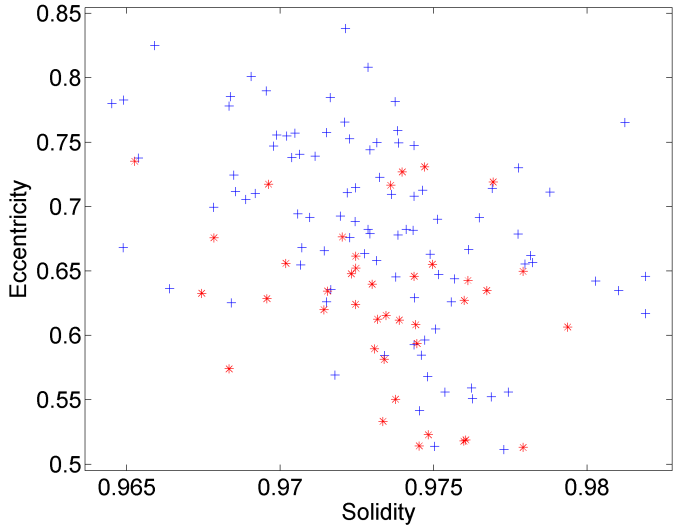

Figure 7.17: Scatter plot of the Eccentricity-feature against the average solidity within each patient when using all 134 patients. The blue plus sign represents good prognosis and the red asterisk symbol represents bad prognosis.

nation of the cell features and the best NO-features (see table 7.4), though not significantly better.

The improvement of including the GLEM4D-feature is far better with respect to the 102 patients. A stable increase of most performance estimates is noted and the increase of CCReq and CCR are significant for all inclusions. The best expected CCReq of 79.8 % and best expected CCR of 82.3 % is also very good and in particular not significantly different for the combination of the cell features and the best NO-features, which as mentioned was an expected CCReq of 79.8 % and an expected CCR of 82.8 %.

In conclusion, the best combination of the GLEM4D-features and either or both geometrical features is the negative GLEM4D-feature and the Eccentricity-feature (this combination also attained an expected CCR of 82.3 % when using the 102 patients and the Parzen window classifier). The results of this feature set are insignificantly different for the combination of the cell feature and the best NO-features. This is very good and in particular respectable as this feature set only contains two features, while the comparing feature set contains seven features. It is however not unexpected. This is because we noted that the Eccentricity-feature was equally good or maybe even better with respect to the CCReq than all cell features. Moreover, the negative GLEM4D-feature was found to have large correlation the NO-features (see the lower row in figure 7.15), which was the features that resulted in significant increased performance when combined with the cell features. Therefore, the combination of the negative GLEM4D-feature and the Eccentricity-feature can simply be seen as a compact feature set describing the same properties as the combination of the cell features and the NO-features.

Table 7.18: The classification results of the negative GLEM4D-feature with the Compactness- and/or the Eccentricity-feature when using the classification method which attained the best expected CCReq; the QDC when only combining with the Compactness-feature and using the 102 patients, the LDC when only combining with the Compactness-feature and using all 134 patients or when combining all three features and using the 102 patients, otherwise NMSC (for the three remaining combinations).

	All 134 patients GLEM4D&Comp.	The 102 patients GLEM4D&Comp.
CCReq	69.6 % [58.0 %, 79.9 %]	78.3 % [65.7 %, 90.6 %]
CCR	70.3 % [60.3 %, 78.2 %]	82.3 % [69.2 %, 90.4 %]
Specificity	70.6 % [57.6 %, 81.8 %]	85.2 % [65.9 %, 95.1 %]
Sensitivity	68.6 % [41.7 %, 91.7 %]	71.4 % [45.5 %, 90.9 %]
	All 134 patients GLEM4D&Ecc.	The 102 patients GLEM4D&Ecc.
CCReq	70.5 % [58.0 %, 80.7 %]	79.8 % [67.8 %, 90.2 %]
CCR	70.2 % [62.8 %, 78.2 %]	81.2 % [73.1 %, 88.5 %]
Specificity	70.0 % [59.1 %, 80.3 %]	82.3 % [70.7 %, 92.7 %]
Sensitivity	70.9 % [41.7 %, 91.7 %]	77.4 % [54.5 %, 100.0 %]
	All 134 patients GLEM4D&Comp.&Ecc.	The 102 patients GLEM4D&Comp.&Ecc.
CCReq	70.6 % [58.3 %, 81.8 %]	78.6 % [66.3 %, 90.6 %]
CCR	68.7 % [57.7 %, 78.2 %]	80.6 % [69.2 %, 88.5 %]
Specificity	67.8 % [54.5 %, 78.8 %]	82.0 % [65.9 %, 92.7 %]
Sensitivity	73.4 % [50.0 %, 100.0 %]	75.1 % [45.5 %, 100.0 %]

Using 28 (left) and 25 (right) learning patterns in each prognosis class.

7.5.2 CSDEMsum-features

To find which features that should be attempted to be combined with the CSDEMsum-features, we will as mentioned in the introduction of this section only study the difference CSDEMsumBright-feature when using the algorithm based on the watershed transform without the edge removal step. Figure 7.18 shows the scatter plots of this difference CSDEMsumBright-feature against the negative GLEM4D-feature, each of the cell features and each of the corresponding NO-features. These plots provide much information, not only about which features we should attempt to combine the CSDEMsum-features with, but also about what the CSDEMsum-features (or at least the difference CSDEMsumBright-feature) measure.

First of all, the CSDEMsumBright-feature under study is relatively strongly correlated with the GLEM4D-feature and the inclusion of both these features is evidently meaningless for our dataset. Because of this correlation, we expect that the CSDEMsumBright-feature is also correlated with the Area-, the

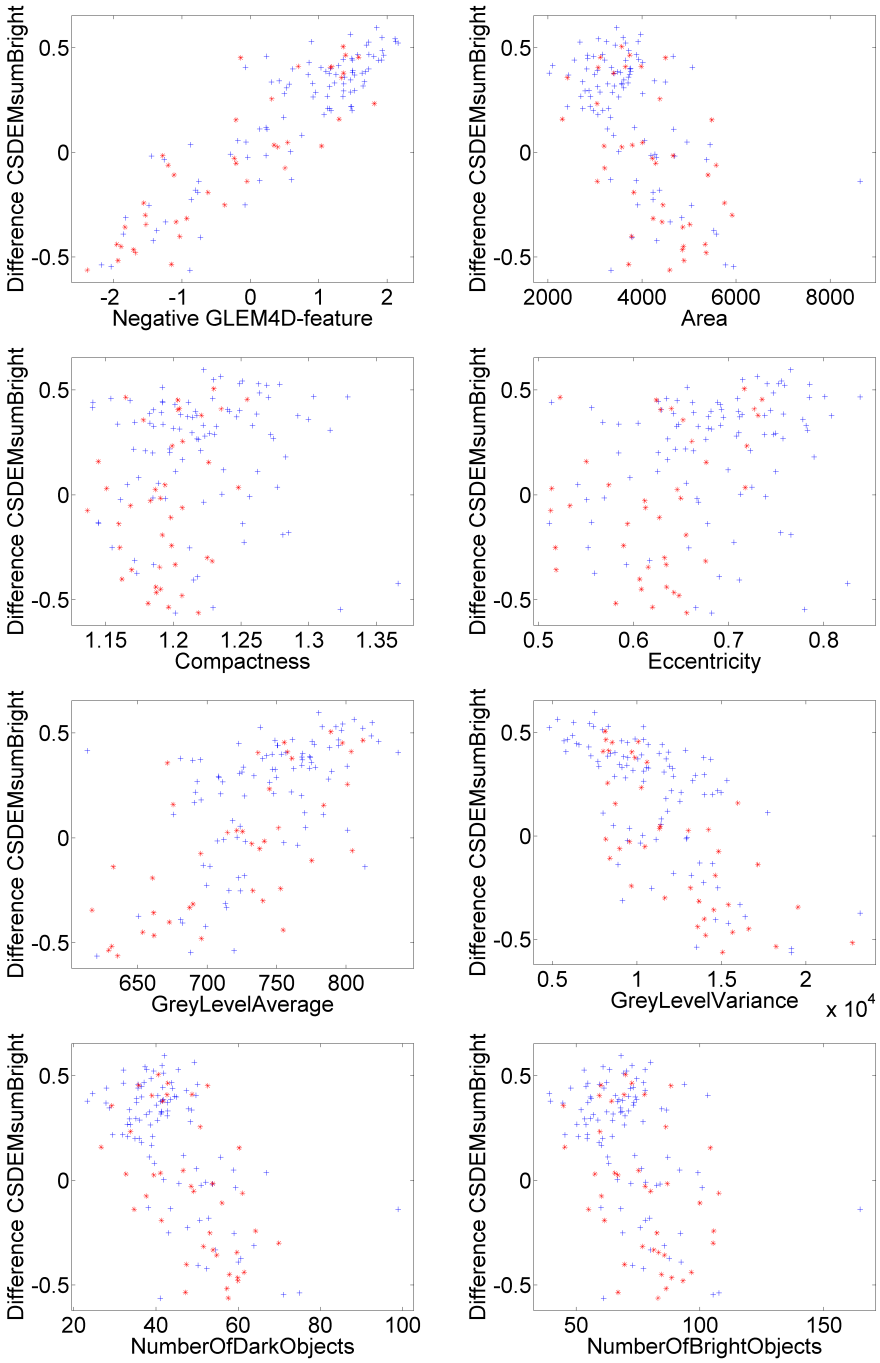

Figure 7.18: Scatter plot of the difference CSDEMsumBright-feature against: first row) the negative GLEM4D- and the Area-feature, second row) the Compactness- and the Eccentricity-feature, third row) the GreyLevelAverage- and the GreyLevelVariance-feature, fourth row) the NumberOfDarkObjects- and the NumberOfBrightObjects-feature for the left and right plot, respectively, and when using all 134 patients. The difference CSDEMsumBright-feature and the NO-features are computed using our watershed segmentation method without the step which removes bright edge objects. The blue plus sign represents good prognosis and the red asterisk symbol represents bad prognosis.

GreyLevelAverage- and the GreyLevelVariance-feature and each of the NO-features. Indeed, the scatter plots of figure 7.18 show that all these correlations are present. More precisely, the mentioned CSDEMsumBright-feature is correlated with the Area-feature, but not as much as the negative GLEM4D-feature, but it is on the other hand more correlated with the GreyLevelAverage- and the GreyLevelVariance-feature. More surprisingly, while the CSDEMsumBright-feature is correlated with the NO-features, this correlation is less than the corresponding correlation of the negative GLEM4D-feature. The result with respect to finding possible feature candidates is however equal; there seems to be no reason to include any of these features in combination with the CSDEMsum-features.

We are thus again left with the geometrical features, which seem to be independent of also this adaptive texture feature. They also seem to provide some new prognostic information beyond the mentioned CSDEMsumBright-feature, though not as much as we observed with the negative GLEM4D-feature. This reduced significance may be caused by the fact that the CSDEMsumBright-feature is significantly better than the negative GLEM4D-feature, even though they are relatively strongly correlated, and that some of the additional prognostic information that the CSDEMsumBright-feature measures may also be measured by the geometrical features.

Turning to the classification results of the combination of either or both geometrical features with the CSDEMsum-features, the best results when using all 134 patients are obtained with the CSDEMsumBright-features in comparison with the corresponding CSDEMsumDark-features and the CSDEMsum-features. It is moreover the negative CSDEMsumBright-feature when using the algorithm based on the watershed transform that obtains the best combination results among the CSDEMsumBright-features[8], but not significantly better than the corresponding difference adaptive texture feature nor when using the corresponding features based on the morphological algorithm. When evaluating on the 102 patients, the relative performance of the CSDEMsum-, CSDEMsumDark- and CSDEMsumBright-features are more similar when combined with either or both geometrical features, but the combined performances of the CSDEMsumDark-features are still significantly worse. With respect to the expected CCReq, it is generally the CSDEMsum-features that performs best in combination with the geometrical features. Among these, it is again the negative adaptive texture feature when using the algorithm based on the watershed transform that performs the best, again typically insignificantly better than the corresponding difference adaptive texture feature and when using the corresponding features based on the morphological algorithm.

The classification results of the combination of either or both geometrical features with the negative CSDEMsumBright-feature when using all 134 patients and the negative CSDEMsum-features when using the 102 patients are shown in table 7.19. We see from these results that it is the combination with the Eccentricity-feature that generally performs the best, just as what we ob-

[8] As mentioned in the introduction of this section, we have only considered the CSDEMsum-features when using the segmentations methods that excludes the step that removes estimated bright primitives sufficiently close to the edge of the nucleus. All comparisons and claims of best classification results in this subsection are therefore restricted to the segmentation methods which excludes this step, though we from the previous results expect that the corresponding performances when including this step would only have been worse.

Table 7.19: The classification results of the negative CSDEMsumBright-feature when using all 134 patients and the negative CSDEMsum-features when using the 102 patients, both in combination with the Compactness- and/or the Eccentricity-feature and when using the classification method which attained the best expected CCReq; the QDC when only combining with the Compactness-feature and using the 102 patients or when combining all three features and using all 134 patients, the LDC when combining with the Eccentricity-feature or both geometrical features and using the 102 patients, otherwise NMSC (for the two remaining combinations). Both the negative CSDEMsumBright-feature and the negative CSDEMsum-features are computed using the algorithm based on the watershed transform without the edge removal step.

	All 134 patients CSDEMsumB.&Comp.	The 102 patients CSDEMsum&Comp.
CCReq	70.0 % [58.7 %, 81.1 %]	76.3 % [62.3 %, 89.4 %]
CCR	71.3 % [60.3 %, 80.8 %]	80.6 % [65.4 %, 90.4 %]
Specificity	71.8 % [57.6 %, 83.3 %]	83.7 % [61.0 %, 95.1 %]
Sensitivity	68.1 % [41.7 %, 91.7 %]	68.9 % [45.5 %, 90.9 %]
	All 134 patients CSDEMsumB.&Ecc.	The 102 patients CSDEMsum&Ecc.
CCReq	70.8 % [59.8 %, 80.7 %]	77.2 % [65.4 %, 89.4 %]
CCR	71.4 % [62.8 %, 80.8 %]	80.3 % [69.2 %, 88.5 %]
Specificity	71.7 % [62.1 %, 83.3 %]	82.6 % [65.9 %, 92.7 %]
Sensitivity	69.9 % [50.0 %, 91.7 %]	71.8 % [45.5 %, 90.9 %]
	All 134 patients CSDEMsumB.&C.&E.	The 102 patients CSDEMsum&C.&E.
CCReq	69.3 % [56.4 %, 80.3 %]	77.3 % [65.1 %, 89.0 %]
CCR	70.3 % [61.5 %, 78.2 %]	80.4 % [69.2 %, 88.5 %]
Specificity	70.7 % [59.1 %, 81.8 %]	82.7 % [65.9 %, 92.7 %]
Sensitivity	67.9 % [41.7 %, 91.7 %]	71.9 % [45.5 %, 100.0 %]

Using 28 (left) and 25 (right) learning patterns in each prognosis class.

served in connection with the GLEM4D-features (see table 7.18). However, we note that the best expected CCReq when using the 102 patients has not changed significantly from the results with only geometrical features (see table 7.17). This is disappointing, especially in light of the significant increase we observed when combining the GLEM4D-features with these geometrical features. The best expected CCReq is also not significant better than with the CSDEMsum-features (see table 7.15), and the best expected CCR actually decreases significantly in comparison with the results of the CSDEMsum-features. We therefore conclude that combining either or both geometrical features with the CSDEMsum-features will generally decrease the performance when using only patients with diploid or aneuploid histograms.

When evaluating using all patients, we note a significant increase in expected CCReq from the best CSDEMsumBright-feature (which as mentioned attained an expected CCReq of 68.4 %) and also from the best CSDEMsum-features (see table 7.15). The best expected CCReq is however not significantly better than with the geometrical features, alone (see table 7.17) or in combination with the GLEM4D-features (see table 7.18), thus the Eccentricity-feature may be the most contributing feature to the good expected CCReq of 70.8 % in table 7.19.

The best expected CCR when using all patients is obtained with the negative CSDEMsumBright-feature when using the algorithm based on the watershed transform in combination with the Eccentricity-feature, which gives an expected CCR of 72.6 % with the Parzen window classifier. This is slightly better than the best expected CCR of all features based on the sum histogram of CSDEMs, which as mentioned was 72.2 %, and may even be significantly better than the best combination of the geometrical features and the GLEM4D-features, which as mentioned was 71.3 %. Because the Eccentricity-feature alone only attains an expected CCR of 69.0 % when using all patients (see table 7.17), the CSDEMsumBright-feature may be the most contributing feature to this good performance estimate.

In total, we can claim that the combination of the negative CSDEMsumBright-feature when using the algorithm based on the watershed transform and the Eccentricity-feature is the generally best performing feature set when using all patients, attaining an expected CCReq of nearly 71 % when using NMSC and nearly 73 % when using the Parzen window classifier. This is slightly or significantly better than the best performance of all other evaluated feature sets. In particular, it is slightly better than the performance of the combination of all cell features and the best NO-features with respect to the expected CCReq, and significantly better with respect to the expected CCR.

If excluding the patients with tetraploid or polyploid histograms, we have noted that the best performance of the CSDEMsum-features is obtained when used alone. The performance of this feature set is still good, attaining an expected CCR of 83.9 %, which is slightly or significantly better than the best expected CCR of all other evaluated feature sets. In particular, this may be significantly better than the combination of the cell features and the best NO-features, which was 82.8 %, and is significantly better than the best combination of the geometrical features and the GLEM4D-features, which was 82.3 %. However, the best expected CCReq of the CSDEMsum-features is only 76.9 %, which is significantly worse than with the combination of the cell features and the best NO-features, and also the best combination of the geometrical features and the GLEM4D-features.

7.6 Classifier complexity and classification method

The last section concluded the investigation of new and improved classifiers, i.e. combinations of a feature set and a classification method. The rest of the chapter will be devoted to give a better understanding of some related aspects of the classifiers that have not jet been discussed. We will begin this discussion by considering the classifier complexity and the classification methods.

When we discussed overfitting in section 6.3, we mentioned that the number of features and complexity of the classification method are essential factors for

the classifier complexity. We also found indications supporting a belief that the optimal classifier complexity may be sufficiently prominent to be reasonably estimated for a given number of learning patterns. We will therefore attempt to estimate this optimal complexity for our datasets. It should however be repeated that because the optimal classifier complexity is a trade-off between the decreased performance caused by more estimation and the increased performance caused by the added complexity, the optimal classifier complexity is not completely determined for a given learning dataset. In particular, the true distribution of the conditional pdfs and the effectiveness of the features are relevant.

The previous sections reveals that the best classification method is typically different with respect to the CCReq and to the CCR. Before we attempt to find the optimal classifier complexity, we must therefore agree upon which measure to use in order to detect the peak in classification performance. As previously mentioned, we are equally interested in classifying patients with either prognosis, thus the CCReq seem most interesting. When studying the relative performance of different classification methods, it is however more relevant to make this comparison with respect to the performance quantity the methods attempts to optimise, if this quantity is equal for the compared methods. For all parametric classification methods and the Parzen window classifier, this quantity is the CCReq when we use an *evened* bootstrap method, but would have been the CCR if not. This is because they are all based on the Bayes' classifier, which chooses the class that corresponds to the maximum a posteriori probability, and estimates the *a priori* probabilities using the corresponding class proportions in the learning dataset, thus weighting the two classes equally when we use an evened bootstrap method. The kNN classifier does also attempt to optimise the CCReq when we use an evened bootstrap method. This is because this classifier indirectly weights each class according to its frequency in the learning dataset. We will therefore in the following base the comparison on the CCReq, both because this is the most interesting quantity and because it is this quantity all used classification methods attempt to optimise.

The previous sections shows that the best classification methods are typically parametric, more precisely, often the NMSC or LDC (with respect to the CCReq). The complexity of any parametric classifier is, using our definition in section 6.1, the number of independent parameters in the classification method. As mentioned in section 6.2.1, this number is $cd + 1$, $0.5d(2c + d + 1)$ and $0.5cd(d + 3)$ for the NMSC, LDC and QDC, respectively. For our case of two classes, this reduces to respectively $2d + 1$, $0.5d(d + 5)$ and $d(d + 3)$. We may attempt to use these formulae to estimate the optimal number of independent parameters.

For all adaptive texture features in the sections 7.2-7.4, the best classification method with respect to the expected CCReq was the LDC, with a single unimportant exception. This indicates that the optimal classifier contains at least 7 independent parameters. As we expect that the 'allowed' number of independent parameters is larger than this, it may seem strange that the QDC with its 10 independent parameters does not perform better than the LDC. This may be explained in light of the scatter plots in all previous sections, which indicate that the patterns of good prognosis typically cluster much more than the patterns of bad prognosis. This will make the estimated variances of the QDC classifiers very different, the variance corresponding to the good prognosis class

will be much smaller than the variance corresponding to the bad prognosis class, which in turn will make the decision region of the good prognosis class relatively larger than when assuming a common variance, thus resulting in a higher CCR, but lower CCReq.

When using all five cell features, we see from section 7.1 that the LDC still performs best with respect to the CCReq. What we do not see is that the difference between this classification method and the NMSC is now much smaller than for the adaptive texture features, which indicates that even LDC is starting to become a too complex classification method. Indeed, when including also the NO-features, thus increasing the number of independent parameters with the LDC from 25 to 42, the NMSC is the best performing classification method with its 15 parameters.

In total, we see that the simple LDC classification method is generally recommendable for our dataset when using about five features or less. If more features are used, then the NMSC is the appropriate choice. Roughly speaking, about 30 independent parameters may be estimated by the classifier before it becomes overfitted. This approximate value is of course dependent on our datasets, but also on the used features, in particular their conditional pdfs and effectiveness.

Table 7.20 shows the complete classification results when using the combination of the cell features and the NO-features which attained the best expected CCReq. Notice how the classification performance significantly decreases with

Table 7.20: The classification results of the cell features and the NO-features when using the algorithm based on the watershed transformation without the edge removal step and evaluating on all 134 patients.

	NMSC	ParzenC
CCReq	70.4 % [58.0 %, 82.2 %]	67.0 % [53.8 %, 78.8 %]
CCR	70.4 % [59.0 %, 78.2 %]	68.7 % [52.6 %, 76.9 %]
Specificity	70.3 % [56.1 %, 80.3 %]	69.5 % [51.5 %, 80.3 %]
Sensitivity	70.4 % [41.7 %, 91.7 %]	64.6 % [41.7 %, 91.7 %]
	LDC	kNNC
CCReq	67.5 % [56.1 %, 78.0 %]	66.9 % [53.0 %, 79.9 %]
CCR	67.3 % [57.7 %, 75.6 %]	69.5 % [51.3 %, 79.5 %]
Specificity	67.3 % [54.5 %, 78.8 %]	70.7 % [48.5 %, 83.3 %]
Sensitivity	67.8 % [41.7 %, 91.7 %]	63.2 % [33.3 %, 91.7 %]
	QDC	NNC
CCReq	64.7 % [51.9 %, 76.9 %]	59.4 % [46.2 %, 72.0 %]
CCR	62.9 % [50.0 %, 73.1 %]	58.0 % [47.4 %, 67.9 %]
Specificity	62.1 % [47.0 %, 77.3 %]	57.3 % [45.5 %, 69.7 %]
Sensitivity	67.4 % [41.7 %, 91.7 %]	61.4 % [33.3 %, 91.7 %]

Using 28 learning patterns in each prognosis class.

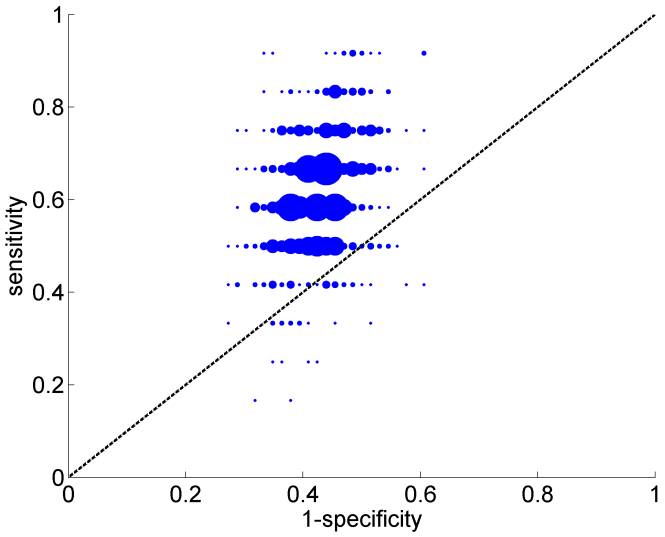

Figure 7.19: The ROC point cloud of the cell features and the NO-features when evaluating on all 134 patients and using the NNC classification method. The NO-features are computed using our watershed segmentation method without the step which removes bright edge objects.

the complexity of the classification method. When the classifier is sufficiently complex, the performance approaches randomness, as indicated by the PIs of the NN classifier and its ROC point cloud in figure 7.19. Notice also that the PIs of the CCR in the nonparametric classification methods is much larger than the corresponding PIs of the parametric methods, even after correcting for the difference in estimated expectation. Also this indicates a too large classifier complexity for the nonparametric classification methods as this uncertainty can be seen as a result of overfitting, either because of a too complex classification method (NNC) or because of the adaption of a relevant parameter (ParzenC and kNNC, respectively the window width and the number of neighbours).

It may surprise some that the Parzen window classifier and the NNC classifier performs so respectably, at least with respect to the estimated expectation, even in this case where the feature space is so sparse and simple classification methods like the LDC results in overfitting. The reason for this is the adaptive choice of window width and number of neighbours, respectively. As the feature space becomes sparser, the typical estimate of both these quantities increases significantly to allow optimal classification of the learning dataset (using the leave-one-out cross-validation method, which was our choice for optimising these parameters). The increase of these quantities results in simpler decision regions and thus also a lower classifier complexity. Therefore, these nonparametric classifiers can be said to attempt to adapt their complexity according to the optimal complexity, but, of course, this adaption is generally suboptimal.

Figure 7.20 illustrates that these estimates are indeed typically large when using the same features as in table 7.20, which we have seen is many features

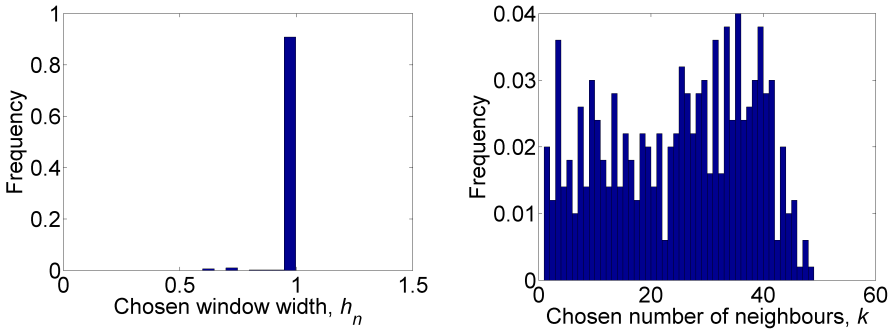

Figure 7.20: Histograms of the frequency of the chosen: left) window width when using ParzenC, right) number of neighbours when using kNNC over the 500 bootstraps when using the cell features and the NO-features and evaluating on all 134 patients. The NO-features are computed using our watershed segmentation method without the step which removes bright edge objects.

with respect to our datasets. From the left histogram we see that the chosen window width is essentially always 1. Because we have standardised the variance of each feature of the learning patterns to 1, this means that the variance of the interpolation function will in each direction be equally large as the variance of the features, thus the classification of a pattern is based on essentially all learning patterns (at least when using a normal window function as we do). Similarly, from the right histogram we see that all chosen number of neighbours up to over 40 are relatively frequent, with slightly more occurrences from about 25 to 40. As we only have 28 learning patterns in each class, this indicates that relatively many learning patterns are included to determine the class of a validation pattern, thus the classification is obviously very coarse and therefore the classifier complexity is low.

If we instead use only a few features, say only the difference CSDEMsum-features when using the same segmentation method, which was the CSDEMsum-features that obtained the best expected CCReq, we obtain the classification result in table 7.21. From these results we see that the classifier complexity do not seem to be a problem anymore. The simplest classifier, the NMSC, does now perform significantly worse than the more complexed classifiers. We however note that it is the LDC, and not e.g. the slightly more complex QDC, which performs the best. With respect to the CCReq, this is as mentioned the typical behaviour and can be seen as a result of the difference in variance between the classes. It is however not typical that the QDC attains a lower CCR, though not significant, nor it is common that the lower limit of its PI is so low relative to the same limit when using the other parametric classification method (though this limit is typically some percent lower). We will however not dwell on what causes this for this precise choice of segmentation method (and features).

We note that the Parzen window classifier and the kNN classifier still perform reasonably, but significantly worse than the best classifier. Much of the relatively small lower limits of the PIs of the CCR of these classifiers in comparison with the corresponding PIs of the parametric classifiers are also gone, but the limits

Table 7.21: The classification results of the difference CSDEMsum-features when using the algorithm based on the watershed transformation without the edge removal step and evaluating on all 134 patients.

	NMSC	ParzenC
CCReq	65.5 % [53.4 %, 77.7 %]	65.7 % [52.3 %, 78.0 %]
CCR	68.1 % [60.3 %, 75.6 %]	68.3 % [55.1 %, 75.6 %]
Specificity	69.3 % [59.1 %, 78.8 %]	69.4 % [51.5 %, 80.3 %]
Sensitivity	61.7 % [33.3 %, 91.7 %]	61.9 % [33.3 %, 83.3 %]
	LDC	kNNC
CCReq	69.2 % [56.1 %, 79.2 %]	65.2 % [50.4 %, 77.7 %]
CCR	70.6 % [61.5 %, 78.2 %]	66.8 % [51.3 %, 76.9 %]
Specificity	71.2 % [60.6 %, 80.3 %]	67.6 % [47.0 %, 81.8 %]
Sensitivity	67.2 % [41.7 %, 91.7 %]	62.7 % [33.3 %, 91.7 %]
	QDC	NNC
CCReq	67.2 % [53.8 %, 79.2 %]	60.6 % [46.6 %, 74.2 %]
CCR	70.1 % [52.6 %, 78.2 %]	59.7 % [47.4 %, 69.2 %]
Specificity	71.4 % [50.0 %, 81.8 %]	59.4 % [45.5 %, 71.2 %]
Sensitivity	63.0 % [33.3 %, 83.3 %]	61.9 % [33.3 %, 91.7 %]

Using 28 learning patterns in each prognosis class.

still seem to be significantly smaller than the corresponding limits when using the NMSC and the LDC. The likely reason is still overfitting because of the adaption of the relevant parameter, but the improved relation may be seen as a reduced risk of overfitting. This suspicion is enforced by the histograms of the estimated parameters for these classifiers when using the same features as in table 7.21, see figure 7.21. As expected, the typical chosen parameter results in the use of far less learning patterns than was the case when using the seven features, see figure 7.20. However, the suspicion was also correct as there still is a significant proportion of unnaturally small choices, see for instances the peak at $k = 1$ in the right histogram, which are estimates that are likely to be too small to result in classifiers that generalise well.

In conclusion, if CCReq is the most interesting quantity, then the LDC is the recommended classification method for our dataset when using five features or less, otherwise the NMSC is the recommended choice. The Parzen window classifier and the kNN classifier perform reasonably, also - or maybe even especially - when using many features, but both methods perform significantly worse than the best parametric method, at least with respect to the CCReq and for few features. With respect to the CCR, the QDC is the best classification method if the number of features is low, otherwise the two mentioned nonparametric methods perform well or maybe even better than its competitor, the NMSC. The NN classifier always performs badly; it is just too complex to be meaningful for our dataset. This may be seen in light of the challenges with our

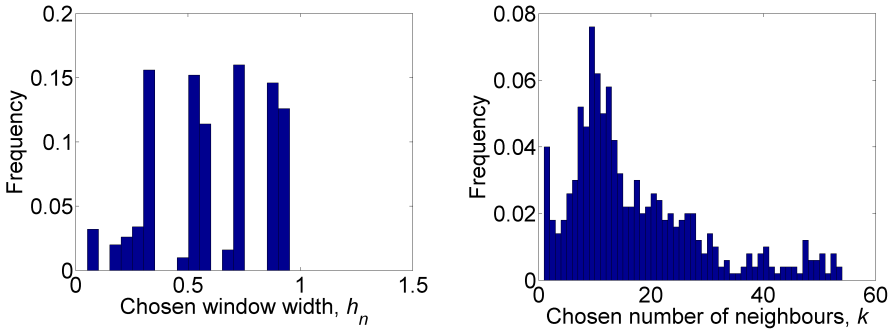

Figure 7.21: Histograms of the frequency of the chosen: left) window width when using ParzenC, right) number of neighbours when using kNNC over the 500 bootstraps when using the difference CSDEMsum-features and evaluating on all 134 patients. The difference CSDEMsum-features are computed using our watershed segmentation method without the step which removes bright edge objects.

dataset, especially the possibility of some incorrectly recorded patient outcomes (see section 2.3.3).

7.7 What if?

When we design and evaluate the classifiers, it is based on some choices other than specifying the dataset, the features, the classification method and the evaluation method. We will in this section study what would have happened if we changed some of these choices. In particular, we will begin with considering the choice of required minimum accuracy in the performance estimates of the classifier, i.e. choosing some other minimum limit than the 30 % of the total number of patterns in the same class in the validation dataset that was specified in section 6.7. This is followed by a discussion of the number of quantification levels per integer entropy in features based on spatial entropy, a choice which we in the discussion in section 3.2.4 mentioned can be seen as a trade-off between the risk of overfitting and the discrimination value of the features. The two last choices that will be considered are the use of a stratified bootstrap method instead of an evened bootstrap method, and finally the effect of using the two different estimates of the common variance when estimating the Mahalanobis distance between the classes at each element in the design of the weight arrays.

It is natural to restrict our attention to a specific relevant feature set when considering the relative performance resulting from the different choices. We will for this use the CSDEMsum-features when using the algorithm based on the watershed transformation without the edge removal step, which was the feature set that attained the best expected CCReq of all evaluated feature sets which contained only features based on the spatial entropy and when evaluating on all 134 patients. The complete classification result of these features for all 134 patients was given in table 7.21. We will, as always, evaluate using all four features in the set of adaptive texture features described in section 3.2.3, and

we will also evaluate using both all 134 patients and the 102 patients.

7.7.1 Partitioning

We will in this subsection study the effect of selecting other limits for the required minimum accuracy in the performance estimates of the classifier. For all previous evaluations, this limit has been set to 30 % of the total number of patterns in the same class in the validation dataset. All evaluations mentioned in this chapter have however reached this limit. In particular, this means that the used partitioning is not a direct application of our estimate of the 'optimal' partitioning.

We have evaluated two other choices of the limit, 10 % and 50 %. Of the four adaptive feature types, it was the difference CSDEMsum-features which obtained the best expected CCReq with respect to all 134 patients, while the negative CSDEMsum-features which obtained the best expected CCReq and best expected CCR with respect to the 102 patients. Table 7.22 shows the classification results with these features.

In comparison with the classification results when using all 134 patients in table 7.21, we see that the expected CCReq and expected CCR increases by a good percent when choosing the limit of 10 %, and decreases by about two percents when choosing the limit of 50 %. After correcting for different expectations, we see that the uncertainty of each PI limit of the CCReq increases with nearly 10 % when choosing the limit of 10 %, but only shrinks by a little

Table 7.22: The classification results of the difference CSDEMsum-features when using all 134 patients and the negative CSDEMsum-features when using the 102 patients. The LDC is used as the classification method because it attained the best expected CCReq and best expected CCR in all cases. All CSDEMsum-features are computed using the algorithm based on the watershed transform without the edge removal step.

	All 134 patients Limit = 10 %	The 102 patients Limit = 10 %
CCReq	70.5 % [47.8 %, 88.8 %]	77.2 % [53.7 %, 97.1 %]
CCR	71.8 % [62.9 %, 79.0 %]	86.4 % [78.9 %, 94.7 %]
Specificity	72.0 % [62.1 %, 81.0 %]	88.9 % [82.4 %, 97.1 %]
Sensitivity	69.0 % [25.0 %, 100.0 %]	65.5 % [25.0 %, 100.0 %]
	All 134 patients Limit = 50 %	The 102 patients Limit = 50 %
CCReq	67.0 % [53.6 %, 75.8 %]	76.0 % [65.3 %, 85.4 %]
CCR	68.2 % [53.2 %, 75.5 %]	81.4 % [72.7 %, 87.9 %]
Specificity	69.1 % [48.6 %, 79.7 %]	87.8 % [79.2 %, 93.8 %]
Sensitivity	64.8 % [40.0 %, 85.0 %]	64.2 % [44.4 %, 83.3 %]

Using 36 (upper left), 32 (upper right), 20 (lower left) and 18 (lower right) learning patterns in each prognosis class.

percent when choosing the limit of 50 %, both in comparison with choosing the limit of 30 %. The PIs of the CCR are however relatively similar after correcting for different expectations. This is a result of the use of an *evened* bootstrap method and uneven datasets, as the number of validation patterns with bad prognosis is relatively much more affected by the changed limit than the number of validation patterns with good prognosis, which is reasonable for all three chosen limits (minimum 34). In total, the limit of 30 % seems reasonable with respect to all 134 patients as it provides relatively reliable estimates while obtaining relatively good expected performances.

Comparing the performance when using the 102 patients in table 7.22 with the right result in table 7.15, we see that the expected CCReq does not change significantly when using either 10 % or 50 % in comparison with 30 %. The PI limits when choosing the limit of 50 % has again increased with nearly 10 % each, and this time we also note a significantly shrunken PI length of 6 % when using the limit of 50 % in comparison with 30 %. The story is however oppositely when looking at the CCR; the expected values are then altered by good two percents for both cases, but none of the PI lengths are significantly different. The total effect is however similar as with all 134 patients; the PIs of the CCReq with the limit of 10 % are ridiculously high and the expected performance with the limit of 50 % is generally significantly lower than with the other limits. We therefore conclude that the limit of 30 % also seems reasonable with respect to the 102 patients.

The effects observed in this subsection are generally representative for other feature sets and classification methods. In particular, the increased uncertainty when only using a small portion of the patterns for validation will always be evident. However, the relative difference in the expected performance will vary between classifiers, mainly dependent on how close each of them are to being overfitted. This is a consequence of the discussed phenomenon of overfitting (see section 6.3), as the number of learning patterns is far more essential for overfitted or nearly overfitted classifiers than if a classifier's risk of overfitting is low.

7.7.2 Quantification

We will in this subsection study the effect of coarse and precise quantification for the CSDEMsum-features. The quantification of these features has in this study been measured by the number of quantification levels per integer entropy, which was set to 5.

We have compared the classification results of the CSDEMsum-features for the choices of 1, 2, 10 and 25 quantification levels per integer entropy. The relative effect of the different quantifications was similar when using either dataset, but most prominent when using all 134 patients. We have thus chosen to only present the classification results when using all 134 patients in this subsection. With respect to this dataset, it was generally the difference adaptive texture feature which attained the best expected CCReq. As the typical effect of the different quantifications is also evident for this feature, we will limited the presentation to this features.

The classification results of the difference CSDEMsum-features when using all 134 patients are shown in table 7.23. These results are significant and easy to interpret. If the quantification is too coarse, values which indicates differ-

Table 7.23: The classification results of the difference CSDEMsum-features when using the algorithm based on the watershed transformation without the edge removal step and evaluating on all 134 patients. The LDC is used as the classification method because it attained the best expected CCReq and best expected CCR in all cases. The corresponding classification result with $q_G = q_V = 5$ is given in the right of table 7.14 (and table 7.21 gives the classification results with all six classification methods with $q_G = q_V = 5$).

	$q_G = q_V = 1$	$q_G = q_V = 2$
CCReq	63.0 % [49.2 %, 76.9 %]	69.6 % [57.2 %, 79.5 %]
CCR	71.4 % [59.0 %, 79.5 %]	70.6 % [60.3 %, 78.2 %]
Specificity	75.2 % [59.1 %, 84.8 %]	71.0 % [57.6 %, 81.8 %]
Sensitivity	50.8 % [25.0 %, 83.3 %]	68.2 % [41.7 %, 91.7 %]
	$q_G = q_V = 10$	$q_G = q_V = 25$
CCReq	68.3 % [56.1 %, 79.5 %]	64.9 % [51.5 %, 75.8 %]
CCR	69.8 % [60.3 %, 76.9 %]	67.9 % [60.3 %, 74.4 %]
Specificity	70.5 % [57.6 %, 80.3 %]	69.2 % [57.6 %, 78.8 %]
Sensitivity	66.1 % [41.7 %, 91.7 %]	60.7 % [33.3 %, 83.3 %]

Using 28 learning patterns in each prognosis class.

ence prognosis will be located in the same element in the property arrays, thus this discrimination value is lost. With reference to the weight arrays of these CSDEMsum-features shown in the lower row in figure 7.14, we see that assigning all values from 8 to 9 to a single element, as setting $q_G = q_V = 1$ results in, will indeed mix values which are estimated to indicate difference prognosis when using five elements in the same range. The significantly decreased performance in the upper left corner of table 7.23 verifies this. Oppositely, if the quantification is too precise, we risk that the designed weight arrays are overfitted to the learning datasets because there are too few occurrences within each element. The weight arrays in figure 7.22 clearly shows that this is the case when designed using 25 quantification levels per integer entropy. The corresponding result in the lower right corner of table 7.23 verifies this, and the result when using 10 quantification levels per integer entropy also shows signs of overfitting.

In comparison with the classification results in table 7.21, we see that the use of 2 quantification levels per integer entropy may be slightly better than the used number of 5. The reason why the designed weight arrays in the lower row in figure 7.14 do not indicate this, may be that these weight arrays are designed using the entire relevant dataset, while the bootstrapped learning datasets will be significantly smaller (about half of all relevant patterns when the 30 % limit is used and reached). From the results with these two quantifications and the results when using 1 and 10 quantification levels per integer entropy, we expect that the optimal number of quantification levels per integer entropy is between 2 and 5 for our CSDEMsum-features for our datasets, and likely closer to 2 than 5. However, we have not and will not search for this number as that would have been to use the entire dataset extensively for learning, thus likely to

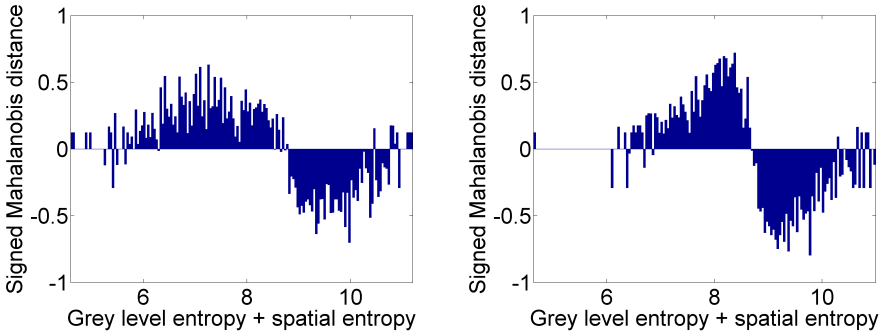

Figure 7.22: The designed weight arrays of the difference: left) CSDEMsumDark-feature, right) CSDEMsumBright-feature when using 25 quantification levels per integer entropy and the algorithm based on the watershed transform without the edge removal step and evaluating on all 134 patients.

result in overfitting. Instead, we will rely on our previous analysis of the weight arrays in figure 7.14, which indicates that these weight arrays are the result of a reasonable quantification, and simply note that the results of this subsection indicate that this choice does indeed seem to be good.

We noted in section 4.2 that the features based on the CSDEM are especially subject to overfitting due to relatively few occurrences when using this property array compared to the standard property arrays. The results of this subsection are still representative, the only difference is that a reasonable quantification for the standard property arrays will be much more precise, but the effect of significantly decreasing or increasing the precision would be similar to that observed in this subsection. We would also like to point out that our CSDEMsum-features have only a single dimension each, thus the relative effect of changing our quantification quantity, the number of levels per integer entropy, is much smaller than when performing the same relative change along each axis of a multidimensional property array, as for instance the CSDEM-features.

7.7.3 Stratified bootstraps

We will in this subsection compare the use of the stratified bootstrap method with the evaluation method used thus far in this chapter; the evened bootstrap method. If otherwise applying our standard evaluation scheme, the use of a stratified bootstrap method will include more learning patterns than the corresponding evened bootstrap method, and the classification methods will also, directly or indirectly, attempt to optimise the CCR and not the CCReq. The first difference is a result of the reached limit of at least 30 % validation patterns, which makes the evened bootstrap method use 30 % of the patterns in the prognosis class with least patterns for validation. The stratified bootstrap method will on the other hand use 30 % of the patterns in each prognosis class for validation, thus using fewer validation patterns in the prognosis class with most patterns and therefore more learning patterns in this prognosis class. The second difference is a consequence of estimating the *a priori* probabilities as the corresponding class proportions. This difference will cause the classifiers evalu-

ated using the stratified bootstrap method to favour classifications to the good prognosis, thus violating the mentioned equal interest of correctly classifying the good and bad prognosis patients. Because of these possibly undesired differences, we will in this subsection compare the evened bootstrap method with four evaluation schemes which apply the stratified bootstrap method, where the three non-standard schemes are obtained by setting the *a priori* probabilities to 0.5 and/or requiring equal number of learning patterns as in the corresponding evened bootstrap method.

When requiring equal number of learning patterns as in the corresponding evened bootstrap method, we would have to specify the number of learning patterns in each class so that the option of stratified is also fulfilled. With respect to all 134 patients, the number of learning patterns has (with the limit of 30 %) always been 56. The number of learning patterns with bad prognosis should thus be $56 * 40/134 \approx 16.72 \approx 17$ in this case, making the number of learning patterns with good prognosis 39. Similarly, the number of learning patterns has (with the limit of 30 %) always been 50 with respect to the 102 patients. The number of learning patterns with bad prognosis should thus be $50 * 36/102 \approx 17.65 \approx 18$ in this case, making the number of learning patterns with good prognosis 32.

We have evaluated the four evaluation schemes which apply the stratified bootstrap method for both datasets and using either of the four adaptive texture features described in 3.2.3, when the common variance is estimated as the standard pooled variance estimate (see equation (3.8)). The relative performance is similar for both datasets, but more prominent for the dataset with all 134 patients. The following presentation will therefore be restricted to the dataset with all 134 patients. For this dataset, it is again the difference CSDEMsum-features which attains the best CCReq, and since the typical effect is also prominent for these features, we will in the following only present the results with them.

Table 7.24 shows the classification results of the difference CSDEMsum-features when using all 134 patients. Comparing the upper row with the lower row or the results when using the standard scheme with the evened bootstrap method in table 7.21, we see that letting the *a priori* probabilities be estimated by the corresponding class proportions results in a highly significant decrease of the expected CCReq when using the stratified bootstrap method. This is not unexpected as the classification methods will now attempt to optimise the CCR. It is therefore interesting to note that the expected CCR does in fact increases with this evaluation scheme, perhaps also significantly[9]. However, the PI of the CCReqs indicates that the classifiers of the upper row in table 7.24 are on the verge of randomness with an expected CCR which only reflects the corresponding class proportions (which is approximately 70 %). This observation is enforced by the ROC point cloud of the best of these two classifier, the one which uses the standard scheme, which is shown in figure 7.23. From this ROC point cloud we see that the performance is only slightly better than random, in fact, we may even claim that this difference is not sufficient to be called significant.

The results are much better when we set the *a priori* probabilities to 0.5, see

[9]It is questionable to call the difference between 71.9 % or 71.7 % and 70.6 % significant, but the difference may be just significant if we correct for fewer number of validation patterns with good prognosis when using the stratified bootstrap method in comparison with the evened bootstrap method.

Table 7.24: The classification results of the difference CSDEMsum-features when using the algorithm based on the watershed transformation without the edge removal step and evaluating on all 134 patients. The LDC is used as the classification method because it attained the best expected CCReq and best expected CCR in all cases. ENOLP *is used as the shorthand for equal number of learning patterns as in the corresponding evened bootstrap method.*

	Standard scheme	ENOLP
CCReq	61.4 % [50.0 %, 72.0 %]	60.9 % [50.3 %, 69.5 %]
CCR	71.9 % [62.5 %, 80.0 %]	71.7 % [65.4 %, 76.9 %]
Specificity	87.6 % [71.4 %, 100.0 %]	87.2 % [74.5 %, 98.2 %]
Sensitivity	35.3 % [16.7 %, 58.3 %]	34.7 % [4.3 %, 56.5 %]
	Equal prior	Equal prior & ENOLP
CCReq	71.0 % [56.5 %, 82.7 %]	68.0 % [55.7 %, 76.0 %]
CCR	71.8 % [60.0 %, 82.5 %]	69.3 % [56.4 %, 76.9 %]
Specificity	73.2 % [60.7 %, 85.7 %]	71.2 % [56.4 %, 83.6 %]
Sensitivity	68.8 % [41.7 %, 91.7 %]	64.8 % [39.1 %, 82.6 %]

Using 66 and 28 (left) or 39 and 17 (right) learning patterns with good and bad prognosis, respectively.

the lower row in table 7.24. This choice of *a priori* probabilities will effectively make all parametric classification methods and the Parzen window classifier optimise the CCReq, even for uneven learning datasets. We note that the kNN and NN classifiers will still optimise the CCR as these classification methods do not use the *a priori* probabilities, but instead indirectly use the corresponding class proportions through its decision rule.

Comparing the result when setting the *a priori* probabilities to 0.5 and using equal number of learning patterns as in the corresponding evened bootstrap method, lower right corner in table 7.24, with the results with the evened bootstrap method (see table 7.21), we see that the evened bootstrap method obtains a good percent better expected CCReq and expected CCR. However, the length of the PI of the CCReq has decreased from 23.1 % to 20.3 %. This indicates that the patients with bad prognosis are more relevant than the patients with good prognosis for both designing and evaluating the classifier. We however note that the length of the PI of the CCR has actually increased from 16.7 % to 20.5 %, which is odd as the CCR weights all patterns equally, thus we expected its uncertainty to be approximately equal because the there are equal number of validation patterns. In total, we conclude that the use of an evened bootstrap method seem slightly better than using the stratified bootstrap method which uses the same number of learning patterns (with or without setting the *a priori* probabilities to 0.5).

If we consider the stratified bootstrap method when setting the *a priori* probabilities to 0.5, but not restricting the number of learning patterns to be equal to the corresponding evened bootstrap method, we note an increase in expected CCReq and expected CCR of between one and two percents. This improvement

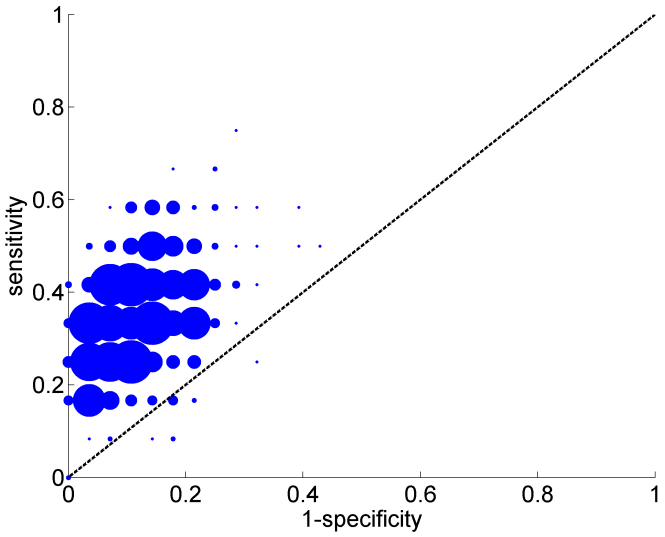

Figure 7.23: The ROC point cloud of the CSDEMsum-features when evaluating on all 134 patients with the standard scheme of the stratified bootstrap method and using the LDC classification method. The CSDEMsum-features are computed using our watershed segmentation method without the step which removes bright edge objects.

is insignificantly different from what we observed when setting the limit to 10 %, see the upper left corner of table 7.22. However, the PIs of this stratified bootstrap method is not ridiculous. Comparing with the results of the evened bootstrap method when letting the limit be 30 %, we note a moderate increase in the length of the PI of the CCReq from 23.1 % to 26.2, while the corresponding length of the CCR increased from 16.7 % to 22.5 %. Both the increased expected performance and increased uncertainties are natural consequences of the use of more learning patterns and less validation patterns, respectively, and the reasonable relation between the increases makes it too difficult to name one of the evaluation schemes better than the other in general. We therefore conclude that this stratified bootstrap method is not bad in comparison with the used evened bootstrap method, but which evaluation scheme is most appropriate will depend on how one weights the differences.

In total, it is only a single evaluation scheme which applies the stratified bootstrap method that could be recommended when the CCReq is at least of some interest, which is the scheme that sets the *a priori* probabilities to be equal and do not restrict the number of learning patterns to be equal to the corresponding evened bootstrap method. Despite the fact that this conclusion is made when evaluating with only a limited set of feature sets, we see not reason why this should not be representative in general. Furthermore, we expect similar relative results for the holdout-validation method depending on how the learning dataset is selected (i.e. to contain equal or unequal number of patterns in each class) and how an unequal number of learning patterns in each class is treated

(i.e. setting the *a priori* probabilities equal or not).

7.7.4 The effect of using the two different estimates of the common variance

We will in this subsection discuss the effect of using the two different estimates of the common variance when estimating the Mahalanobis distance between the classes at each element in the design of the weight arrays. The two estimates are the average of the individual variance estimates (see equation (3.7)) and the standard pooled variance estimate (see equation (3.8)).

We mentioned in the previous subsection that we estimated the common variance as the standard pooled variance estimate. The reason why this has not been specified sooner is because these estimates will be identical for our features when using the evened bootstrap method. This is however not true for all features when using arbitrary datasets. For instance, if grouping on cell areas (as we do in the GLEM-features), some area groups in some patients may have no cell images. Such patients are natural to ignore when computing the estimated expectation and variance which is used in the Mahalanobis distance between the classes. The number of patients within a specific area group may thus differ, which makes it possible that the two estimates of the common variance differs and thus that the resulting classifiers perform differently even in the case of an *evened* bootstrap method. However, when using well populated area groups, most patients have some cell images within each area group, thus the difference between the approaches will be negligible in such situations. Because of our great concern for the overfitting problem, the only features in this study which uses area grouping, the GLEM-features, use well populated area groups. In fact, every patient have some cell images with each of the used area groups, thus the two approaches are identical for our GLEM-features. A similar comment could be applied if separating on segmentation classes where all cell images of some patients may have no pixels of a specific segmentation class. However, our segmentation methods will for our dataset result in no such patients, thus the approaches will be identical also for our adaptive texture features which depends on a segmentation.

To detect any potential difference between the use of the two different estimates of the common variance, we must thus use the stratified bootstrap method. From the discussion in the previous subsection, we see that we should set the *a priori* probabilities equal to 0.5 in order to obtain reasonable classifiers. The resulting evaluation schemes are evaluated with the CSDEMsum-features for the four adaptive feature types described in section 3.2.3 and using both datasets. Again we find similar results for both datasets and have chosen to restrict the following presentation to the dataset with all 134 patients. Also this time it is the difference CSDEMsum-features that attains the best CCReqs, thus we will also limit the presentation to these features.

Comparing the classification results when estimating the common variance as the average of the individual variance estimates in table 7.25 with the result of using the standard pooled variance estimate in table 7.24, we see that the expected performance of both CCReq and CCR decreases with 0.2-0.6 %, which is not significant. It is not unexpected that these performances are similar. This is because we see from figure 7.14 that the assumption of common variance is often valid for the most essential elements in the property arrays of these

Table 7.25: The classification results of the difference CSDEMsum-features when estimating the common variance as the average of the individual variance estimates and using the algorithm based on the watershed transformation without the edge removal step and evaluating on all 134 patients. The LDC is used as the classification method because it attained the best expected CCReq, best expected CCR and best expected sensitivity in both cases.

	Equal prior	Equal prior & ENOLP
CCReq	70.4 % [57.1 %, 83.3 %]	68.3 % [56.1 %, 76.0 %]
CCR	71.3 % [60.0 %, 82.5 %]	69.5 % [57.7 %, 76.9 %]
Specificity	72.6 % [57.1 %, 85.7 %]	71.3 % [56.4 %, 83.6 %]
Sensitivity	68.3 % [41.7 %, 91.7 %]	65.3 % [43.5 %, 78.3 %]

Using 66 and 28 (left) or 39 and 17 (right) learning patterns with good and bad prognosis, respectively.

CSDEMsum-features. The fact that the F-test is strongly dependent on the normality assumption is unimportant for this relation, as the statistic of the F-test is precisely the ratio of the individual variances and these variances are approximately equal whenever the test is not rejected[10]. In fact, the F-test at significance level 0.05 is so precise relative to indicating significant differences between the two variance estimates that we may expect approximately similar estimated discrimination values also in elements where the F-test is rejected. It is thus only natural that the relative classification results of applying either of these variance estimates are insignificant different for the CSDEMsum-features.

The observations of this subsection are too dependent on the property array to be called generally representative. However, when using the average of the property arrays of the cell images as the patient's property array and we are in a situation where all patients are likely to have some cells indicating either prognosis, the individual variance estimates of each element is likely to be approximately equal if the property arrays are rather densely filled with occurrences. In this case, the observations of this subsection are likely to be representative, i.e. the use of either of the two estimates of the common variance is likely to give approximately the same results.

7.8 Comparison with DNA ploidy analysis

A short comparison of our best image analysis approaches with the convincing results of DNA ploidy analysis is in order. We see from table 3.1 in section

[10]This is in general a simplification because the ratio may be very large or small without resulting in a reject if at least one of the degrees of freedom is very small. For the F-test in figure 7.14, the degrees of freedom are 93 and 39 when using all 134 patients, and 65 and 35 when using the 102 patient, thus the 95 % two-sided CIs of the statistic in the F-test are $[0.60, 1.76]$ for all 134 patients and $[0.57, 1.85]$ for the 102 patients. This makes the maximum difference between the two variance estimates about 10 % when the F-test is not rejected (and the proportions in the two sets of realisations are about 70 % and 30 %, as they are in our datasets).

3.1.2 that the DNA ploidy analysis described in this section obtained a CCReq of 82.6 % and a CCR of 83.6 % when using all 134 patients. This is highly significantly better than the best approaches using image analysis, which attained comparatively humble expectations of 70.8 % and 72.6 % in CCReq and CCR, respectively.

The relative performance is however much better when excluding the patients with tetraploid and polyploid histogram. From table 3.2, we see that the mentioned DNA ploidy analysis now obtains a CCReq of 84.8 % and a CCR of 85.3 %. Though this is significantly better than when including all patients, it is a minor increase relative to the about 10 % increased expected performance observed with the best image analysis approaches, which obtained an expected CCReq of 79.8 % and a expected CCR of 83.9 %, notably with different feature sets.

In total, even though we have introduced novel features which performs very well on our dataset, the mentioned DNA ploidy analysis is still superior to the best image analysis methods for this dataset. Because the DNA ploidy analysis exploits the biomedical understanding of carcinogenesis, while our proposed image analysis approaches perform a more general analysis of the internal structure of the DNA-specific stained nuclei, we are still pleased with the results of our approaches in light of the performance of all other image analysis methods. This satisfaction is enforced by the fact that the difference in expected CCR between our novel image analysis approach and the mentioned DNA ploidy analysis is about equal for the patients with diploid or aneuploid histogram.

7.9 Summary

We have in this chapter studies the proposed segmentation methods. We found that the removal of estimated bright primitives sufficiently close to the edge of the nucleus had a negative effect on the performance of the features based on the CSDEMs. Both separation algorithms were however equally good with respect to the CSDEMs, and they also resulted in good NO-features which combined well with the cell features; the combined performance was in particular significantly better than when using the segmentation method used in [49].

We have studies the performance of the features based on the CSDEMs. The CSDEM-features performed reasonable in themselves, but showed signs of being overfitted. This was dealt with by using their sum histograms instead, resulting in the CSDEMsum-features which performed significantly better with respect to the CCReq and all 134 patients. The best CSDEMsum-features where typically based on the watershed transform algorithm and the classification method which assumes normality and equal covariance matrices (LDC). The best expected performance estimates of these features (with the standard evaluation scheme) were a CCReq of about 69 % and an CCR of about 71 % with respect to all patients, while about 77 % in CCReq and about 84 % in CCR with respect to only the patients with diploid or aneuploid histograms.

The only tested features which seemed reasonable to combine the CSDEMsum-features with were the geometrical features. Any combination with these features did not significantly increase the performance with respect to only the patients with diploid or aneuploid histogram, but the combination of the CSDEMsumBright-feature and the Eccentricity-feature increase the expected CCReq with about 2

% when using the LDC and instead increased the expected CCR with about 2 % when using the Parzen window classifier, both with respect to all patients. The results of this combination are very good with respect to all patients. The expected performance of about 71 % in both CCReq and CCR when using the LDC is in fact the best overall performance of any feature set with respect to all patients, but it is not significantly better than e.g. the combination of the cell features and the best NO-features.

In comparison with the best features based on the GLEM, the GLEM4D-features, we have seen that the CSDEMsum-features is generally significantly better. As the GLEM is the most promising choice of property array for our dataset, it is exhilarating to note that our proposed choice of property arrays, the CSDEMs, results in features which are individually significantly better. If combining the GLEM4D-features with the geometrical features, which yet again were the only tested features which seemed reasonable to combine the GLEM4D-features with, the best combination obtained significant better expected CCReq with respect to the 102 patients in comparison with the combination of the CSDEMsumBright-feature and the Eccentricity-feature, but insignificant different with respect to all 134 patients and the expected CCR when using either dataset. We therefore note that the GLEM4D-features are generally better combined with the other tested features in this study than the CSDEMsum-features, but this is only if we exclude the patients with tetraploid or polyploid histogram. If including all patients, we can again note that our CSDEMsum-features are better, though not significantly.

For all features based on the CSDEMs, the difference adaptive texture feature was generally the best performing features among the set of four features described in section 3.2.3 with respect to all patients. The negative adaptive texture feature was generally best with respect to the 102 patients, or even with all patients for the features based on the GLEM. We have noted that it is interesting that the negative adaptive texture feature is often performing better than the difference feature, and claimed that this may be caused by the existence of several normal cells within even the most essential part of the tumour, as was mentioned in section 2.3.2.

With respect to the classification method, we have seen that the parametric classification methods are generally best with respect to our dataset. In particular, the LDC can be recommended when using five features and less, otherwise the NMSC is the recommended classification method. The appropriateness of applying simple classification methods may be seen in light of the challenges with our dataset, especially the possibility of some incorrectly recorded patient outcomes (see section 2.3.3). We have however also seen that the two nonparametric classification methods which adapt an essential parameter by evaluating on the learning dataset are also generally reasonable. It was in particular interesting to note that the typical resulting complexity when using these nonparametric classification methods was adapted according to the risk of overfitting, though in a suboptimal fashion, resulting in an acceptable performance even in the case of relatively many features.

In total, we have seen that two of our proposed segmentation methods are generally very good, at least with respect to the NO-features and features based on the CSDEMs. We have recorded good performance of the CSDEM-features and in particular the features from their sum histograms. We have also seen that these performance estimates could have been increased by applying other eval-

uation schemes. In particular, the application of a stratified bootstrap method where the *a priori* probabilities are set to 0.5 did result in significantly increased performance estimates without making the corresponding PIs ridiculously large. In either case, we are pleased with the generally promising classification results from the use of property arrays which are based on the spatial entropy, where the object size is used as the contextual measurement and the segmentation is based on one of two promising segmentation methods which we have proposed. This satisfaction is also not overshadowed by the fact that our proposed approaches are, as all evaluated image analysis approaches, subordinate to the approach based on DNA ploidy analysis.

Chapter 8

Conclusion

The main aim of this study was to develop an automatic algorithm that reliably estimates the prognosis of novel patients with early ovarian cancer. In opposed to traditional approaches based on statistical texture analysis, the prognosis estimation in this study was based on exploiting the internal structure of DNA-specific stained nuclei by applying a novel texture analysis concept coined the *class specific dual entropy matrix (CSDEM)*. The computation of the CSDEM was based on a novel, refined adaptive segmentation method to extract small dark and bright structures within the nuclei. The segmentation method included modifications of Niblack's adaptive segmentation method and the validation step of Yanowitz and Bruckstein's segmentation method, as well as either morphology or the watershed transformation. The area of the segmented objects were used to estimate a spatial entropy of the dark or bright structures of each nucleus, and combined with the estimated grey level entropy within the same segments to obtain an element in the CSDEM. Finally, we used the CSDEM to obtain some very few, but powerful novel adaptive texture features by adaptively estimating the discrimination value of each of its elements by using the combined knowledge of all relevant CSDEMs of all nuclei across a number patients.

We applied a proper evaluation method based on statistical bootstrapping to estimate the performance of our novel adaptive texture features. By using Fisher's linear discriminant in combination with a threshold based on a normality assumption, we obtained an average of specificity and sensitivity of nearly 70 % with respect to a dataset which contained 134 patients. This is significantly better than what was obtained with the previously most promising method based on texture analysis and at least about equally good as all other approaches based on image analysis. Combining the best of our novel adaptive texture features with a single other feature, we obtain an average of specificity and sensitivity of 71 % with the just mentioned classification method, and a correct classification rate of 73 % when using the Parzen window classifier. Both these performances are the best we have obtained among all feature sets based on image analysis.

We have seen that DNA ploidy analysis is a method unrelated to digital image analysis that can be used to group the patients into two subsets. It has been indicated that many relevant properties are opposite for patients in these two groups with respect to the true prognosis. When evaluating using one of the subsets, we obtained a correct classification rate of 84 % with the

mentioned classification method based on Fisher's linear discriminant and a normality assumption. This performance is the best we have obtained among all feature sets based on image analysis, perhaps also significantly better than all other feature sets. Moreover, it was also shown that the uncertainty of this estimate is relatively low.

The good performance of our novel adaptive texture features when separating using DNA ploidy analysis facilitate to a two-step recognition system. Unfortunately, the low number of patients in the complimentary subset prevents us from reasonably evaluating the performance when using this subset. We can therefore not validate the good performance of the two-step recognition system for novel patients in general, but we can nevertheless postulate that the performance would have been similar on the complimentary subset and therefore valid for all patients.

We have proposed novel, adaptive segmentation methods where at least two of them have been shown to perform reasonably and in particular equally well or better than the other tested segmentation methods. We have proposed a novel texture analysis concept, the CSDEM, which resulted in features that are significantly better than the previously most promising features based on texture analysis and also all evaluated feature sets based on image analysis. The classification results are generally very good, especially in light of how few features contributes to the promising results. Also in light of the use of proper performance estimation, we expect that our approach will generalise well on an independent validation dataset. Moreover, because of the combination of high adaptivity in all stages of our approach and an addressed concern for the overfitting problem, we expect relatively good generalisation beyond the case under study. Our novel approach thus seems to hold a promise of reliable estimation of the prognosis, which is necessary to make a qualified selection of the appropriate adjuvant treatment. Nevertheless, caution must be called for, especially because our approach has not yet been evaluated on an independent validation dataset, and new proper tests must as always be performed in the case of generalisations.

Chapter 9

Further work

- Evaluate the performance of the CSDEMsum-features when using an independent validation dataset.

- Apply survival analysis to both investigate the estimated performance when using this method (which in particular could include patients who died of causes unrelated to ovarian cancer) and gain better insight of our classifiers by relating the estimated prognosis to the time since the (last relevant) surgery.

- We have observed highly significant increased performance when evaluating only on the patients with diploid or aneuploid histogram, but were unable to reasonably evaluate the performance on the complementary subset of patients because of very few patients with bad prognosis. Instead of separating on ploidy type, we could have included the ploidy type as a (discrete) feature and evaluated the performance of this feature combined with one or two of our features based on the CSDEM. The performance of such classifiers would indeed have been interesting to investigate.

- All of our classification results are based on averaging the property estimates of each cell images of a particular patient to obtain the corresponding property estimate of the corresponding patient, where each property estimate is either a feature value or a property array. As indicated in section 2.3.2, this approach is likely to be suboptimal in general. One should therefore investigate the prognostic value of using other characteristics and particularly of using specific subset(s) of the cell images of each patient to obtain the property estimates of the patients.

- We noted in the discussion in section 3.2.4 that the use of adaptive texture features versus predefined texture features may boil down to a choice between more discrimination value in each element of the property arrays versus more precise property arrays. It would as an extension to this be interesting to investigate how the classification performance of predefined texture features are influences by the chosen quantification, and also to compare the results of some reasonable predefined texture features to the results of some adaptive texture features for the same choice of property

array(s), but using individually appropriate quantifications. Related comparisons are also of interest, e.g. the effect of different parameter choices for both predefined and adaptive texture features, and their combinations.

- A set of adaptive texture features can be obtained by basing the design of the weight array on all scenes in the learning dataset. Section 3.2.3 described one such feature set, which is based on the Mahalanobis distance between the classes. We here argued that the difference adaptive texture feature is likely to be the generally best performing feature of this feature set. We also mentioned that the estimated discrimination value of each element of the property array is in this case the standard T-statistic used in pooled two-sample t-tests under the null hypothesis of equal expectations. This provides us with the idea that the statistic of other two-sample tests may also be appropriate to estimate the discrimination value of the elements in a property array. In particular, we may relax the assumption of common variance by using the standard T-statistic used in two-sample t-tests. If the number of learning patterns is small or the normality assumption is inappropriate, we could also use the statistic of some non-parametric hypothesis test, e.g. the Mann-Whitney test [36] (also called [11, pp.752–755] the Wilcoxon rank-sum test [71]), or the Kruskal-Wallis test [30] if there are more than two classes.

  The application of the statistic in other hypothesis tests than the pooled two-sample t-test, or the application of some other estimate of the discrimination value of each element in a property array like for instance an estimate of the *Bhattacharyya distance* between the classes, have generally a couple of other positive consequences which are worth mentioning. First of all, each method for estimating the discrimination value of an element in a property array will also lead to a criterion function that could be used for feature selection[1]. Secondly, if we relax the normality assumption, we will no longer need the justification of this assumption that is provided by the central limit theorem when using the average (or sum) of the cell property arrays as the patient property array. Thus the relaxation of the normality assumption will also make it more reasonable to use other characteristics of the cell property arrays and to use highly specific subset(s) of the cells within a patient.

  Methods of designing the weight array which also include an inter-element

[1] The set of realisations in a specific element in the property arrays of the learning patterns will always be one-dimensional. In feature selection, we need to be able to compare multiple features. To apply the method for estimating the discrimination value of an element in a property array as a criterion function for feature selection, we must therefore either define the criterion function in terms of the individual contributions of each feature in each of the compared sets (e.g. the sum) or use a generalisation of the method to multiple dimensions (this is preferable). We should also mention that this relationship often also goes the other way, i.e. a criterion function for feature selection could lead to a method for estimating the discrimination value of an element in a property array. The direct application of any criterion function for feature selection is however not always appropriate as the precise value of the criterion function is not of interest for feature selection, only their order, but their relative values are of importance when used as a method for estimating the discrimination value of an element in a property array. In particular, an estimate of the mutual information between the assumed true underlying distribution of a specific element of a property array and the discrete random variable giving the true *a priori* probabilities could be directly used as an estimate of the discrimination value of that element.

analysis should also be investigated. In particular, methods which generalise the estimated discrimination value of each element to a rougher description of the entire weight array could be of major importance, especially if the element estimates are unreliable in themselves. One such approach is to fit the initially designed weight array to a surface. A more adapted approach may be to also increase the estimated discrimination value of elements where few property arrays are nonzero, where the level of increment should be determined on the basis of the entire structure of the initially designed weight array. The advantage with such increments can be illustrated by considering a one-dimensional property array where the true probability of occurrence of either class is equally distributed with the exception of the expectation, which is e.g. -1 and 1. If we in the property array of a novel pattern observe high *and* unlikely values, then the element-based estimated discrimination values of these elements are likely to be zero because the values did not occur in the property arrays of the learning patterns. However, the presence of such values is likely to strongly indicate that the true class is the one with expectation 1.

An alternative or additional method for reducing the risk of overfitting and to reasonable use infrequent (and therefore often unreliable) elements of the property arrays is to allow the quantification steps to vary depending on how reliable the resulting elements can be expected to be. One such approach is to use the *Lloyd-Max quantiser* which determines the length of the quantification steps depending on the number of occurrences within the resulting elements[2]. The application of such methods could also lead to better discrimination in densely filled regions because these regions are likely to be divided into more elements (than with a fixed quantification) while maintaining enough occurrences to ensure reliable estimation of their individual discrimination value.

- The Lloyd-Max quantiser has other natural applications in our context. Firstly, it could be used to reduce the number of grey levels in the cell images. Secondly, it could be used with the features that apply area grouping (which the GLEM-features did) to both include cell images with infrequent areas without resulting in unreliable estimates, and to allow better discrimination within the well populated area groups. In both cases, it would be interesting to compare the classification results of such application with the choices made in this study (linear scaling and three fixed area groups, respectively) and other choices.

  For these uses of the Lloyd-Max quantiser, as well as for the use mentioned in the item above, it is likely best to compute the quantification based on the entire set of learning patterns. The other natural options are to compute the quantification for each patient or cell image. Such usage has two drawbacks. Firstly, it will make the interpretation of the resulting elements dependent on the patient or cell image, thus no general and specific interpretation can be made. This will actually contradict one of

[2]The Lloyd-Max quantiser obtains a quantification based on a set of one-dimensional occurrences. To use this approach to quantise multidimensional weight arrays, it is necessary to develop a multidimensional generalisation of the method. This is because the sequential application of the one-dimensional approach along the different axes could have fatal outcome on the resulting feature efficiency.

the criteria of the property arrays that was mentioned in section 3.2.2. Secondly, it will lead to an indirect standardisation of the element values in question. For the case of grey level and area, such standardisations have been reported to significantly reduce the resulting feature efficiency [48, p.94].

- Consider the application of other segmentation methods. In particular, methods for obtaining the initial set of segmentations that are not based on Niblack's method should be investigated.

- Consider the use of other contextual measurements than the object size in the CSDEMs (and their sum histograms), e.g. the frequency and relative orientation of the estimated primitives or second-order entropy measurements which e.g. measures the clustering of objects with approximately equal size. Refined contextual measurements will require a dataset with better spatial resolution in the cell images and possibly also more cell images within each patient.

- Investigate the prognostic value of using some of the standard property arrays after applying a segmentation method, i.e. evaluate the class specific generalisation of the standard property arrays.

- We request a more thorough analysis of the appropriateness of the estimated PI of the estimator of some relevant performance quantities, especially the CCReq and the CCR, when the PI is estimated from the collection of all subsets (for the cross-validation method) or the collection of all bootstraps (for the bootstrap method) by using the variance of the collection and the normality assumption or by using the histogram percentiles. In particular, this (or these two) estimated PI(s) should be compared to the estimated PI obtained by using other methods, e.g. the mentioned proposed method in [70, pp.6–7] that addresses the issues with a small number of validation patterns and the dependencies of the learning and validation patterns of different subsets or bootstraps while estimating the estimators variance. In particular, the effect of the statistical dependencies (between learning datasets, between validation datasets and between different subsets or bootstraps) and other dependencies, e.g. the number of validation patterns, the true value the estimator attempts to estimate and the complexity of the classifier (in particular both the number of features and the classification method), have on the estimated PIs should be further investigated, and also the effect of such dependencies on the relationship between the estimated variances and the true variance of the estimator.

- We have studied how the entire dataset should be optimally partitioned based on a criterion function where we assume that the conditional pdfs are normally distributed. It would be interesting to generalise these results to other distributions and criterion functions, and in particular to study how the resulting 'optimal' partitioning relates to the number of features and the classification method (as we did for our choices of distributions and criterion function).

- More general and particularly more adapted classification methods should be investigated. This includes the use of nested variance analysis, which allows a direct analysis of the highly hierarchical structure of patients and cells found our dataset and also to perform independent tests on both cell and patient level. This also includes the use of survival analysis. For instance, by using a method based on the Cox proportional hazards regression, we allow the classifier to include the time of the relapse, which may be important because it is likely that a relapse after e.g. a couple of months is a much clearer case than a relapse after e.g. nine years. Also, this regression model allows us to reasonably include all patients, even those who died of causes unrelated to ovarian cancer. For the patients who died of causes unrelated to ovarian cancer, the regression model will use the information that they did not relapse prior to their death, but ignore (censor) their contribution from this point on. Lastly, by using this regression model we could easily also produce other interesting estimates like for instance the probability of relapsing within any specific time (and thereby not restricting our attention to relapse only within ten years).

References

[1] Albregtsen, F., and Nielsen, B. Texture Classification based on Cooccurrence of Gray Level Run Length Matrices. *Australian Journal of Intelligent Information Processing Systems 6*, 1 (2000), 38–45. 26, 30, 32

[2] Baheerathan, S., Albregtsen, F., and Danielsen, H. E. New texture features based on the complexity curve. *Pattern Recognition 32*, 4 (1999), 605–618. 32

[3] Bellman, R. *Adaptive control processes: A guided tour*. Princeton University Press, 1961. 78

[4] Braga-Neto, U. M., and Dougherty, E. R. Is cross-validation valid for small-sample microarray classification? *Bioinformatics 20*, 3 (2004), 374–380. 91, 92, 93

[5] Campbell, F. W., and Robson, J. G. Application of fourier analysis to the visibility of gratings. *The Journal of Physiology 3*, 197 (1968), 551–566. 24

[6] Cover, T. M. The Best Two Independent Measurements Are Not the Two Best. *IEEE Transactions on Systems, Man and Cybernetics SMC-4*, 1 (1974), 116–117. 87

[7] Cover, T. M., and Van Campenhout, J. M. On the Possible Orderings in the Measurement Selection Problem. *IEEE Transactions on Systems, Man and Cybernetics 7*, 9 (1977), 657–661. 87

[8] Danielsen, H. E. *Premalignant Changes in DNA Organization in Mouse Liver After Diethylnitrosamine Treatment*. PhD thesis, The Norwegian Radium Hospital and Institute for Cancer Research : The Norwegian Cancer Society, 1991. 5, 6, 58, 122

[9] Danielsen, H. E., Farrants, G., and Reith, A. Changes in Chromatin Organization in Regenerating, Preneoplastic and Neoplastic Hepatocytes During Experimental Liver Carcinogenesis in Mice. In *Premalignant Changes in DNA Organization in Mouse Liver After Diethylnitrosamine Treatment*, H. E. Danielsen, Ed. University of Oslo, Oslo, Norway, 1991. 5

[10] De Valois, R. L., Albrecht, D. G., and Thorell, L. G. Spatial frequency selectivity of cells in macaque visual cortex. *Vision Research 22*, 5 (1982), 545–559. 24

[11] Devore, J. L., and Berk, K. N. *Modern Mathematical Statistics with Applications*. Duxbury Press, 2007. ii, 29, 30, 34, 67, 68, 103, 116, 170

[12] Ding, C., and Peng, H. Minimum Redundancy Feature Selection from Microarray Gene Expression Data. In *Proceedings of the IEEE Computational Systems Bioinformatics Conference* (2003), pp. 523–528. 90

[13] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*, 2 ed. Wiley-Interscience, 2001. ii, 28, 63, 64, 65, 66, 67, 69, 72, 76, 77, 80, 81, 84, 85, 86, 87

[14] Efron, B. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *Journal of the American Statistical Association 78*, 382 (1983), 316–331. 92

[15] Elashoff, J. D., Elashoff, R. M., and Goldman, G. E. On the choice of variables in classification problems with dichotomous variables. *Biometrika 54*, 3–4 (1967), 668–670. 87

[16] Encyclopædia Britannica. cell (biology), 2010. Retrieved November 29 2010 from http://www.britannica.com/EBchecked/topic/101396/cell. 1

[17] Fisher, R. A. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics 7* (1936), 179–188. 85

[18] Galloway, M. M. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing 4*, 2 (1975), 172–179. 6, 24

[19] Gonzalez, R. C., and Woods, R. E. *Digital Image Processing*, 3 ed. Prentice Hall, 2008. 36

[20] Haralick, R. M., Shanmugam, K., and Dinstein, I. Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics 3*, 6 (1973), 610–621. 6, 24, 25, 26

[21] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 5th printing with corrections, 2 ed. Springer, 2011. 91, 92, 93

[22] Hein, J. L. *Discrete Structures, Logic, and Computability*, 2 ed. Jones and Bartlett Publishers International, 2002. iii, 66

[23] Henige, D. A Cooperative Publishing Model for Sustainable Scholarship. *Journal of Scholarly Publishing 37*, 2 (2006), 99–118. ii

[24] Hughes, G. F. On the Mean Accuracy of Statistical Pattern Recognizers. *IEEE Transactions on Information Theory 14* (1968), 55–63. 83

[25] Isaksson, A., Wallman, M., Göransson, H., and Gustafsson, M. G. Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recognition Letters 29* (2008), 1960–1965. 91, 92, 93, 94

[26] Jain, A. K., Duin, R. P. W., and Mao, J. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 1 (2000), 4–37. 86, 87, 88

[27] Knutsson, H., and Granlund, G. H. Texture analysis using two-dimensional quadrature filters. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management* (1983), pp. 1–8 (relative pages for citation purposes). 56

[28] Kosary, C. L. Cancer of the Ovary. In *SEER Survival Monograph: Cancer Survival Among Adults: U.S. SEER Program, 1988-2001, Patient and Tumor Characteristics*, L. A. G. Ries, J. L. J. Young, G. E. Keel, M. P. Eisner, Y. D. Lin, and M.-J. D. Horner, Eds. National Cancer Institute, SEER Program, NIH Pub. No. 07-6215, Bethesda, MD, 2007, ch. 16, pp. 133–144. 10

[29] Kristensen, G. B., Kildal, W., Abeler, V. M., Kaern, J., Vergote, I., Tropé, C. G., and Danielsen, H. E. Large-scale genomic instability predicts long-term outcome for women with invasive stage I ovarian cancer. *Annals of Oncology 14*, 10 (2003), 1494–1500. 6, 9, 16, 19, 20, 21

[30] Kruskal, W. H., and Wallis, W. A. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association 47*, 260 (1952), 583–621. 170

[31] Lachenbruch, P. A., and Mickey, M. R. Estimation of Error Rates in Discriminant Analysis. *Technometrics 10*, 1 (1968), 1–11. 91

[32] Lilliefors, H. W. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association 62*, 318 (1967), 399–402. 116

[33] Lyche, T. Lecture Notes for Inf-Mat 4350, 2010, 2010. Retrieved February 10 2011 from http://www.uio.no/studier/emner/matnat/ifi/INF-MAT4350/h10/book2010.pdf. 27, 89

[34] Mahalanobis, P. C. On the generalised distance in statistics. *Proceedings National Institute of Science of India 2*, 1 (1936), 49–55. 27

[35] Maître, H., Bloch, I., and Sigelle, M. Spatial entropy: A tool for controlling contextual classification convergence. *IEEE International Conference on Image Processing 2* (1994), 212–216. 39

[36] Mann, H. B., and Whitney, D. R. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics 18*, 1 (1947), 50–60. 170

[37] Marill, T., and Green, D. M. On the Effectiveness of Receptors in Recognition Systems. *IEEE Transactions on Information Theory 9*, 1 (1963), 11–17. 88

[38] Niblack, W. *An Introduction to Digital Image Processing*, 2 ed. Prentice-Hall, 1986. 34

[39] Nielsen, B. *Radial Differentiation of Low Dimensionality Adaptive Texture Feature Vectors in Cell Nuclei as a Tool in Tumour Pathology.* PhD

thesis, Department of Informatics, University of Oslo and Division of Digital Pathology, Department of Pathology, The Norwegian Radium Hospital, 2001. 6, 29

[40] Nielsen, B., Albregtsen, F., Baheerathan, S., and Danielsen, H. E. Peel-off-scanning to obtain radial differentiation of fractal and complexity features in cell nuclei. In *Proceedings of the Vision Interface* (2000), pp. 1–7 (relative pages for citation purposes). 56

[41] Nielsen, B., Albregtsen, F., and Danielsen, H. E. Fractal signature vectors and lacunarity class distance matrices to extract new adaptive texture features from cell nuclei. In *Fractals in Biology and Medicine*, vol. 3 of *Mathematics and Biosciences in Interaction*. Birkhäuser Basel, 2002, pp. 55–65. 24, 31

[42] Nielsen, B., Albregtsen, F., and Danielsen, H. E. Nuclear texture analysis of two different prognostic classes of early ovarian cancer based on gray level entropy matrices. *Analytical Cellular Pathology 25* (2003), D.17. 32, 33

[43] Nielsen, B., Albregtsen, F., and Danielsen, H. E. Low Dimensional Adaptive Texture Feature Vectors From Class Distance and Class Difference Matrices. *IEEE Transactions on Medical Imaging 23*, 1 (2004), 73–84. 29, 32

[44] Nielsen, B., Albregtsen, F., and Danielsen, H. E. Fractal Analysis of Monolayer Cell Nuclei from Two Different Prognostic Classes of Early Ovarian Cancer. In *Fractals in Biology and Medicine*, W. Alt, F. Adler, M. Chaplain, A. Deutsch, A. Dress, D. Krakauer, R. T. Tranquillo, G. A. Losa, D. Merlini, T. F. Nonnenmacher, and E. R. Weibel, Eds., vol. 4 of *Mathematics and Biosciences in Interaction*. Birkhäuser Basel, 2005, pp. 175–186. 32

[45] Nielsen, B., Albregtsen, F., and Danielsen, H. E. Statistical Nuclear Texture Analysis in Cancer Research: A Review of Methods and Applications. *Critical Reviews™ in Oncogenesis 14* (2008), 89–164. 5, 6, 7, 15, 16, 26, 32, 34, 83, 87, 88, 91, 95, 100, 104

[46] Nielsen, B., Albregtsen, F., Kildal, W., Abeler, V. M., Kristensen, G. B., and Danielsen, H. E. The prognostic value of adaptive nuclear texture features from patient gray level entropy matrices in early stage ovarian cancer. *Cellular Oncology (submitted)* (2011). 33, 34, 37, 60

[47] Nielsen, B., Albregtsen, F., Kildal, W., and Danielsen, H. E. Prognostic classification of early ovarian cancer based on very low dimensionality adaptive texture feature vectors from cell nuclei from monolayers and histological sections. *Analytical Cellular Pathology 23*, 2 (2001), 75–88. 10, 14, 29, 32

[48] Nielsen, B., and Danielsen, H. E. Prognostic value of adaptive textural features - The effect of standardizing nuclear first-order gray level statistics and mixing information from nuclei having different area. *Cellular Oncology 28*, 3 (2006), 85–95. 24, 32, 34, 60, 172

[49] Nordby, P. A combined structural/statistical texture analysis of monolayer ovarian cancer cell nuclei. Master's thesis, Department of Informatics, University of Oslo, 2010. 23, 29, 34, 35, 36, 37, 44, 48, 49, 51, 52, 53, 55, 56, 57, 59, 109, 110, 130, 163

[50] Ørom, U. A., Derrien, T., Beringer, M., Gumireddy, K., Gardini, A., Bussotti, G., Lai, F., Zytnicki, M., Notredame, C., Huang, Q., Guigo, R., and Shiekhattar, R. Long Noncoding RNAs with Enhancer-like Function in Human Cells. *Cell Press 143* (2010), 46–58. 2

[51] Peng, H., Long, F., and Ding, C. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 8 (2005), 1226–1238. 88, 89, 90

[52] Pennisi, E. DNA Study Forces Rethink of What It Means to Be a Gene. *Science 316*, 5831 (2007), 1556–1557. 2

[53] Pudil, P., Novovičová, J., and Kittler, J. Floating search methods in feature selection. *Pattern Recognition Letters 15*, 11 (1994), 1119–1125. 88

[54] Randen, T., and Husøy, J. H. Filtering for Texture Classification: A Comparative Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 4 (1999), 291–310. 7

[55] Raudys, u. J. On Dimensionality, Sample Size, and Classification Error of Nonparametric Linear Classification Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 6 (1997), 667–671. 82

[56] Raudys, u. J., and Jain, A. K. Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 3 (1991), 252–264. 8, 64, 66, 67, 74, 77, 80, 83, 87, 88, 90, 91, 93, 94, 95, 96

[57] Rice, J. A. *Mathematical Statistics and Data Analysis*, 3 ed. Duxbury Press, 2007. ii

[58] Schmitt, C. A., Fridman, J. S., Yang, M., Baranov, E., Hoffman, R. M., and Lowe, S. W. Dissecting p53 tumor suppressor functions in vivo. *Cancer Cell 1*, 3 (2002), 289–298. 5

[59] Schulerud, H., and Albregtsen, F. Many are called, but few are chosen. feature selection and error estimation in high dimensional spaces. *Computer Methods and Programs in Biomedicine 73*, 2 (2004), 91–99. 82, 84, 93, 136

[60] Schulerud, H., Kristensen, G. B., Liestol, K., Vlatkovic, L., Reith, A., Albregtsen, F., and Danielsen, H. E. A review of caveats in statistical nuclear image analysis. *Analytical Cellular Pathology 16*, 2 (1998), 63–82. 7, 15, 16, 93, 102

[61] Shannon, C. E. A Mathematical Theory of Communication. *The Bell System Technical Journal 27* (1948), 379–423 and 623–656. 32

[62] Somol, P., Pudil, P., Novovičová, J., and Paclík, P. Adaptive floating search methods in feature selection. *Pattern Recognition Letters 20*, 11–13 (1999), 1157–1163. 88

[63] Szymańska, K., and Hainaut, P. TP53 and mutations in human cancer. *Acta Biochimica Polonica 50*, 1 (2003), 231–238. 4

[64] Tsybrovskyy, O., and Berghold, A. Primary unit for statistical analysis in morphometry: patient or cell? *Analytical Cellular Pathology 18*, 4 (1999), 191–202. 15, 16

[65] Tuceryan, M., and Jain, A. K. Texture analysis. In *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1998, pp. 207–248. 6, 14, 23, 56

[66] Tupin, F., Sigelle, M., and Maître, H. Definition of a spatial entropy and its use for texture discrimination. *IEEE International Conference on Image Processing 1* (2000), 725–728. 39

[67] University of Oslo. The IMRaD outline, 2011. Retrieved March 2 2011 from http://www.uio.no/studier/emner/hf/imk/MEVIT4725/h04/resources/imrad.xml. 8

[68] Unser, M. Sum and Difference Histograms for Texture Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*, 1 (1986), 118–125. 30, 60

[69] Whitney, A. W. A Direct Method of Nonparametric Measurement Selection. *IEEE Transactions on Computers C-20*, 9 (1971), 1100–1103. 88

[70] Wickenberg-Bolin, U., Göransson, H., Fryknäs, M., Gustafsson, M. G., and Isaksson, A. Improved variance estimation of classification performance via reduction of bias caused by small sample size. *BMC Bioinformatics 7*, 127 (2006), 1–8 (relative pages for citation purposes). 94, 172

[71] Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics Bulletin 1*, 6 (1945), 80–83. 170

[72] Yanowitz, S. D., and Bruckstein, A. M. A New Method for Image Segmentation. *Computer Vision, Graphics, and Image Processing 46* (1989), 82–95. 36, 49, 109

[73] Yogesan, K., Albregtsen, F., and Danielsen, H. E. Gray Level Variance Matrix: A New Approach to Higher Order Statistical Texture Analysis. In *Proceedings of the 3rd International Conference on Automation, Robotics and Computer Vision* (1994), vol. 2, pp. 658–663. 114

[74] Yogesan, K., Jørgensen, T., Albregtsen, F., Tveter, K. J., and Danielsen, H. E. Entropy-Based Texture Analysis of Chromatin Structure in Advanced Prostate Cancer. *Cytometry 24*, 3 (1996), 268–276. 32