**NTNU – Trondheim**
Norwegian University of
Science and Technology

# Quantified is simplified; Treating the spatial entropy as continuous for prognostics of early ovarian cancer

Håvard Malm Geithus

Master of Science in Computer Science
Submission date: June 2013
Supervisor: Richard E. Blake, IDI
Co-supervisor: Professor Håvard E. Danielsen, IMI, OUS
M. Sc. Andreas Kleppe, IFI, UiO

Norwegian University of Science and Technology
Department of Computer and Information Science

# Acknowledgements

This thesis was submitted to the Faculty of Information Technology, Mathematics and Electrical Engineering at the Norwegian University of Science and Technology (NTNU) in partial fulfillment of the requirements for the degree *Master of Science* in *Computer Science*.

The study was started in January 2013 and completed in June 2013 and was carried out at the Depertment of Computer and Information Science (IDI) and in cooporation with the Institute for Medical Informatics (IMI) at the Oslo University Hospital (OUS).

I would like to thank my friend and supervisor M.Sc. Andreas Kleppe for his continuous support, excellent advice and contagious professionalism and precision. My thanks and great appreciation also go to my supervisors Professor Richard E. Blake and Professor Håvard E. Danielsen for making this study possible.

Trondheim, June 2013
Håvard Malm Geithus

# Abstract

A substantial number of studies have proven that analysing the texture of DNA-specific stained cancer cell nuclei can provide robust and reliable prognostic information. Such information is important to make a qualified selection of the appropriate treatments for the patients.

A recent texture approach based on adaptive features extracted from the class specific dual entropy matrix (CSDEM) has shown promising results. The approach used relatively coarse quantification of the entropy values to reduce overfitting. This quantification can easily reduce the performance of the approach, and will certainly require detailed domain knowledge in order to fully utilise its potential.

We will in this study describe a method that uses the class specific entropy values in their continuous nature. The method uses an adaptive continuous discrimination function, based on density estimation, that is able to estimate the discriminative value of the entropies on a continuous scale.

We have evaluated our method using statistical bootstrapping on a dataset containing about 38 000 cell nucleus images collected from 134 patients with early ovarian cancer. We achieve results that are consistently better than the quantified approach based on CSDEM, and our results are more easily obtained as domain knowledge requirements are reduced. Considering our method as a generalisation to the continuous domain, this is a good result that reinforces the promise of using the class specific entropies for prognostics of early ovarian cancer.

# Contents

# 1. Introduction

A substantial number of studies have proven that analysing the texture of DNA-specific stained cancer cell nuclei can provide robust and reliable prognostic information [8]. Such information is important to make a qualified selection of the appropriate treatments for the patients. A recent texture approach based on a novel texture analysis concept, coined the *class specific dual entropy matrix* (CSDEM), has shown promising results on a dataset containing about 38000 cell nucleus images collected from 134 patients with early ovarian cancer. The approach used relatively coarse quantification of the entropy values to reduce overfitting. However, such quantification can also easily reduce the performance of the approach, and tweaking the quantification parameters to obtain the best possible performance will typically require detailed knowledge about the dataset. We will in this study improve and generalise the approach by using the entropy values in their continuous nature to adaptively estimate their discriminative value on a continuous scale. This generalisation is likely to improve performance and significantly reduce the required domain knowledge.

We will begin this section with a short introduction of the dataset used in this study. Then we introduce the study that presented the approach we are generalising, and how we describe it in parallell to our concepts. Finally we describe the main principles of the segmentation method that is applied to the digital images of the dataset.

## 1.1 Dataset

In this thesis we will study a dataset of about 38000 cell nucleus images captured from 134 patients treated for early ovarian cancer in the 1980s. As part of the treatment, every patient had both ovaries and the uterus completely removed. From the relevant ovary, a cancerous tissue sample was extracted and the nuclei of its comprising cells were stained to highlight DNA. The nucleus images were then produced by an optical microscopy imaging technique in which light travels through the DNA-specific stained nucleus, is partially absorbed or reflected, thus causing illuminance variation at the camera's sensor chip. Shading correction was later performed to remove undesirable regularities from the images. Each cell nucleus was then segmented using a manually chosen global threshold while requiring that the region of nucleus pixels contained no holes when using 8-connectivity, and finally the non-epithelial, incomplete and connected nuclei were discarded. We will subsequently refer to the images of the remaining, segmented nuclei as *cell nucleus images*.

Figure 1.1.1 shows three representative cell nucleus images from each of the

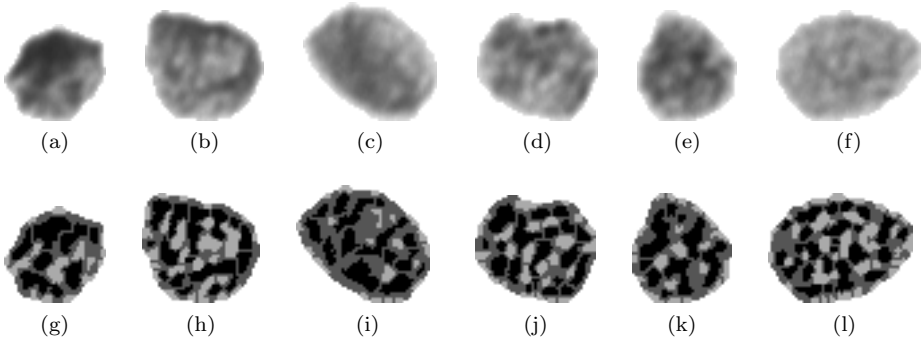


*Figure 1.1.1: A selection of cell nucleus images (a-f) and below; their segmentations after applying the watershed-based algorithm (g-l). The three leftmost cell nucleus images (a-c) are from a patient with good prognosis, while the rightmost three (d-f) are from a patient with poor prognosis. The segmentation parameters are* $w = 9$, $k_d = 0.3$ *and g)* $k_b = 0.3$, *h)* $k_b = 0.3$, *i)* $k_b = 0.6$, *j)* $k_b = 0.4$, *k)* $k_b = 0.5$, *l)* $k_b = 0.6$

two prognosis groups of the dataset. Every cell nucleus image is a gray-level image in which the pixel darkness is related to the density of DNA. Since this is a projection of a three-dimensional structure onto the two-dimensional plane, we lose information about the precise spatial configuration of DNA. Two disjunct regions might e.g. appear as one as a result of the projection. In a sense, this is a challenge when using this dataset for texture analysis.

The patients in the dataset are categorised either as being *relapse-free* or having *ovarian cancer relapse* with respect to a ten year period following the surgery. We will refer to these categories as *good prognosis* and *poor prognosis*, respectively. Among the 134 patients, 97 are labelled good prognosis, while 37 poor prognosis. This partitioning into prognosis groups differs somewhat from that in the study of Kleppe [5] where the numbers were 94 and 40, respectively. The reason why the three patients have changed prognosis group is that the clinical data leading to the prognoses used in the present study have been revisited after the study in [5]. A consequence of this discrepancy is that our classification results are not directly comparable with the classification results presented in [5]. We have therefore reimplemented the most promising approach in [5], and re-evaluated the classifier with the updated prognosis groups. That result will serve as a benchmark for our results and is available in section 6 where the results are presented.

## 1.2 Previous work

The texture analysis method which achieved the best results for this particular dataset, is described in [5]. The study introduced a full classification system for estimating the prognosis of unseen patients with early overian cancer, comprising some novel and quite promising elements. In particular, a novel adaptive segmentation method and novel texture features. These elements and ideas are to a large extent carried into this work. Instead of describing the previous approach in detail here, it will be described alongside with the presentation of our approach, while making sure to emphasize what we are doing differently and

why. In short telling, the main difference is that we handle the entropies in their continuous nature, with all the implications this entails. In contrast, the entropies were quantified in [5], which can reduce the classification performance compared to our continuous approach. In addition, it may also require parameters to be tuned for the specific dataset in order to achieve its best possible performance.

## 1.3 Segmentation

A sequence of processing steps is performed on the cell nucleus images to produce the segmented cell nucleus images. This segmentation pipeline and its rationale will now be described briefly. Note that the segmentation method was first introduced in [5] and the reader is hereby referred to that study for a more detailed discussion of the method.

The basic aim of the segmentation method is to segment the cell nucleus image into three primitives or classes, which corresponds to the bright, gray and dark regions of the cell nucleus images. The role of the grey primitive is merely to act as a margin between the dark and bright primitives, i.e. we wish to separate the dark and bright regions from the grey regions, rather than separating the dark and bright regions from eachother. Later, it is only from the dark and bright primitives we will extract texture features.

The first step of the segmentation pipeline is to use an extended version of Niblack's adaptive segmentation algorithm with two thresholds, as proposed by Nordby [9]. Given a window size $w$, uncertainty parameters $k_d$ and $k_b$, the input image $A \in \mathbb{N}_0^{m,n}$and the segmentation image $N \in \{0,1,2\}^{m,n}$, this extension is defined as

$$N(i,j) = \begin{cases} 0 & \text{if } A(i,j) < t_d(i,j) \\ 1 & \text{if } t_d(i,j) \leq A(i,j) \leq t_b(i,j) \\ 2 & \text{if } A(i,j) > t_b(i,j) \end{cases} \tag{1.3.1}$$

where

$$t_d(i,j) = \mu_w(i,j) - k_d\sigma_w(i,j) \tag{1.3.2}$$

$$t_b(i,j) = \mu_w(i,j) + k_b\sigma_w(i,j) \tag{1.3.3}$$

for all the pixels $i = 1, ..., m$ and $j = 1, ..., n$, 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 $w$ centered at $(i,j)$ in $A$.

There are three parameters that needs to be specified, $w$, $k_b$ and $k_d$. To fit these parameters we will use the gradient magnitude of the input image to describe the fitness of the segmentation, and the parameters that maximises the fitness are selected. The set of possible values for the window size $w$ is set to $\{5,7,9\}$. The uncertainty parameters $k_b$ and $k_d$ can both take values from the set $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Basically, we let the fitness measure how well the edges of segmented primitives corresponds with the edge responses in the gradient magnitude image, as suggested by Yanowitz and Bruckstein [13] in connection with the validation step of their segmentation method. Specifically, we compute the mean gradient magnitude of the objects boundary pixels for

the dark and bright segmentation class, and let the fitness measure be the mean of these two means.

While [9] used the $L_1$-norm (also known as taxicab norm) of the estimated first order derivatives to calculate the gradient magnitude, [5] argues to rather use the $L_2$-norm (euclidean norm) of the estimated first order derivatives. The reason being that the $L_1$-norm produces a gradient magnitude estimate that emphasizes diagonal intensity changes significantly more than the gradient magnitude estimate produced by the $L_2$-norm [5, p.48]. The $L_2$-norm is therefore more appropriate for estimating the gradient magnitude and is used in our implementation.

After the images are segmented using the parameters which are optimal with respect to the fitness function, one important issue remains to be adressed. When two or more bright or dark primitives become direct neighbours or overlap, they are likely to be segmented as a single dark or bright region, but this is problematic as they should be treated as separate primitives. Concretely, for every cell nucleus image we will later extract the size of a primitive and the grey level values of the primitives in the corresponding input image. Such merged primitives must thus be separated in order to obtain the correct number of primitives and correct size of these primitives. For this purpose we use the well-known watershed algorithm. It is very well suited to separate bright primitives with multiple intensity peaks, and dark primitives with multiple intensity valleys. A selection of cell nucleus images from a good prognosis patient and a poor prognosis patient, and their segmentations after applying the watershed transform, are available in figure 1.1.1. The reader is referred to [5, p.53] for a detailed description of how the watershed algorithm is applied and implemented in that and the present study.

# 2. Class specific dual entropies

In this section we will introduce the concept of *class specific dual entropies* (CSDE). We first give the definition, then we describe the *CSDE-space* and the *CSDEsum-space* in terms of these dual entropies. Finally we describe how these spaces were quantified in [5].

## 2.1 Definition

In the style of Maître et al. [6, p.212-213] and Tupin et al. [11, p.725], let every location in the cell nucleus image be described by the following three values:

- The gray level $g \in \{0, 1, ..., G-1\}$, where $G$ is the number of possible gray levels. The images are uniformly requantified so that $G = 64$.
- The segmentation label $l \in \{1, 2, ..., L\}$, where $L$ is number of segmentation classes. We will in this study use the segmentated cell nucleus images from the previous section.
- The context value. We will in this study use the size of the object encapsulating the pixel in the segmentation image, $s \in \{1, 2, ..., S\}$, where $S$ is the largest possible pixel area of a segmented object.

Let $q(g, s, l)$ denote the probability that the combination of grey level $g$, primitive size $s$ and segmentation label $l$ occurs at a particular location. We will now go through a series of distribution marginalisations and finally compute the Shannon entropy of each of two marginals. These entropies are the class specific dual entropies.

The class (or label) marginal distribution is

$$q(g, s|l) = \frac{q(g, s, l)}{\sum_{g'=0}^{G-1} \sum_{s'=0}^{S} q(g', s', l)} \tag{2.1.1}$$

If we perform another level of marginalisation, we arrive at the class specific grey level histogram $q(g|l)$ and the class specific primitive size histogram $q(s|l)$;

$$q(g|l) = \sum_{s=1}^{S} q(g, s|l) \tag{2.1.2}$$

$$q(s|l) = \sum_{g=0}^{G-1} q(g, s|l) \tag{2.1.3}$$

and now we can define the class specific grey level entropy $\epsilon_l$ and the class specific spatial entropy $\zeta_l$, which is simply the Shannon entropy of each of the two previous marginals

$$\epsilon_l = -\sum_{g=0}^{G-1} q(g|l) \log q(g|l) \tag{2.1.4}$$

$$\zeta_l = -\sum_{s=1}^{S} q(s|l) \log q(s|l) \tag{2.1.5}$$

These two entropies constitute a specific instantiation of class specific dual entropies.

## 2.2 CSDE-space and CSDEsum-space

Define CSDE-space as the Euclidean space $[0, \infty) \times [0, \infty)$ by associating spatial entropy values with the horizontal axis and grey level entropy values with the vertical axis. Given a segmenation class, a cell nucleus image then corresponds to a single point in this space.

To get a feeling for how populated this space is going to be, recall that our dataset includes about 38000 cell nucleus images from 134 patients. These patients are partitioned into two prognosis groups; 97 patients of good prognosis and 37 patients of poor prognosis. This means there are about 10 000 cell nucleus images associated with poor prognosis. The rest are associated with good prognosis. Furthermore, in the bootstrap procedure (described in section 5.1) the prognosis groups are each further split into a 70% training set and a 30% test set, leaving only about 7000 points in the training set for the case of poor prognosis. In the case of good prognosis, the number of points in the training set is about 19000. In other words, because the prognosis group sizes are skewed, the number of relevant points in CSDE-space can be about as low as 7000.

Our approach to discriminating between these two prognosis groups is presented in section 3. In short, it relies on some distance measure between the probability density at arbitrary locations in CSDE-space when populated by good prognosis patients and poor prognosis patients separately. Reliability and accuracy of the density estimates of CSDE-space must thus be given special attention, and especially so considering the estimates are calculated from as few as 7000 samples. How the tradeoff between reliability and accuracy is controlled is covered in section 4 about non-parametric density estimation.

We will also be interested in the projection of CSDE-space onto the identity line to significantly increase the density of samples. This projection is achieved by simply summing the two entropies corresponding to the axes in the CSDE-space. We will refer to the projected space as the *CSDEsum*-space.

## 2.3 Quantifiation

Another approach for dealing with reliability, different from ours, is to coarsely quantify the entropies. This approach was taken in [5, p.41], giving the *class specific dual entropy matrix* (CSDEM), which is a quantification of CSDE-space. This introduces two additional parameters $q_G$ and $q_S$ denoting the number of quantification levels per integer entropy for the class specific grey level and spatial entropy, respectively. Given a segmentation class, a cell nucleus image corresponds to a single point in this matrix, and the quantification parameters and a rounding function specify the exact location in the matrix at which the dual entropies contribute:

$$\delta(x - r(q_G\epsilon_l), y - r(q_S\zeta_l)) \tag{2.3.1}$$

where $\delta$ is the Kronecker delta and $r : [0, \infty) \mapsto \mathbb{N}_0$ is any rounding function. The CSDEM is thus a matrix of zeros, except from position $(r(q_G\epsilon_l), r(q_S\zeta_l))$ where the value is one. Intuitively, the quantification parameters imposes a grid on CSDE-space, and the rounding function dictates which of the four surrounding corners the dual entropies "snaps into".

# 3. Adaptive texture features

In this section we will describe a technique for adaptively extracting features from the training set[1]. The ideas originate from Walker et al. [12], and was later developed by Albregtsen et al. [2, 7, 8, 1]. We will not give a general description, but rather give a description specific to our study. First we describe the basic principes, then we describe our approach for handling continous properties (section 3.1) and finally we describe how discrete or quantified properties were handled in [5] (section 3.2).

The basic principle of adaptive features is to let the samples of the training set design one weight function for each class of the classification problem. Each of these weight functions estimates the discriminative value of each point in property space with respect to the true class. If there are any regions in property space that consistently yields high values in the weight funciton of a given class, or a proper subset of classes, then this is a region of high discriminative value.

In our case we have only two classes; the two prognosis groups, and this allows us to simplify and use a function of the form $f : \mathbb{R}^2 \mapsto \mathbb{R}$ in which positive values represents the relative evidence for the good prognosis group, and similarly for the negative values; they represent the relative evidence for the poor prognosis group. We will refer to this function as the discrimination function. The purpose of the discrimination function is to adaptively extract features of high discriminatory value from the training set.

Given that our properties are the continuous dual entropies, $f$ is a mapping from locations in CSDE-space to a real-valued discrimination value. If we quantify we will instead have a *discrimination matrix* which can be written as the the function $D : \mathbb{N}_0^2 \mapsto \mathbb{R}$, assuming the matrix is zero-indexed.

## 3.1 Continuous discrimination function

Here we define the discrimination function used in this study. To aid our definition, we first define

- $\Lambda = \{\lambda_0, \lambda_1, \lambda_2\}$ as the set of segmentation classes
- $\Omega = \{\omega_0, \omega_1\}$ as the set of prognosis groups.

[1] Note that one must prior to this step specify the properties to use. In this study the properties are the dual entropies (given a segmentation class)

In the following, because the grey class is ignored, we will use the convention of denoting the bright and dark segmentation classes as $\lambda_0$ and $\lambda_1$, respectively, while the good and poor prognosis groups are denoted by $\omega_0$ and $\omega_1$, respectively.

Denote the probability density function (pdf) of the training samples $\vec{x}$ with segmentation class $\lambda_l$ and prognosis group $\omega_c$ by $f_{\vec{X}|\Omega=\omega_c,\Lambda=\lambda_l}(\vec{x})$. Using these pdfs we specify one discrimination function for each of our two relevant segmentation classes,

$$f_b(\vec{x}|c) = \log \frac{f_{\vec{X}|\Omega=\omega_0,\Lambda=\lambda_0}(\vec{x}) + c}{f_{\vec{X}|\Omega=\omega_1,\Lambda=\lambda_0}(\vec{x}) + c} \tag{3.1.1}$$

and

$$f_d(\vec{x}|c) = \log \frac{f_{\vec{X}|\Omega=\omega_0,\Lambda=\lambda_1}(\vec{x}) + c}{f_{\vec{X}|\Omega=\omega_1,\Lambda=\lambda_1}(\vec{x}) + c} \tag{3.1.2}$$

where $c$ is some constant used to mitigate the problem of low reliability of the density estimates in sparsely populated regions, which often will result in a very high discrimination value if $c$ is excluded or set to zero. Unreliable density estimates close to zero is especially unfortunate because the logarithm is so sensitive close to zero, as are unreliable estimates where the denominator is close to zero, even after applying the logarithm. The letters $b$ and $d$ are used as subscripts to denote the bright and dark segmentation classes, respectively, and we will refer to these two transformation functions as the bright and dark discrimination functions, respectively. This naming reflects our use of these functions, which is to transform a pair of entropy values into a single feature of high discriminative value.

After a discrimination function has been computed using the training set, we can extract the dual entropies $\vec{x}$ from a cell nucleus image belonging to a patient whose prognosis is unknown and evaluate the discriminative value of $of\vec{x}$ using this function.

## 3.2 Discrete discrimination matrix

The entropies was quantified in [5]. As mentioned earlier, in section 2.3, quantification was achieved by specifying a fixed number of values per integer for each of the two entropies, and a rounding function.

Instead of continuous probability density functions, we are now dealing with discrete probability functions, and in practice, the discrimination function becomes a finite $I \times J$ discrimination matrix $D$. In our context, the sizes of the dimensions of $D$ can be determined from the number of grey level values and the maximum object size in the segmented cell nucleus images. In our definitions, we will be reserving subscripts for the specification of prognosis group $\omega$ and segmentation class $\lambda$, so our matrices are always index using parantheses, like this: $D(i, j)$.

To give a precise description of how the elements of $D$ are calculated we need some additional definitions. Define the matrices $\mu_{\Omega=\omega_c,\Lambda=\lambda_l}$ and $\sigma_{\Omega=\omega_c,\Lambda=\lambda_l}$ for prognosis group $\omega_c$ and segmentation class $\lambda_l$, both having dimensions $I \times J$. These are the parameters of $IJ$ normal distributions, reflecting the assumption

that each element of the quantified CSDE-space is normally distributed. Let us formalise this assumption by defining the matrix $Q$ of random variables:

$$Q_{\Omega=\omega_c,\Lambda=\lambda_l}(i,j) \sim \mathcal{N}(\mu_{\Omega=\omega_c,\Lambda=\lambda_l}(i,j), \sigma^2_{\Omega=\omega_c,\Lambda=\lambda_l}(i,j)) \tag{3.2.1}$$

and let us also assume that

$$\sigma_{\Omega=\omega_0,\Lambda=\lambda_l}(i,j) = \sigma_{\Omega=\omega_1,\Lambda=\lambda_l}(i,j) = \sigma_{\Lambda=\lambda_l}(i,j) \tag{3.2.2}$$

for $i = 1, ..., I$ and $j = 1, ..., J$.

We should note that each observation of $Q_{\Omega=\omega_c,\Lambda=\lambda_l}(i,j)$ is the normalised contribution of all the cell nucleus images belonging to a $I \times J$ patient with prognosis class $\omega_c$. We then compute the arithmetic mean and estimated standard deviation of those observations to estimate the parameters of $Q_{\Omega=\omega_c,\Lambda=\lambda_l}(i,j)$. This ensures that each patient contributes equally to the estimation of $Q$, no matter how many cell nucleus images the different patients have.

To calculate the elements of the discrimination matrix, we estimate the Mahalanobis distance for measuring the distance between two distributions representing good and poor prognosis. In the univariate case this is simply

$$m(\mu_1, \mu_2) = \frac{|\mu_1 - \mu_2|}{\sigma}. \tag{3.2.3}$$

where $\sigma$ is the common standard deviation of the two distributions. By omitting the absolute value, we let the sign specify which of the two prognosis groups the elements in $D$ is evidence for. Let us now specify one discrimination matrix for each of our segmentation classes

$$D_b(i,j) = \frac{\mu_{\Omega=\omega_0,\Lambda=\lambda_0}(i,j) - \mu_{\Omega=\omega_1,\Lambda=\lambda_0}(i,j)}{\sigma_{\Lambda=\lambda_0}(i,j)} \tag{3.2.4}$$

and

$$D_d(i,j) = \frac{\mu_{\Omega=\omega_0,\Lambda=\lambda_1}(i,j) - \mu_{\Omega=\omega_1,\Lambda=\lambda_1}(i,j)}{\sigma_{\Lambda=\lambda_1}(i,j)} \tag{3.2.5}$$

for $i = 1, ..., I$ and $j = 1, ..., J$.

Note that, as in [5, p.34], we have just made the assumption that each element of the quantified CSDE-space is normally distributed. If the cells of each patient were independent, we could use the central limit theorem to justify this assumption, but Schulerud et al. [10] points out that the cells of a patient are not and should not be treated as independent. We may therefore not apply the central limit theorem to justify this assumption, but we can still suspect that the normal approximation is good enough to use the estimate of the Mahalanobis distance as a distance measure between the classes, as the Mahalanobis distance is a relatively robust measurement, at least if the distributions are fairly symmetric.

## 3.3 Discussion

We have described the continuous discrimination function used in this work, and the discrete discrimination matrix used in [5]. Let us now discuss some of their similarities and differences.

For every cell nucleus image, *both* discrimination functions provide an estimate of its discriminative value with respect to our two prognosis groups. If we let each cell contribute to the feature value of its patient by its discriminative value, we are in effect letting each cell contribute with its own *weighted* and *signed* vote. The weighting is highly desirable, because as described in section 5, the cells have are not homogeneous in terms of cancer.

The trade-off between *reliability* and *accuracy* of the density estimates is of major importance, as this tradeoff is directly transfered to our discrimination functions, which in turn is used to create the features for classification. We will now discuss how the quantified and the continuous approach differs in this respect.

Let us first consider the method of quantification. Here, the reliability is increased by imposing a coarse grid on CSDE-space, resulting in a collection of equally sized bins $B$. Some of these bins will be highly populated by entropies, while other sparsely populated. The reliability in each bin will therefore vary in general. If we now present a query point $\vec{x}$, and use the discrimination estimate of the bin containing $\vec{x}$, then we will have an accurate estimate if $\vec{x}$ is located exactly at the center of the bin, while the reliability of the estimate depends on how many entropies were in that bin. If $\vec{x}$ were located near the edge of a bin, we would still use the estimate for the center of that bin, thus the accuracy is reduced, but the reliability could be said to be equal.

Now consider using the Parzen window estimator (section 4.2) with a rectangular, possibly square, window function equal to the bin sizes in the previous example. If we estimated the densities at the center of all bins $b \in B$, we would get the same estimates as in the quantified method. However, if we computed the estimate of a query point $\vec{x}$ close to the edge of a bin, the Parzen window may use entropies across the grid lines in the quantification approach, , and may also *not* use some entropies within the encapsulating bin. The Parzen window estimates thus have higher overall accuracy than the quantification method, because each discrimination estimate is computed specificly for the location of $\vec{x}$.

We can make a similar argument for the kNN density estimator (section 4.4). Here, we fix the number of samples $k$ used to estimate the density in any given point, thus giving direct control of the reliability. This allows the kNN density estimator to achieve high accuracy where possible, while still respecting the reliability requirement, as specified through $k$. The equivalent quantification approach would have variable bin sizes in order to fix the number of entropies within each bin, but these estimates would still only be accurate at the center of each bin, thus the accuracy is inferior to that of the non-parametric kNN density estimator. Our continuous approach therefore has clear benefits in terms of accuracy.

Regardless of using fixed bin sizes or variable bin sizes; the optimal bin sizes and their locations depends on the nature of the classes in the dataset and the chosen property space. To make the most out of the quantification approach, the bins must reliably and accurately estimate the important regions of the true unknown discrimination function. This includes distinguishing between regions with different discriminatory ability. In practice, we can therefore argue the optimal specification of the quantification parameters requires domain knowledge, which in this context means quite detailed information about the dataset at hand, as it seems difficult to devise a general approach for choosing the optimal

quantification parameters without simply testing many combinations or by first estimating the discrimination function in the continuous domain and then fit the parameters based on the continuous estimate.

To summarise, in both the continuous and the quantification approach, we have to consider the trade-off between accuracy and reliability, but in the quantification approach we *also* have to consider how the bins resulting from quantification align with the regions of different discriminative value in property space. In practice, when using the quantification approach we would also have to tolerate some deviation from the true discriminatory ability, because the estimates are most suitable for the center of the bin. Such considerations should preferably be unnecessary in order to extract the best possible features adaptively, and the loss of accuracy is also undesirable.

# 4. Non-parametric density estimation

A natural generalisation of the approach in [5] is to use the entropy values in their continuous nature. In doing so, their discriminative value can be estimated continuously, thus enabling higher accuracy of the adaptive features. However, we have to give special attention to the reliability of the estimates. If not, performance is likely to suffer.

In this section we will investigate various ways of generalising to using the continuous values directly. We will first introduce the basic idea for nonparametric estimation of probability density functions (pdfs), then we will look at three specific methods following from this idea. In particular, we will be presenting the Parzen window, the k-nearest neighbour and finally a combination of these two which we refer to as simply Parzen-kNN. The following introduction is inspired by Duda et al. [3].

## 4.1 Theory

We will now describe a way to estimate the probability density function of a random variable. Let the probability density function of interest be called $f$. The probability $P$ of the vector $\vec{x}$ falling within a region $R$ of the sample space can be expressed as

$$P = \int_R f(\vec{x})\, d\vec{x} \tag{4.1.1}$$

Suppose now that $n$ samples, $\vec{x}_1, ..., \vec{x}_n$ are drawn independently from a random variable with the probability density function $f$. Since $f(\vec{x})$ gives the relative likelihood of taking on the given value $\vec{x}$, we can attempt to use our samples $\vec{x}_1, ..., \vec{x}_n$ to estimate $f$. Now, either a sample falls into the region $R$ with probability $P$ or else it does not with probability $1 - P$. It is clear that $k$ of these $n$ samples falls into $R$ with probability following the binomial distribution

$$P_k = \binom{n}{k} P^k (1 - P)^{n-k} \tag{4.1.2}$$

with an expected value of

$$E\left[P_k\right] = nP \tag{4.1.3}$$

Note that any binomial distribution, $Bin(n, p)$, is the sum of $n$ independent Bernoulli trials, $Bern(p)$, each having the same probability $p$. Given $X \sim Bin(n, p)$, then if $n$ is large, $X/n$ becomes a reasonable estimator of $p$. For our case, this means that $P_k/n$ becomes a reasonable estimator of $P$, that is, if we denote the estimator of $P$ as $\hat{P}$, then

$$\hat{P} = \frac{P_k}{n} \tag{4.1.4}$$

By conveniently naming our observation of $P_k$ as $k$, we are assuming that for large $n$

$$P = \frac{k}{n} \tag{4.1.5}$$

Furthermore, if we assume the region $R$ is small enough for $p$ not to vary much within it, we can write

$$\int_R f(\vec{x})\, d\vec{x} \simeq f(\vec{x})V \tag{4.1.6}$$

where $V$ is the volume of this region $R$. From this equation and equation 4.1.1 when using the estimator of $P$ in equation 4.1.4, we then arrive at the following estimator for the density function $f$

$$\hat{f}(\vec{x}) = \frac{P_k/n}{V} \tag{4.1.7}$$

Because we want to estimate the density at an input vector $\vec{x}$, a practical thing to do is letting the region adapt to that input. Let the adapted region be denoted by $R_{\vec{x}}$. Typically, $R_{\vec{x}}$ is centered at $\vec{x}$. There exists [3] three well-known necessary conditions for $\hat{f}(\vec{x})$ to converge to $f(\vec{x})$. These conditions says the following; as $n$ approaches infinity, the volume of the region should approach zero while the number of samples falling into the region approaching infinity, however, it must do so at a rate such that the number of samples $k$ falling into $R_{\vec{x}}$ is negligible compared to the total number of samples $n$. Since we have a limited number of samples $n$, there are additional considerations to be made, apart from the these conditions of convergence. In particular, if we let $V$ become too small we run the risk of getting no samples inside a region even though the samples are relatively dense in that location. In practice we need to find a good balance between small $V$ and large $k$ in order to produce both relatively accurate and relativley reliable density estimates for the intended use. In the following two sections we will look at two approaches, which are commonly used for estimaton of class conditional density, and then a combination of these two.

## 4.2 Parzen window density estimation

In the Parzen window density estimator, we are essentially fixing the volume and structure of the region. The density estimates as computed by the Parzen window is

$$\hat{f}(\vec{x}) = \frac{1}{nV} \sum_{i=1}^{n} \varphi\left(\frac{\vec{x} - \vec{x}_i}{h}\right) \tag{4.2.1}$$

where $\varphi$ is the so-called *window function*, and $h$ the *window width* or *bandwidth*. The window function makes it so that each sample contributes according to its *distance* from the point of interest, $\vec{x}$. In this thesis we will be using a *Gaussian window function*, and thus we let $V = 1$ since the window is already normalised. In the d-dimensional case we then have

$$\hat{f}(\vec{x}) = \frac{1}{n\sqrt{(2\pi)^d|H|}} \sum_{i=1}^{n} \exp\left(-\frac{1}{2}(\vec{x} - \vec{x}_i)^T H^{-1}(\vec{x} - \vec{x}_i)\right) \tag{4.2.2}$$

where $H$ is the non-singular bandwidth matrix, and $\vec{x}$ a d-dimensional vector. As we can see, the role of matrix $H$ corresponds to that of the covariance matrix $\Sigma$. In the one-dimensional case this equation reduces to

$$\hat{f}(x) = \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x - x_i}{h}\right)^2\right) \tag{4.2.3}$$

giving a bandwidth matrix with only a single element, $h$.

Note the computational consequence of using a Gaussian window function is that we have to consider all the samples $\vec{x}_1, ..., \vec{x}_n$ when calculating the estimate for a given query location $\vec{x}$.

## 4.3 Bandwidth selection

One way to select the bandwidth is to look at and assess density estimate plots for different bandwidths. This interactive approach requires some intuition about what the true distribution looks like, but in one or two dimensions visualization is relatively easy. Another approach is to use *rule-of-thumb* bandwidth selection, which gives a formula arising from the optimal bandwidth for a reference distribution [4, p.73]. Such a formula is useful when the true distribution resembles some known distribution. However, if the true distribution is sufficiently similar to a known distribution whose parametrization is known, then we might be better off with parametric estimation.

Suppose the reference distribution is the d-variate normal distribution $\mathcal{N}_d(\mu, \Sigma)$ with $\Sigma = diag(\sigma_1^2, ..., \sigma_d^2)$. We can then use the following rule-of-thumb for the bandwidth matrix $H = diag(h_1^2, ..., h_d^2)$,

$$\hat{h}_j = n^{-1/(d+4)}\hat{\sigma}_j \tag{4.3.1}$$

This rule-of-thumb is known as *Scott's rule* [4, p.73].

## 4.4 kNN density estimation

In the kNN density estimator we are fixing the number of samples $k$ that falls into the region. The volume of the region must therefore adapt to the location $\vec{x}$ of interest. This gives rise to some interesting properties. Concretely, at locations densely populated by samples, the volume can be relatively small. While at sparsely populated locations the volume must be relatively large. The accuracy of the estimates are therefore high where possible, while the reliability of the estimates is the same everywhere. In contrast, the Parzen window has

a fixed volume and thus fixed accuracy while its reliability depends on how densely populated the region is.

To calculate the actual kNN density estimate we will use the estimator from equation 4.1.7, that is;

$$\hat{f}(\vec{x}) = \frac{P_k/n}{V}$$

and take the region to be a hypersphere centered at $\vec{x}$ with radius $r$. Its volume is thus $2r$ in the one-dimensional case, and $\pi r^2$ in the two-dimensional case. All points whose Euclidean distance to $\vec{x}$ is larger than $r$ is outside the region. Such a crisp region boundary typically creates estimates that fluctuates more, so they are not as smoothed as the estimates produced by the Parzen density estimator with a Gaussian window function.

As a comment on computational complexity, to calculate the estimate for a given query location $\vec{x}$ we now only have to consider a subset $K \subseteq \{\vec{x}_1, ..., \vec{x}_n\}$ where $|K| = k$. The crisp region of the kNN density estimator is thus better in terms of computation, provided that we pre-organise all $n$ samples in a data structure which is suitable for fast lookup of these $k$ samples.

## 4.5 Combining two principles: Parzen-kNN density estimation

Here we will present a hybrid between the two density estimators discussed so far. We will refer to the hybrid as the Parzen-kNN density estimator. The idea is to fix the number of samples $k$ that falls into the region, and then use the radius $r$ of this region as the bandwidth of the Parzen window with a Gaussian window. Note that the bandwidth is then locally dependent, i.e. varies between different locations of $\vec{x}$. If we take the region to be a hypershere, we get a bandwidth matrix of the form $H = diag(r^2, ..., r^2)$. In the one-dimensional case we see this as the bell-shaped Gaussian centered at $x$, with a standard deviation $r$. No assumptions are being made. However, in the n-dimensional case, we are in fact making the assumption of independent features with equal variances. The reason for this assumption is not belief of it being true, but rather that it seems questionable that a *full* covariance matrix estimated *locally* should be used to evaluate the contribution of a sample at a more distant location, e.g. far outside the local region used to estimate the covariance matrix. When assuming independent features we omit the directional component of the covariance matrix and thus it only represents the idea of local population density, i.e. is used to dictate the tradeoff between accuracy and reliably by using the local density of samples.

In terms of computation, this approach is the most costly of the presented non-parametric density estimators. We have to do the same amount of work as the regular Parzen method with a Gaussian window, and in addition, we must perform a lookup of the $k$ nearest neighbours of the query point $\vec{x}$ in order to compute the radius $r$.

# 5. Classification and evaluation

In this thesis we are studying a dataset of 134 patients, each of which are associated with good or poor prognosis, see section 1.1. In other words; our data is *labelled* and we will use it to train and test a classifier. Specifically, given a patient whose prognosis group is unknown, we want to reliably estimate whether the patient has good or poor prognosis. This is the classifier task, and it conforms to the well known setting of *supervised learning*.

We will now describe how the feature values of the patient is computed from its associated cell nucleus images using the discrimination functions defined in section 3.1. Recall that these two functions estimate the discriminative value of a single cell nucleus image with respect to the bright and the dark segmentation class. Given some patient whose cell positions in CSDE-space are $\vec{x}_1, ..., \vec{x}_m$ in a segmentation class, we let the patient be represented by the arithmetic mean of the feature values of its cells. Thus, a patient is represented by the point $(b', d')$ in the feature space, where

$$b' = \frac{1}{m} \sum_{i=1}^{m} f_b(\vec{x}_i|c) \tag{5.0.1}$$

$$d' = \frac{1}{m} \sum_{i=1}^{m} f_d(\vec{x}_i|c) \tag{5.0.2}$$

and where the functions $f_b$ and $f_d$ are the discrimination functions. Note that each cell may now contribute differently to the feature values of its patient, and it does so in accordance with our idea of discriminative value. This is desirable because cells are not homogenous in terms of cancer [5, p.14-16]. In other words; all the cell nucleus images of a patient could include non-cancerous cells along with cells associated with the cancer itself, and if the classifier considers these cells equal, performance is likely to suffer.

Representing a patient using these features, we then train a *linear discriminant classifer* (LDC) to find an optimal decision boundary for our classification task. We could therefore say that the choice of *analytical unit* is the patient.

Consider the alternative choice in which we choose the analytical unit to be the cell. Then we would have to allow ourselves to assume that a single cell can have cancer, which is not normal to claim [5, p.14-16]. Another problem of classifying on the cell level, is that the cells can not be considered as independent [10].

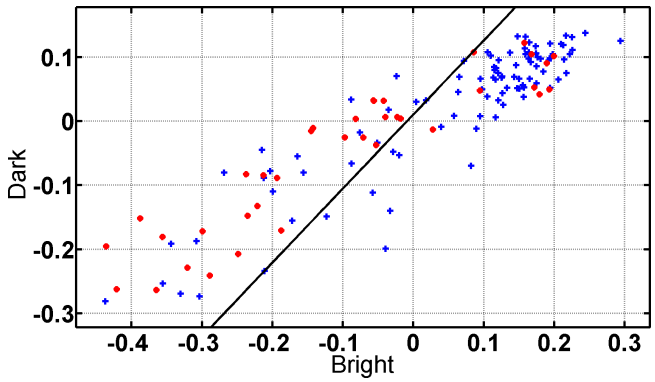


*Figure 5.0.1: A linear discriminant classifier trained using all 134 patients and the Parzen-kNN density estimates of CSDE-space. The parameters are* $c = 0.1$ *and* $k = 2000$*. Results are deferred to section 6.*

When training the classifier, we will use equal *a priori* probabilities. This is because we are equally interested in classifying each prognosis groups correctly, regardless of the sizes of the groups, as opposed to equally interested in classifying each patient correctly.

An example of how the feature space and decision boundary may look is given in figure 5.0.1. This is a *resubstitution plot* in which all patients are used for training and testing. The classification may therefore be too optimistic and "unfair", since the performance is estimated using samples which the classifier has seen during training. We will refer to this particular way of estimating the classifier's performance as *resubstitution*. While inappropriate in most situation, it can aid us in choosing parameters for computationally expensive density estimators.

## 5.1 Bootstrapping

Estimating classifier performance using resubstitution is not appropriate because it doesn't measure generalisation ability to novel patients, which is what we wish to estimate. To estimate the classification performance of unseen patterns, we will use a type of *statistical bootstrapping*.

In our bootstrapping method, the patients are randomly partitioned into a training set and a test set. We will use the fairly common ratio that puts 70% of the patients into the training set and the remaining 30% into the test set. The discrimination functions $f_b$ and $f_d$ will be computed using the density estimates acquired by only using the patients in the training set. Then all patients are represented by the arithmetic mean of their cells feature values, and the LDC classifier is fitted using only the patients in the training set. Now we use this trained decision boundary to classify the patients from the test set. This counts as a single bootstrap. However, we require many such bootstraps to give a more reliable and complete estimate of classifier performance. This is discussed in the following section.

| Quantity | Definition | Format |
|---|---|---|
| CCReq | $\frac{Specificity+Sensitivity}{2}$ | ? % [? %, ? %] |
| CCR | $\frac{TP+TN}{TP+FP+TN+FN}$ | ? % [? %, ? %] |
| Specificity | $\frac{TN}{TN+FP}$ | ? % [? %, ? %] |
| Sensitivity | $\frac{TP}{TP+FN}$ | ? % [? %, ? %] |

*Table 5.2.1: The definition of the four quantities used in this study to describe a classifier's performance. The abbreviations are as following; correct classification rate (CCR), correct classification rate assuming equal a priori probabilities (CCReq), true positive (TP), false positive (FP), true negative (TN) and false negative (FN). Positive indicates classified as poor prognosis, while negative indicates classified as good prognosis. In the format column, the leftmost numbers are the estimated expected performances, while the numbers contained in square brackets gives the estimated 95 % two-sided PI.*

## 5.2 Reporting classification results

In this section we will present how the results are reported. For every bootstrap result we calculate four performance quantities from the confusion matrix; *CCReq*, *CCR*, *specificity* and *sensitivity*. The definitions of these quantities and the format in which they are presented are listed in table 5.2.1. When comparing the performance of different classifiers, we will use the expected CCReq as the primary measurement, as we consider it the best single descriptor of performance in this context. The reason for this is the same as why we use equal *a priori* probabilities, i.e. that we are equally interested in classifying each prognosis groups correctly.

In this thesis the bootstrap is run five-hundred times, from which the expected value and the 95% *prediction interval* (PI) of each performance quantity is reported. The 95 % prediction interval is an estimate of an interval in which the performance quantity of 95% of all future classifiers will fall, given that the classifier is based on the same features and classification method, and trained using $97 * 0.7 \approx 68$ patients with good prognosis and $37 * 0.7 \approx 26$ patients with poor prognosis.

# 6. Results and discussion

We will in this section present the classification results for a variety of scenarios. We present the results of our non-parametric density estimators introduced in section 4 as well as some parametric density estimators, both operating in CSDE-space and CSDEsum-space. In addition, we discuss how our results compare to a *benchmark result*, which is the best texture analysis result achieved prior to this work. These benchmark results are presented in table 6.0.1 and 6.0.2, and they quantify CSDEsum-space and CSDE-space, respectively, using five levels per integer value. As mentioned in section 1.1, the benchmark is the result of our reimplementation of the most promising approach in [5] and has been re-evaluated using the updated prognosis groups.

One issue we are facing with all our methods is the choice of the parameter $c$ in the discrimination functions. Some of the methods also have an additional parameter $k$. Ideally we would perform a search over the parameter space, run five-hundred bootstraps for each parameter combination and record the expected performance quantities. Then we would represent this information in a plot, call it the *bootstrap performance plot*, and try to spot any appearing trends about which regions of this parameter space are the most promising. The trend reveals whether a bootstrap result is good simply because of bootstrap variation, or if there is a real reason for it. Then, if necessary, we would refine our search in the most promising region in order to more precisely locate the optimal parameter set. Unfortunately this approach is only practical in some scenarios, because the computational burden is too extensive for some of the methods. For all the parametric methods and the non-parametric kNN method, such a search only takes a reasonable amount of time. However, for the Parzen window and Parzen-kNN methods, every single bootstrap takes a significant amount of time, making fine parameter search too slow. In these cases, what we will do is manually assess whether parameters $c$ and $k$ are reasonable by looking at the resulting resubstitution density estimates and discrimination functions.

Now follows a brief description of considerations we have taken when choosing parameters manually. We have chosen $c$ such that the noise due to unreliable estimates in sparsely populated regions is supressed. This is the effect of choosing a relatively large value for $c$, as was discussed in section 3.1. The most important aspect of choosing $k$ is its significant influence on the discrimination function. In particular, an increase of k increases the reliability of the estimates and decreases their accuracy. Thus we wish $k$ to be the smallest possible value where the discrimination function seems to be reliably estimated. Another observation aiding the manual choice of $k$ is that the kNN density estimator is highly sensitive to its value, while Parzen-kNN is somewhat less sensitive due to its smooth Gaussian region. Additional effects of the parameters $c$ and $k$

| | |
|---|---|
| CCReq | 71.8 % [59.4 %, 85.1 %] |
| CCR | 72.2 % [60.0 %, 82.5 %] |
| Specificity | 72.7 % [55.2 %, 86.2 %] |
| Sensitivity | 70.9 % [45.5 %, 100.0 %] |

*Table 6.0.1: The benchmark classification results achieved when using the updated prognosis groups and the CSDEMsum features in [5] which use five levels per integer value.*

| | |
|---|---|
| CCReq | 69.1 % [53.3 %, 82.3 %] |
| CCR | 70.9 % [57.5 %, 82.5 %] |
| Specificity | 73.1 % [58.6 %, 86.2 %] |
| Sensitivity | 65.1 % [36.4 %, 90.9 %] |

*Table 6.0.2: The benchmark classification results achieved when using the updated prognosis groups and the CSDEM features in [5] which use five levels per integer value.*

on the discrimination function, and their interpretations, are interleaved in the following sections.

Every scenario we present is accompanied by its best achieved classification result in a table along with the parameters that achieved that result. In the scenarios where many parameters were tested we also present the bootstrap performance plot as mentioned above. It is also interesting to investigate the resubstitution density estimates and the resubstitution discrimination functions. However, in CSDEsum-space there are four resubstituion plots for every parameter combination, while in CSDE-space there are six.

For the convenience of reading, these plots are gathered in appendix A, even though we sometimes refer to them frequently.

## 6.1 CSDEsum-space

In this section we look at the results when our density estimates are in CSDEsum-space. As earlier mentioned this is a projection of CSDE-space onto the identity line yielding a space much more densely populated by samples. In projecting, we lose information if a change in the spatial entropy is not identical as the same change in the grey level entropy, but on the other hand we also make it easier for our non-parametric density estimators to provide both reliable and accurate estimates. The results in this section should be compared to the CSDEMsum benchmark result in table 6.0.1.

### 6.1.1 Parzen

The Parzen density estimator was run for a limited number of values for $c$, and, as mentioned in section 4.3, the chosen bandwidth is the optimal bandwidth under the assumption of normally distributed samples, with a diagonal covariance matrix. The results are summarized in figure 6.1.1. We can see the classifier performance is peaking, although not significantly, at $c = 0.1$. At the peak we have an expected CCReq of 72.9%, which is a small improvement over the benchmark result of 71.8%. However, the difference is so small that we should

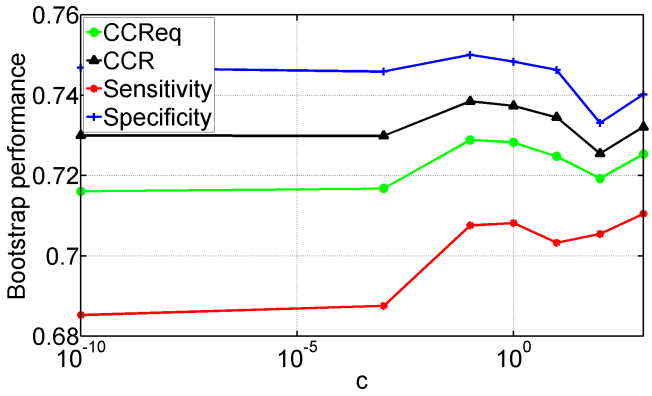


*Figure 6.1.1: Bootstrap performance when using the Parzen density estimator in CSDEsum-space. It was run for* $c \in \{10^{-10}, 0.001, 0.1, 1, 10, 100, 1000\}$. *The best expected CCReq is achieved for* $c = 0.1$.

| Parzen (c = 0.1) | |
|---|---|
| CCReq | 72.9 % [58.9 %, 86.8 %] |
| CCR | 73.9 % [62.5 %, 85.0 %] |
| Specificity | 75.0 % [62.0 %, 89.7 %] |
| Sensitivity | 70.8 % [45.5 %, 90.9 %] |

*Table 6.1.1: The classification results when using the CSDEsum density estimates from Parzen, and using the parameter c which achieved the best expected CCReq.*

not rule out that it is caused by bootstrap variations. All of the expected means and prediction intervals for $c = 0.1$ can be found in table 6.1.1.

Looking at the Parzen density estimates in figure A.1.1, we can see they are rather smooth. This quality is partly caused by our choice of window function; the Gaussian. Notice also how the density estimates for the good prognosis group, $\omega_0$, strongly resembles a narrow normal distribution, while in the case of the poor prognosis group, $\omega_1$, one may suspect that there is two normally distributed components which are close together. This is much more pronounced for the bright feature, $\lambda_0$, but it is also noticable for the dark feature, $\lambda_1$. In the discrimination functions (figure A.1.1) this causes the negative valleys to become even deeper at the locations where this additional component contributes.

The density estimates does reinforce the assumption of normality for $\omega_0$, but much less so for $\omega_1$, where there could be two components.

### 6.1.2 kNN

We tested the kNN density estimator for a large number of parameters $c$ and $k$ in the bootstrap method. First we performed a coarse search (see figure 6.1.2) in which we identified $c = 0.1$ as the best value for $c$, then we fixed $c$ at this value, and increased the granularity of $k$. The outcome of the finer search is presented in the bootstrap plot in figure 6.1.3, and the best result of these are presented in table 6.1.2. We have also included a plot showing the reubstitution plot of a trained LDC decision boundary when using the paramteres that gave the best results, see figure 6.1.3. Notice how the majority of good prognosis

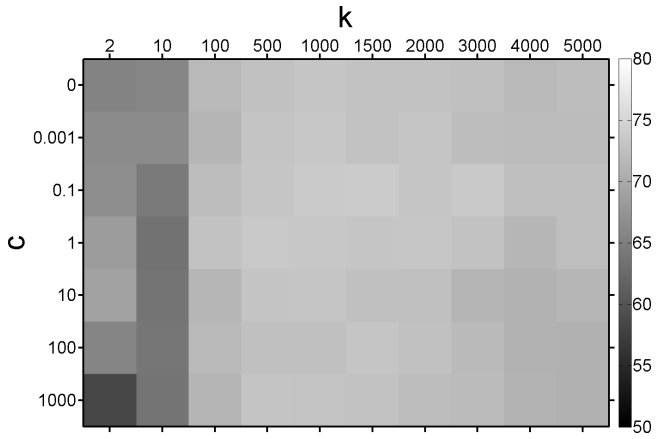


*Figure 6.1.2: Coarse parameter search showing the bootstrap's expected CCReq when using the kNN density estimator in CSDEsum-space. The greyscale display range is specified as [50, 80]. The acutal values are in the range [58.4, 73.8].*

| kNN (c = 0.1, k = 1350) | |
|---|---|
| CCReq | 74.7 % [61.8 %, 87.5 %] |
| CCR | 75.4 % [65.0 %, 87.5 %] |
| Specificity | 76.3 % [62.1 %, 89.7 %] |
| Sensitivity | 73.1 % [45.5 %, 90.9 %] |

*Table 6.1.2: The classification results when using the CSDEsum density estimates from kNN, and using the parameters c and k which achieved the best expected CCReq.*

patients are clustered in the upper right area of the figure.

The bootstrap plot shows that all its runs produced an expected CCReq above 73%. We can therefore be confident that we have improved upon the CSDEMsum benchmark result in table 6.0.1.

If we momentarily disregard the noise of the density curves produced by the kNN density estimator (figure A.1.2) and only consider the domain of those two curves in which $f \geq 2$, the curves are actually strikingly similar to those of the Parzen density estimator. For the remaining part of the domain, in which $f < 2$, there is however a significant difference. In particular, the tails of the kNN density estimates are much longer, meaning the estimates for $f < 2$ are much higher than those produced by the Parzen density estimator.

To understand why this happens, notice that as the hypersphere moves away from a more densely populated region, the radius $r$ must increase to keep the number of samples within the hypersphere constant. In doing so, the hypersphere expands into the more dense region from which it is leaving, thus increasing the value of the density estimate of the location it is currently in. In short; as the window becomes wider, new samples are encountered at a rate faster than the window width increases.

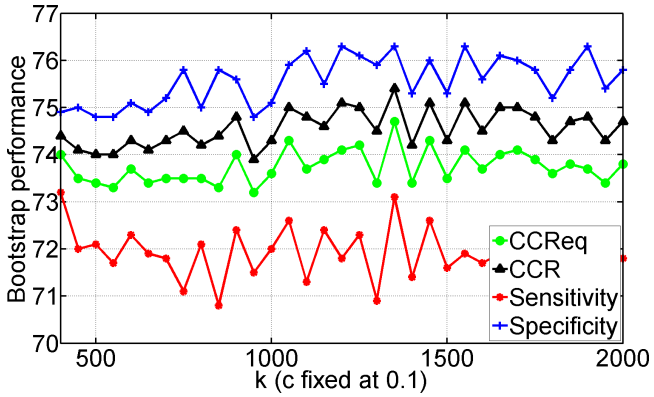


*Figure 6.1.3: Bootstrap performance when using the kNN density estimator in CSDEsum-space. It was run for $k \in \{400, 450, 500, ..., 1950, 2000\}$, while $c$ was fixed at 0.1. The best expected CCReq is achieved for $k = 1350$.*

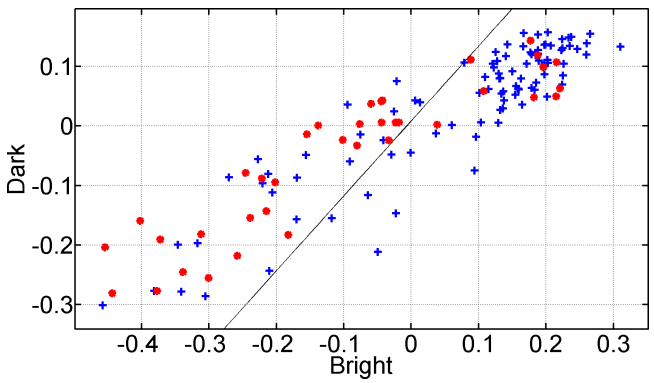


*Table 6.1.3: A resubstitution plot showing a linear discriminant classifier trained using the density estimates from kNN with the parameters $c = 0.1$ and $k = 1350$ which achieved the best expected CCReq.*

| Parzen-kNN (c = 0.1, k = 1000) | |
|---|---|
| CCReq | 73.7 % [59.6 %, 86.2 %] |
| CCR | 74.9 % [62.5 %, 85.0 %] |
| Specificity | 76.4 % [62.1 %, 89.7 %] |
| Sensitivity | 71.1 % [45.5 %, 90.9 %] |

*Table 6.1.4: The classification results when using the CSDEsum density estimates from Parzen-kNN, and using the parameters c and k which achieved the best expected CCReq.*

### 6.1.3 Parzen-kNN

The Parzen-kNN density estimator was tested for a much smaller set of parameters than the kNN density estimator, due to its relatively high computational cost. In both the kNN and the Parzen density estimators, the value $c = 0.1$ was identified as reasonable, and considering that Parzen-kNN is a combination of those two methods, we found it reasonable to expect this value of $c$ to work well, also in this case. The value was therefore fixed at $c = 0.1$, while we tried a wide range of values for $k$, see figure 6.1.4.

We observe that the density estimates produced by the Parzen-kNN density estimator shares characteristics with both the Parzen and the kNN density estimators. Conceretely, only a narrow Gaussian window function is needed to include $k$ samples in densely populated regions. This makes local variation more pronounced in these regions (if there is any), and the effect is observable in both the estimated densities for the good prognosis group, $\omega_0$, in figure A.1.3 in that the curves are more "wiggly". In contrast, the estimated densities for the poor prognosis group, $\omega_1$, are much more smooth (same figure). We know there are far less samples associated with poor prognosis than with good prognosis in our dataset, and a reasonable interpretation is thus that the Parzen-kNN density estimator behaves more like the kNN density estimator in dense regions, and more like the Parzen density estimator in sparse regions. A high number of samples is required for the Gaussian window to become so narrow, that it starts to behave somewhat similarly to the crisp region of the kNN density estimator. Whether this is an advantage or not depends on the situation. In this study, however, we find that the kNN density estimator performs better than the Parzen-kNN density estimator.

We also observe that the resubstitution density estimates in sparse regions have numeric higher value than in the case of the kNN and the Parzen density estimator, even though Parzen-kNN used $k = 1000$ and kNN used $k = 1350$. Recall that the Parzen-kNN window function is a Gaussian that requires $k$ samples to lie within a radius of one standard deviation. Its window is therefore much wider than the kNN hypersphere, even though $k$ is the same, and we think this is the explanation of this observation.

The best expected mean of the Parzen-kNN density estimator's CCReq is 73.7%, see table 6.1.4. This is almost exactly the average of the expected CCReq's for the Parzen and the kNN density estimators.

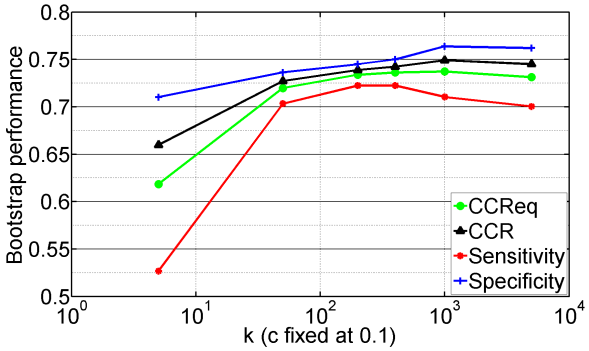


*Figure 6.1.4: Bootstrap performance when using the Parzen-kNN density estimator in CSDEsum-space. It was run for $k \in \{5, 50, 200, 400, 1000, 5000\}$, while $c$ was fixed at 0.1. The best expected CCReq is achieved for $k = 1000$.*

### 6.1.4 Normal distribution

The bootstrap plot of the normal distribution density esimates (figure 6.1.5) shows our first encounter with sensitivity becoming higher than specificity. From the definition of sensitivity and the convention of letting "positive" denote poor prognosis, we can see that the sensitivity measures the fraction of correctly classified patients with poor prognosis. Conversely, specificity measures the fraction of correctly classified patients with good prognosis. Given the skew in our dataset towards good prognosis, this is why observe the expected CCReq is larger than the expected CCR; the classifier got better at classifying poor prognosis patients, but at the cost of the other prognosis group which has a significantly larger number of patients in it.

The best expected CCReq of 73.1% was achieved for $c = 10$, and the detailed results are shown in table 6.1.5. It is interesting to note that the performance of the normal density estimates is highly dependent on $c$, considering we introduced $c$ to compensate for low reliability in sparse regions when using non-parametric density estimates. What we observe as we increase $c$ is that the positive region and the negative region of the discrimination function approaches each other, but only up to a certain point, and which point this is depends on how large the values of the density estimates are to begin with. A side effect is that the transition between the prognosis groups becomes steeper or more abrupt. It also reduces the total range of entropies that has significant discriminative value, and thus the more extreme values are supressed. These effects can be seen by plotting the discrimination curve for increasing values of $c$, and judging from the bootstrap plot, the classifier has exploited this effect for the benefit of increasing the sensitivity.

### 6.1.5 Gaussian mixture model

When looking at the resubstitution density plots when using either of our non-parametric estimation methods, we observe a slight tendency of bimodality for the poor prognosis densities. We tried to model this using a *Gaussian mixture model* (GMM) in which the model assumed a single component for good prognosis estimates, but *two* components for poor prognosis estimates. We also tested

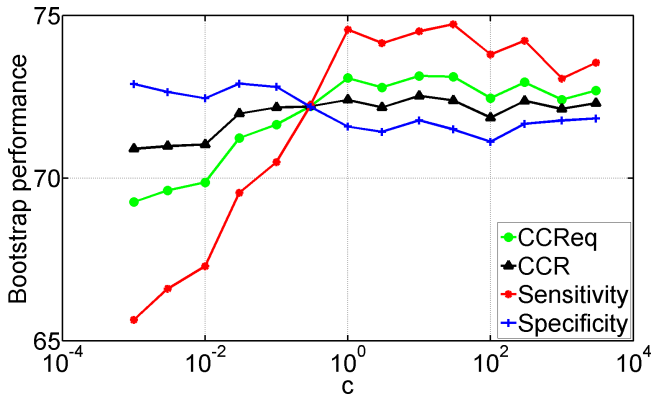


*Figure 6.1.5: Bootstrap performance when using the parametric normal density estimator in CSDEsum-space. It was run for* $c \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000, 3000\}$. *The best expected CCReq is achieved for* $c = 10$.

| Normal distribution (c = 10) | |
|---|---|
| CCReq | 73.1 % [59.4 %, 85.1 %] |
| CCR | 72.5 % [60.0 %, 85.0 %] |
| Specificity | 71.8 % [58.6 %, 86.2 %] |
| Sensitivity | 74.5 % [54.5 %, 90.9 %] |

*Table 6.1.5: The classification results when using the CSDEsum density estimates from a fitted normal distribution, and using the parameter c which achieved the best expected CCReq.*

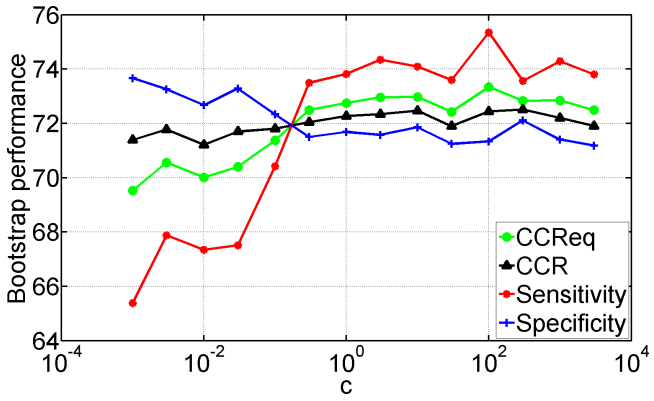


*Figure 6.1.6: Bootstrap performance when using the parametric GMM density estimator in CSDEsum-space and assuming a shared $\Sigma$. It was run for $c \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000, 3000\}$. The best expected CCReq is achieved for $c = 10$.*

| GMM (shared $\Sigma$, c = 100) | |
|---|---|
| CCReq | 73.3 % [60.7 %, 85.7 %] |
| CCR | 72.4 % [60.0 %, 85.0 %] |
| Specificity | 71.3 % [55.2 %, 86.2 %] |
| Sensitivity | 75.3 % [54.5 %, 100.0 %] |

*Table 6.1.6: The classification results when using the CSDEsum density estimates from a fitted GMM, and using the parameter c which achieved the best expected CCReq.*

two different assumptions on the covariance matrix in the GMM. The results of these two assumptions are presented together in this section.

The first assumption was that the two components of poor prognosis had the same covariance matrix, the last assumption was they needed not have the same covariance matrix. The best expected CCReq's (see figures 6.1.6 and 6.1.7) were 73.3% and 73.2%, respectively, and their full results are available in table 6.1.6 and 6.1.7 respectively.

If we look at the parameters of the density curves for good prognosis, $\omega_0$, we can see they are all the same for the normal density estimator and both our GMM density estimators. This is completely as expected as they are all fitted using the same model. However, for the poor prognosis group, $\omega_1$, the GMM gives two normal curves whose means are very close to each side of the mean that we got when fitting a single normal component. Overall, the density curves are very similar to those of the parametric normal denstity estimator, and instead of getting a curve with two visible components, as was the case when using a non-parametric density estimator, we got something looking like a single curve, only a little wider.

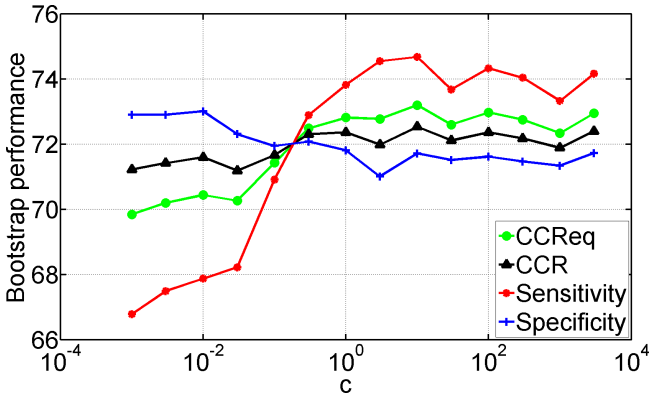


*Figure 6.1.7: Bootstrap performance when using the parametric GMM density estimator in CSDEsum-space and not restricting $\Sigma$. It was run for $c \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000, 3000\}$. The best expected CCReq is achieved for $c = 10$.*

| GMM (non-shared $\Sigma$, c = 10) | |
|---|---|
| CCReq | 73.2 % [59.4 %, 85.7 %] |
| CCR | 72.5 % [60.0 %, 85.0 %] |
| Specificity | 71.7 % [58.6 %, 86.2 %] |
| Sensitivity | 74.7 % [54.5 %, 100.0 %] |

*Table 6.1.7: The classification results when using the CSDEsum density estimates from a fitted GMM, and using the parameter c which achieved the best expected CCReq.*

## 6.2 CSDE-space

Here we present the results for discrimination functions based on density estimation in CSDE-space. The results in this section should be compared to the benchmark result in table 6.0.2. This is because our continuous approach can be seen as a generalisation of the quantification approach, and our hypothesis is that our continuous approach may obtain more reliable and accurate estimates of the discrimination value (see section 3.3) which may result in better classification performance. To test this, we should compare the approaches in the same domain, i.e. in the continuous CSDE-space and the quantification of this space into a CSDEM.

When looking at the discrimination functions produced in this section we notice that most of them are similar to each other (there are some exceptions). In particular, for both functions $f_b(\vec{x}|c)$ and $f_d(\vec{x}|c)$ we observe a *separating margin* between the positive region (red) and negative region (blue). If we simplify and think of the margin as a straight line, it would often times be oriented roughly orthogonally onto the identity line, and its width is dependent on the parameters $c$ and $k$. This observation tells us that the identity line is reasonable line to project onto. However, whether this is the optimal projection is not known.

Concretely, by increasing $k$ the two density estimates, from which the sicrimination function is calculated, are smoothed more strongly, which increases the size of the region where the estimated probabilities are approximately equal. This results in the peaks of the positive and the negative regions in the discrimination function to become more separated. This is immediately appearant if we rewrite the discrimination function using the quotient rule for logarithms

$$\log \frac{x}{y} = \log x - \log y$$

The result of increasing $k$ is thus a longer and less abrupt transition between the positive and negative peak, or in other words; a wider margin.

On the other hand, if we increase $c$, the positve and negative regions slowly approach each to a certain point depending on the maximum values of the density estimates. To see this, notice that $c$ affects the density estimates of sparse regions (low numerical value) relatively more than it affects density estimates of dense regions (high numerical value). This yields a somewhat sharper transition or a thinner margin. Note that in our case the parameter $c$ is unable to equalize the effect that $k$ has on the margin. The reason is that as we increase $k$, we lower the maximum values of the density estimates, thus reducing the ability of $c$ to affect the margin.

The concept of a margin is interesting because in other known classification methods, such as the *support vector machine* (SVM), the concept of margin width can be used to provide an upper bound for generalization error. However, this is not directly applicable in our case, as our margin affects the discrimination function and not the decision boundary, but we simply notice that the margin decreases the significance or contributing ability of points in CSDE-space that are close to the boundary between the positive and negative region. Having some control of this margin and its smoothness is beneficial because the best transition between strong evidence for poor prognosis and strong evidence for good prognosis is unknown.

| Parzen (c = 0.3) | |
|---|---|
| CCReq | 70.0 % [57.2 %, 84.5 %] |
| CCR | 71.0 % [60.0 %, 82.5 %] |
| Specificity | 72.3 % [58.6 %, 89.7 %] |
| Sensitivity | 67.6 % [36.4 %, 90.9 %] |

*Table 6.2.1: The classification results when using the CSDE density estimates from Parzen with a diagonal bandwidth matrix.*

### 6.2.1 Parzen

Due to the high computational cost of the Parzen density estimator when used in a two-dimensional space like CSDE, the parameter $c$ was only selected manually. To select the parameter we looked at which values for $c$ produced reasonable density estimates, and we also used our experience of which values of $c$ had worked well for other methods or settings. The choice was $c = 0.3$, for which an expected CCReq of 70.0% was achieved. The full results are available in table 6.2.1. The resubstitution density estimates and discrimination functions are shown in figure A.2.1. Notice that the positive region of the dark discrimination function is significantly larger than the three other regions in the discrimination plots.

We also observe that the separating margin between the positive and negative region in the discrimination plot is aligned roughly orthogonally on the identity line. If we look closely, we do notice a small difference in margin orientation for the bright and dark discrimination functions. Concretely, in the dark discrimination function, the "separating margin" is somewhat less orthogonal than in the bright case.

### 6.2.2 kNN

For the kNN density estimator we fixed $c = 0.1$ after assessing the resulting resubstitution plots from a range of values. We do note that the performance of the kNN density estimator drops significantly in CSDE-space compared to CSDEsum-space. The bootstrap perfomance plot (figure 6.2.1) shows that the decline in performance is consistent and not coincidental. When looking at the resubstitution density plots in figure A.2.2 we can see they have the most complex shapes among our density plots in CSDE-space, and the discrimination functions are thus also complex (same figure). If we try increasing $k$ to produce more smooth density estimates, we observe in the bootstrap plot, a further drastic performance drop. After looking at the produced plots for $k$ from 1000 through 5000, we realised that as we increase $k$, the estimated densities in the sparse regions rises at a higher rate in the case of poor prognosis than in the case of good prognosis. The result is that for high values of $k$, there is no longer a close to linear separating margin, but rather the negative region begins to surround the positive region, and for $k = 5000$ it is completely surrounded. Note that we did observe a corresponding effect in CSDEsum-space, in the discrimination curves of the kNN density estimator (figure A.1.2) and the Parzen-kNN density estimator (figure A.1.3), yet the performance of those two density estimators were consistently better than the Parzen density estimator (also in CSDEsum-space) for which this surrounding of the positve region was not observed. If

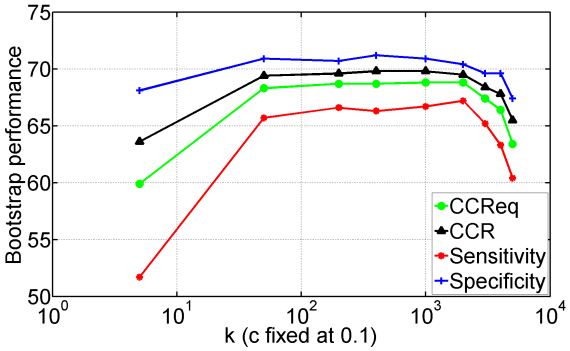


*Figure 6.2.1: Bootstrap performance when using the kNN density estimator in CSDE-space. It was run for $k \in \{5, 50, 200, 400, 1000, 2000, 3000, 4000, 5000\}$, while $c$ was fixed at 0.1. The best expected CCReq is achieved for $k = 1000$.*

| kNN (c = 0.1, k = 1000) | |
|---|---|
| CCReq | 68.8 % [53.8 %, 80.6 %] |
| CCR | 69.8 % [57.5 %, 80.0 %] |
| Specificity | 70.9 % [55.2 %, 86.2 %] |
| Sensitivity | 66.7 % [36.4 %, 90.9 %] |

*Table 6.2.2: The classification results when using the CSDE density estimates from kNN, and using the parameters c and k which achieved the best expected CCReq.*

the performance drop of the kNN density estimator in CSDE-space is caused by the surrounding of the positive region, then it must be due to the change in discriminative value for the entropy values not in surrounded positive region, but whose sum is the same as entropies in the positive region.

### 6.2.3 Parzen-kNN

The Parzen-kNN density estimator was tested using a diagonal bandwidth matrix with elements equal to the square of the minimal hypersphere radius containing $k$ samples. The parameters $k$ and $c$ where chosen manually, in the same way as $c$ was chosen for Parzen denisty estimation in CSDE-space and an expected CCReq of 67.3% was achieved. The full results are presented in table 6.2.3. The density plots and discrimination functions are shown in figure A.2.3. We will now look closer at this figure. Notice, in the discrimination functions, that the negative valley is relatively shallow compared to the height of the positve peak. We can understand why this is the case by looking the the density plots, from which the discrimination functions are computed. In particular, the good prognosis density plots have much larger values than the poor prognosis plots. We have identified two causes that contributes to this effect. First, the poor prognosis samples are somewhat more dispersed in the CSDE-space and the Gaussian window is therefore often times larger than if the samples were more focused in CSDE-space. This claim is supported by e.g. the corresponding kNN density plots for CSDE-space, see figure A.2.2. Secondly, as there are fewer samples associated with poor prognosis than with good prognosis, the

| Parzen-kNN (c = 0.1, k = 2000) | |
|---|---|
| CCReq | 67.3 % [52.0 %, 79.9 %] |
| CCR | 69.3 % [57.5 %, 80.0 %] |
| Specificity | 71.7 % [55.2 %, 86.2 %] |
| Sensitivity | 63.0 % [36.4 %, 90.9 %] |

*Table 6.2.3: The classification results when using the CSDE-space density estimates from Parzen-kNN, and using the parameters c and k which achieved the best expected CCReq.*

minimal Gaussian windows containing $k$ samples will often differ for poor and good prognosis due to this simple reason. On to basis of these observations, one could argue that $k$ should be a function of the number of samples from the density it tries to estimate.

### 6.2.4 Normal distribution

In CSDE-space, we tested two assumptions for the covariance matrix of the parametric normal density estimator; unrestricted and diagonal. The unrestricted covariance matrix produced the best results, with an expected CCReq of 72.1%, while the diagonal covariance matrix produced a best expected CCReq of 71.6%. These detailed results are listed in table 6.2.4 and 6.2.5, respectively. Judging from the bootstrap plot in figure 6.2.2 when not restricting the covariance matrix, it is possible that the peak is due to bootstrap variation, but otherwise the expected CCReq's are stable just below 72%. We can therefore be confident that we have improved upon the CSDEM benchmark result of 69.1% in table 6.0.1. It is interesting to note that the parametric methods are outperforming non-parametric methods in CSDE-space, while the opposite is the case in CSDEsum-space.

The resubstitution density estimates and discrimination functions are shown in figure A.2.4. When comparing these discrimination functions with those from the non-parametric methods, we can see that these are simpler. In particular, the positive and the negative regions are smaller and has a shorter transition to the surrounding region of low discriminative value, which is also more uniform.

In section 6.1.4 we observed and discussed how the sensitivity increased drastically with $c$, meaning the classifier got better at classifying poor prognosis patients as $c$ increased. When we look at *both* the bootstrap plots when using an unrestricted covariance matrix (figure 6.2.2) and a diagonal covariance matrix (figure 6.2.3), we observe a similar trend. By animating an image sequence of density plots for increasing $c$, we can also see that the same observation made in the CSDEsum-space holds for CSDE-space. In particular; the positive and negative region of the discrimination functions move towards each other and shrink as $c$ increases. It is difficult to know precisely why this benefits sensitivity, while specificity suffers. Our hypothesis would be that some of the entropies from the poor prognosis patients contribute positively towards a good prognosis labelling only when $c$ is small, but as $c$ increases they no longer contribute significantly to any of the prognosis groups. These would be entropies located at the "edges" of the positive region for small $c$, but not at the "edge" that represents the margin between the positive and the negative region, as these

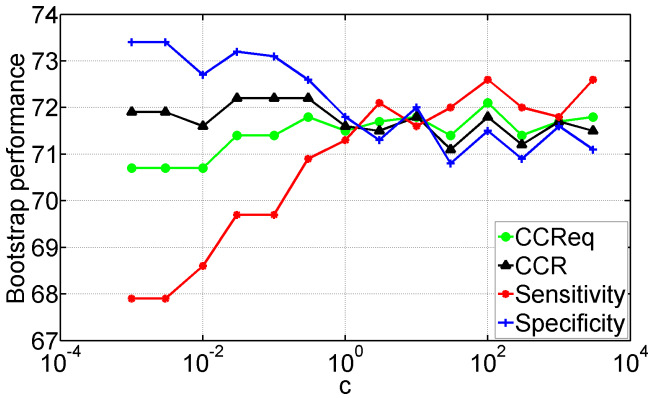


*Figure 6.2.2: Bootstrap performance when using the parametric normal density estimator in CSDE-space and not restricting Σ. It was run for c ∈ {0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000, 3000}. The best expected CCReq is achieved for c = 100.*

| Normal distribution (arbitrary Σ, c = 100) | |
|---|---|
| CCReq | 72.1 % [58.9 %, 85.1 %] |
| CCR | 71.8 % [60.0 %, 82.5 %] |
| Specificity | 71.5 % [55.2 %, 86.2 %] |
| Sensitivity | 72.6 % [45.5 %, 90.9 %] |

*Table 6.2.4: The classification results when using the CSDE-space density estimates from a fitted bivariate normal distribution with no covariance matrix restrictions, and using the parameter c which achieved the best expected CCReq.*

entropies would only contribute even more towards a good prognosis labelling because the regions move towards each other as $c$ increases. If we assume that there are many such entropies from the good prognosis patients as well, then we can understand why the increase in $c$ affects sensitivity positively, but specificity negatively.

### 6.2.5 Gaussian mixture model

In the case of the Gaussian mixture model we assume two components for the poor prognosis distribution and a single component for the good prognosis distribution. This is the same choice that we made for GMM in CSDEsum-space.

| Normal distribution (diagonal Σ, c = 100) | |
|---|---|
| CCReq | 71.6 % [59.4 %, 84.0 %] |
| CCR | 71.4 % [60.0 %, 82.5 %] |
| Specificity | 71.1 % [55.2 %, 86.2 %] |
| Sensitivity | 72.1 % [45.5 %, 90.9 %] |

*Table 6.2.5: The classification results when using the CSDE-space density estimates from a fitted bivariate normal distribution with a diagonal covariance matrix restriction, and using the parameter c which achieved the best expected CCReq.*

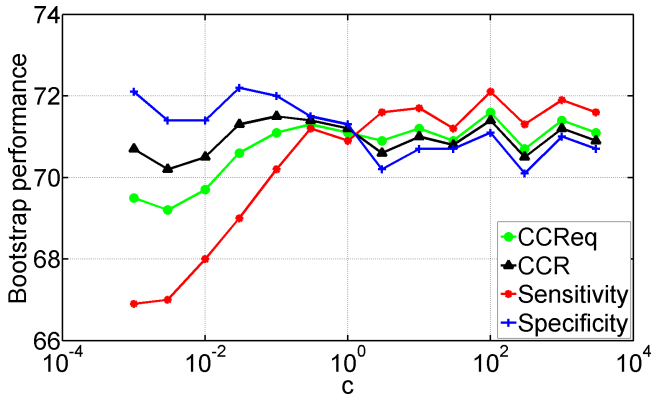


*Figure 6.2.3: Bootstrap performance when using the parametric normal density estimator in CSDE-space and diagonal covariance matrix. It was run for* $c \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000, 3000\}$. *The best expected CCReq is achieved for* $c = 100$.

To limit the number of possible discrimination functions, we will say that whenever we assume diagonal or unrestricted covariance matrices, this assumption applies to two both components in the poor prognosis group and also to the single component in the good prognosis group. For the two components of the poor prognosis we will either use individual covariance matrices or assume a shared covariance matrices. Whenever we specify that the covariance matrices are non-shared or shared, it applies only to the two components for poor prognosis, i.e. the component for good prognosis will always use a separate covariance matrix. This gives four possible discrimination functions in total, and we will now summarise the results.

In all cases, we observe that the sensitivity increases as $c$ increases, but the preference for very high values of $c$, as was the case when assuming normal distributions in CSDE-space in section 6.2.4, is not present here. This is evident when inspecting the bootstrap plots for all cases; non-shared and diagonal covariance matrices (figure 6.2.4), shared and diagonal covariance matrices (figure 6.2.5), shared and unrestricted covariance matrices (figure 6.2.7) and finally the non-shared and unrestricted covariance matrices (figure 6.2.6).

The case that achieved the best result was for the most degrees of freedom; the individual and unrestricted covariance matrices, and with $c = 0.3$. Here we achieved an CCReq of 70.5% and a CCR of 69.0%. The full results when using the best $c$ of each case are listed in tables 6.2.6 through 6.2.9. We observe that the introduction of a second component in the poor prognosis group results in relatively complex discrimination functions, see figures A.2.6 through A.2.9. We observe discrimination functions with multiple positive regions and multiple negative regions, and thus there is no longer a simple separating margin between the two prognosis groups, although the tendency of two large main regions, as was the case in the parametric normal case is still there. If we compare the density estimates of the poor prognosis groups in all four cases, we see that the two components are roughly located in the same location regardless of which case we are in. In particular, for the poor prognosis and bright segmentation class, there is always a small component located roughly to the north west

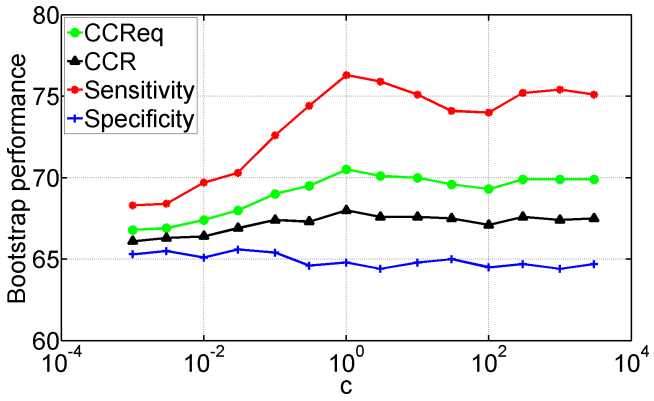


*Figure 6.2.4: Bootstrap performance when using the parametric GMM density estimator in CSDE-space and a non-shared, diagonal covariance matrix. It was run for* $c \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000, 3000\}$. *The best expected CCReq is achieved for* $c = 1$.

| GMM (non-shared, diagonal $\Sigma$, c = 1) | |
|---|---|
| CCReq | 70.5 % [56.0 %, 82.8 %] |
| CCR | 68.0 % [55.0 %, 80.0 %] |
| Specificity | 64.8 % [48.3 %, 79.3 %] |
| Sensitivity | 76.3 % [45.5 %, 100.0 %] |

*Table 6.2.6: The classification results when using the CSDE density estimates from a fitted GMM, and using the parameter c which achieved the best expected CCReq. The mixture model assumed two components for cells belonging to poor prognosis patients and a single component for cells from good prognosis patients. The two components for poor prognosis are assumed to have non-shared, diagonal covariance matrices.*

of the center of a larger component. While for the poor prognosis and dark semgentation class, there is a small component located roughly to the south west of the center of a larger component. This is interesting as it indicates that our assumption that there are two different clusters of cells within the poor prognosis group may be reasonable. However the discrimination functions resulting from these density plots does not improve classifier performance significantly over the CSDEM benchmark, in table 6.0.2. In the case where we assume the two components to have a shared, diagonal covariance matrix, the performance is at best equal to the CSDEM benchmark.

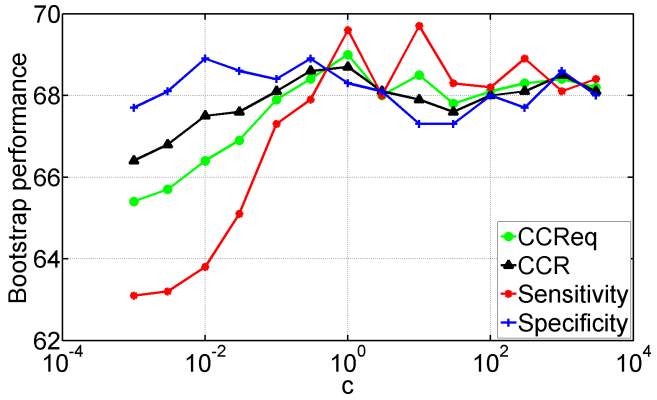


*Figure 6.2.5: Bootstrap performance when using the parametric GMM density estimator in CSDE-space and a shared, diagonal covariance matrix. It was run for* $c \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000, 3000\}$*. The best expected CCReq is achieved for* $c = 1$*.*

| GMM (shared, diagonal $\Sigma$, c = 1) | |
|---|---|
| CCReq | 69.0 % [56.1 %, 81.7 %] |
| CCR | 68.7 % [57.5 %, 80.0 %] |
| Specificity | 68.3 % [51.7 %, 86.2 %] |
| Sensitivity | 69.6 % [36.4 %, 90.9 %] |

*Table 6.2.7: The classification results when using the CSDE density estimates from a fitted GMM, and using the parameter c which achieved the best expected CCReq. The mixture model assumed two components for cells belonging to poor prognosis patients and a single component for cells from good prognosis patients. The two components for poor prognosis are assumed to have a shared, diagonal covariance matrix.*

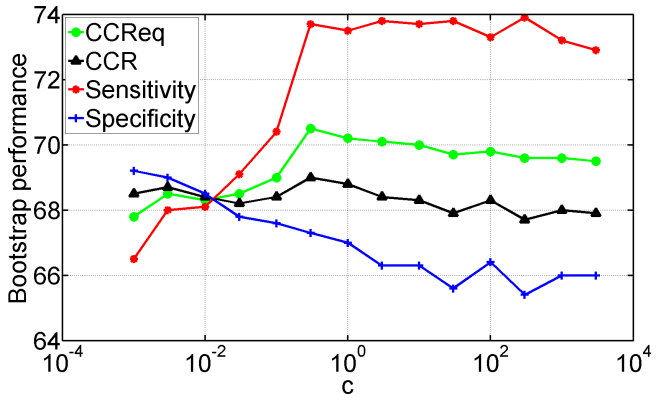


*Figure 6.2.6: Bootstrap performance when using the parametric GMM density estimator in CSDE-space and a non-shared, arbitrary covariance matrix. It was run for* $c \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000, 3000\}$*. The best expected CCReq is achieved for* $c = 0.3$*.*

| GMM (non-shared, arbitrary $\Sigma$, c = 0.3) | |
|---|---|
| CCReq | 70.5 % [56.6 %, 82.8 %] |
| CCR | 69.0 % [57.5 %, 80.0 %] |
| Specificity | 67.3 % [48.3 %, 82.8 %] |
| Sensitivity | 73.7 % [45.5 %, 100.0 %] |

*Table 6.2.8: The classification results when using the CSDE density estimates from a fitted GMM, and using the parameter c which achieved the best expected CCReq. The mixture model assumed two components for cells belonging to poor prognosis patients and a single component for cells from good prognosis patients. The two components for poor prognosis are assumed to have non-shared and arbitrary covariance matrices.*

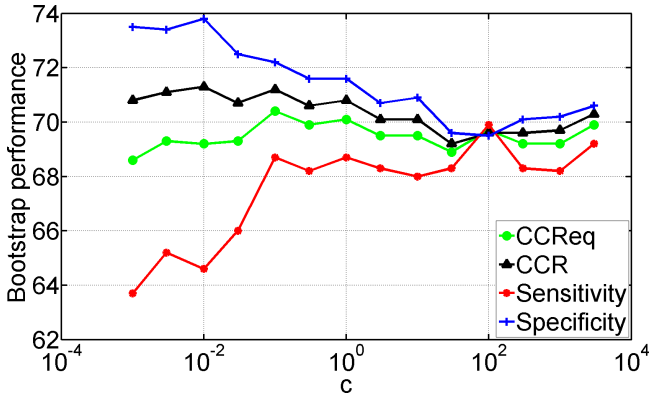


*Figure 6.2.7: Bootstrap performance when using the parametric GMM density estimator in CSDE-space and a shared, arbitrary covariance matrix. It was run for* $c \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000, 3000\}$. *The best expected CCReq is achieved for* $c = 0.1$.

| GMM (shared, arbitrary$\Sigma$, c = 0.1) | |
|---|---|
| CCReq | 70.4 % [57.2 %, 82.9 %] |
| CCR | 71.2 % [57.5 %, 82.5 %] |
| Specificity | 72.2 % [55.2 %, 86.2 %] |
| Sensitivity | 68.7 % [45.5 %, 90.9 %] |

*Table 6.2.9: The classification results when using the CSDE density estimates from a fitted GMM, and using the parameter c which achieved the best expected CCReq. The mixture model assumed two components for cells belonging to poor prognosis patients and a single component for cells from good prognosis patients. The two components for poor prognosis are assumed to have a shared and arbitrary covariance matrix.*

# 7. Conclusion

The main aim of this study was to improve and generalise a recent method for reliably estimating the prognosis of patients with early ovarian cancer. That approach was based on a novel texture analysis concept coined the *class specific dual entropy matrix* (CSDEM); a quantification of class specific dual entropy space (CSDE-space). In the present study, we defined CSDE-space as the Euclidean space comprising all possible combinations of spatial entropy and grey level entropy values for a specific segmentation class. Then we described how to extract adaptive features using a discrimination function that can exploit the entropy values in their continuous nature, and we discussed how this continuous approach relates to the quantification approach, which is the approach we wanted to improve and generalise.

We pointed out that the quantification approach is more difficult than our continuous approach to apply in an optimal way. This is because setting the quantification parameters may require domain knowledge. In particular, we argued that optimal sizes and locations of the bins depends on the nature of the discrimination function and that such considerations should preferably be unnecessary to extract the best features adaptively.

The best adaptive features were achieved using our continuous discrimination functions and density estimates from the non-parametric kNN density estimator in CSDEsum-space. Using these features we achieved an average of specificity and sensitivity of 74.7 % and a correct classification rate of 75.4%. These results are an improvement of nearly 3% over the best texture analysis results achieved prior to this study.

We also see a consistent improvement by using our continuous approach in CSDE-space rather than the quantified approach based on the CSDEM. In this space, our best result was achieved using the parametric normal density estimator with no restrictions on the covariance matrix. Using these features we achieved an average of specificity and sensitivity of 72.1% and a correct classification rate of 71.8%.

When we look at our results overall, one of the interesting trends we observe is that the non-parametric methods are outperforming parametric methods in CSDEsum-space, and conversely, parametric methods are outperforming non-parametric methods in CSDE-space. The achievable accuracy and reliability of the non-parametric density estimates is highly dependent on the number of samples available from the distribution they are estimating. If we then consider how much more densely populated the CSDEsum-space is compared to CSDE-space, while also keeping in mind that the projection of CSDE to CSDEsum maintains much of the available discriminative information, we can understand why we observe this trend. In particular, it is evident from the classification

results in CSDE-space that the performance loss due to discrepancies between the unknown true distribution and the assumed multivariate normal distribution is less than the loss caused by making no assumptions and therefore obtaining poor estimates using a non-parametric method.

To evaluate the performance of our adaptive texture features, we applied a proper evaluation method based on statistical bootstrapping. We can therefore expect our approach to generalise well, but this should be formalized by evaluating our approach on an independent test set.

Our continuous approach is consistently better in both CSDE-space and CSDEsum-space than its quantified counterpart. This strongly suggests that our continuous approach can achieve more reliable and accurate estimates than the quantification approach. Considering that our method is a generalisation of the CSDE and CSDEsum quantification approach, this is a good result that reinforces the promise of using the class specific grey level and spatial entropies for prognostics. This study therefore represents one step towards more reliable estimation of prognosis. Such information is important to make a qualified selection of the appropriate treatments for the patients.

# 8. Further work

- Evaluate the performance of the features obtained in CSDEsum-space when using an independent test set.
- The projection of CSDE-space onto the identity line might not be optimal, especially if we change the contextual measurement different from using the object size. The projection might also be suboptimal for datasets from other types of cancer. These relationships should be investigated.
- Investigate the possibility of estimating a pdf for each patient and then compare it to the pdfs of good prognosis and poor prognosis patients in the training set.
- The cell nucleus images are not homogeneous in terms of cancer, thus one could try to apply unsupervised learning to test whether there exists clusters of cells, with respect to a property space of interest. If distinctive clusters can be identified, then we can study if there is any connection between cluster assignment and prognostic value.
- Investigate the possibility of estimating the continuous discrimination functions directly, i.e. not through the estimation of densities as we have done is this study

# References


[1] F. Albregtsen and B. Nielsen. Texture classification based on cooccurrence of gray level run length matrices. *Australian Journal of Intelligent Information Processing Systems*, 6(1):38–46, 2000. 15

[2] F. Albregtsen, B. Nielsen, and H. E. Danielsen. Adaptive gray level run length features from class distance matrices. In *15th Proceedings of the International Conference on Pattern Recognition*, volume 3, pages 738–741, 2000. 15

[3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley, New York, 2001. 21, 22

[4] W. Härdle. *Nonparametric and semiparametric models*. Springer, Berlin, 2004. 23

[5] A. Kleppe. *Prognostics from adaptive spatial entropy in early ovarian cancer cell nuclei*. Master's thesis, 2011. 8, 9, 10, 11, 13, 15, 16, 17, 21, 25, 29, 30

[6] H. Maître, I. Bloch, and M. Sigelle. Spatial entropy: a tool for controlling contextual classification convergence. In *IEEE International Conference on Image Processing.*, volume 2, pages 212–216. 11

[7] B. Nielsen, F. Albregtsen, and H. E. Danielsen. Low dimensional adaptive texture feature vectors from class distance and class difference matrices. *IEEE Transactions on Medical Imaging*, 23(1):73–84, 2004. 15

[8] B. Nielsen, F. Albregtsen, and H. E. Danielsen. Statistical nuclear texture analysis in cancer research: A review of methods and applications. critical reviews$^{\mathrm{TM}}$ in oncogenesis. 14(2-3):89–164, 2008. 7, 15

[9] P. Nordby. *A combined structural/statistical texture analysis of monolayer ovarian cancer cell nuclei*. Master's thesis, 2010. 9, 10

[10] H. Schulerud, G. B. Kristensen, K. Liestøl, L. Vlatkovic, A. Reith, F. Albregtsen, and H. E. Danielsen. A review of caveats in statistical nuclear image analysis. *Analytical Cellular Pathology*, 16(2):63–82, 1998. 17, 25

[11] F. Tupin, M. Sigelle, and H. Maître. Definition of a spatial entropy and its use for texture discrimination. In *IEEE International Conference on Image Processing*, volume 1, pages 725–728. 11

[12] R. F. Walker, P. T. Jackway, and I. D. Longstaff. Recent developments in the use of the co-occurrence matrix for texture recognition. In *Proceedings of the 13th International Conference on Digital Signal Processing*, volume 1, pages 63–65. 15

[13] S. D. Yanowitz and A. M. Bruckstein. A new method for image segmentation. *Computer Vision, Graphics, and Image Processing*, 46(1):13, 1989. 9

# A. Density and discrimination plots

## A.1 CSDEsum-space

### A.1.1 Parzen

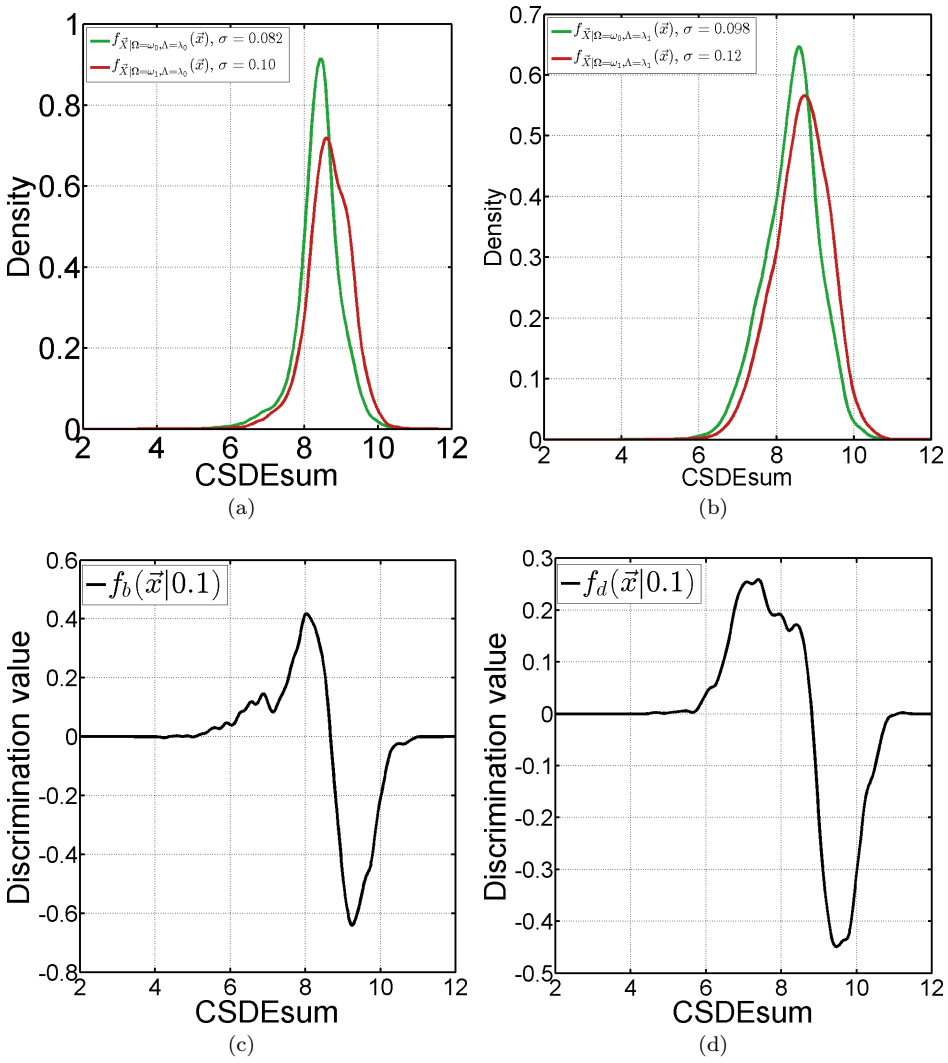


*Figure A.1.1: Parzen resubstitution plots. Density estimates uses the optimal bandwidth when assuming normal distribution. Green curves are the estimates from patients with good prognosis, while the red curves are for poor prognosis. The discrimination functions use $c = 0.1$. **a)** Bright segmentation class. **b)** Dark segmentation class. **c)** Bright discrimination function. **d)** Dark discrimination function.*

### A.1.2 kNN

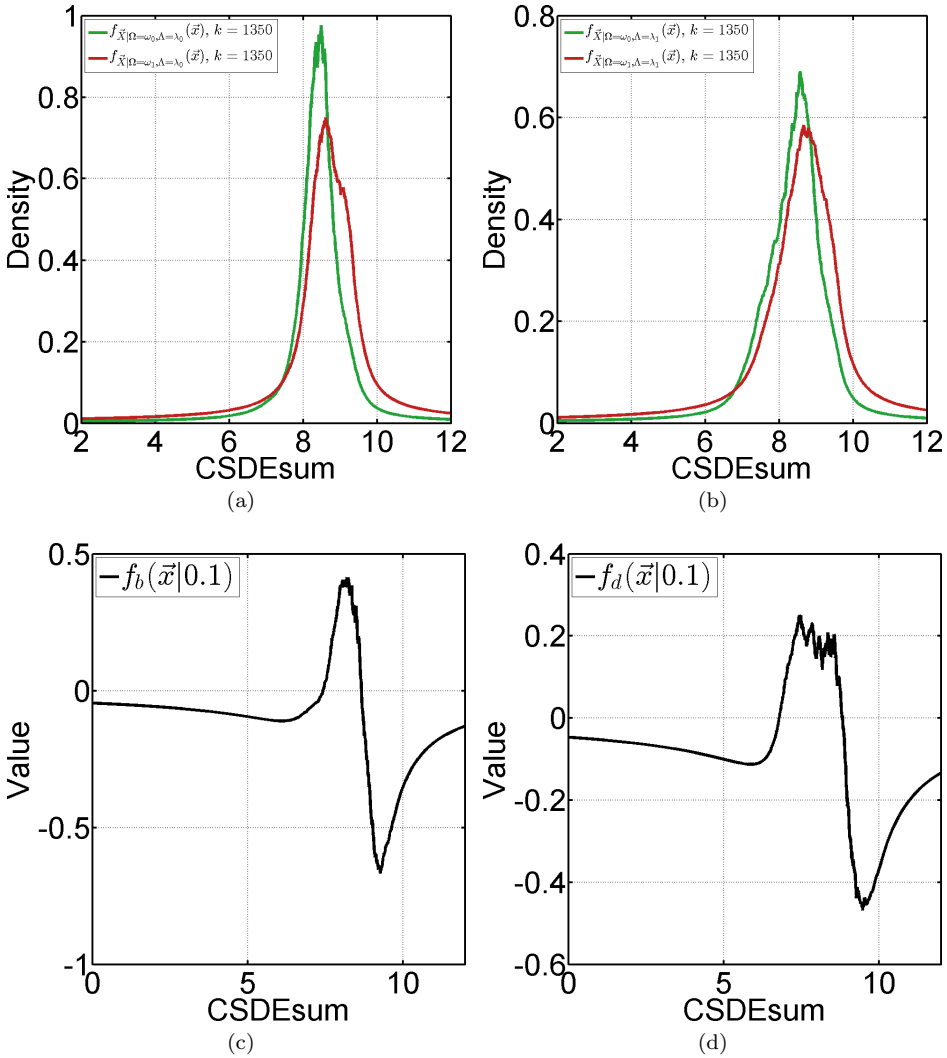


*Figure A.1.2: kNN resubstitution plots. Density estimates use $k = 1350$. The green curve represents good prognosis, while red curve poor prognosis. The discrimination functions use $c = 0.1$. **a)** Bright segmentation class. **b)** Dark segmentation class. **c)** Bright discrimination function. **d)** Dark discrimination function.*

### A.1.3 Parzen-kNN

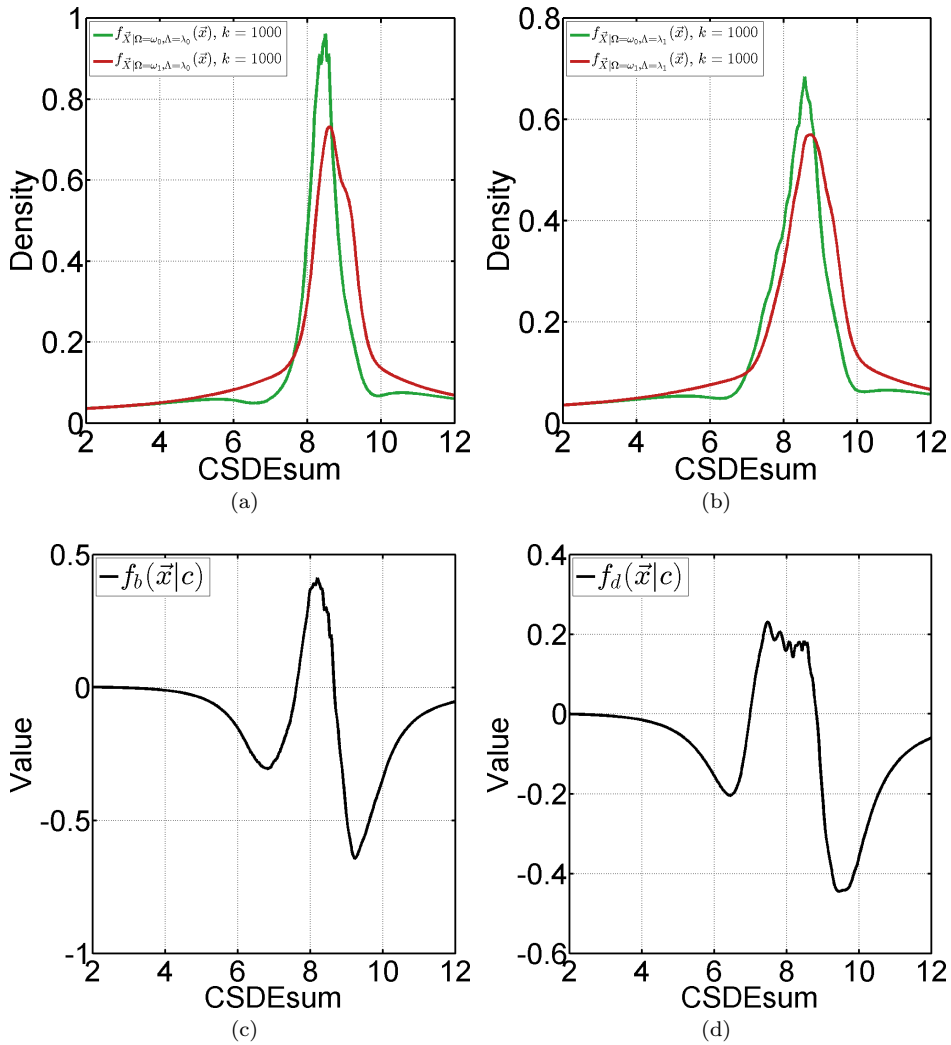


*Figure A.1.3: Parzen-kNN resubstitution plots. Density estimates use $k = 1000$. The green curve represents good prognosis, while red curve poor prognosis. The discrimination functions use $c = 0.1$. **a)** Bright segmentation class. **b)** Dark segmentation class. **c)** Bright discrimination function. **d)** Dark discrimination function.*

### A.1.4 Normal distribution

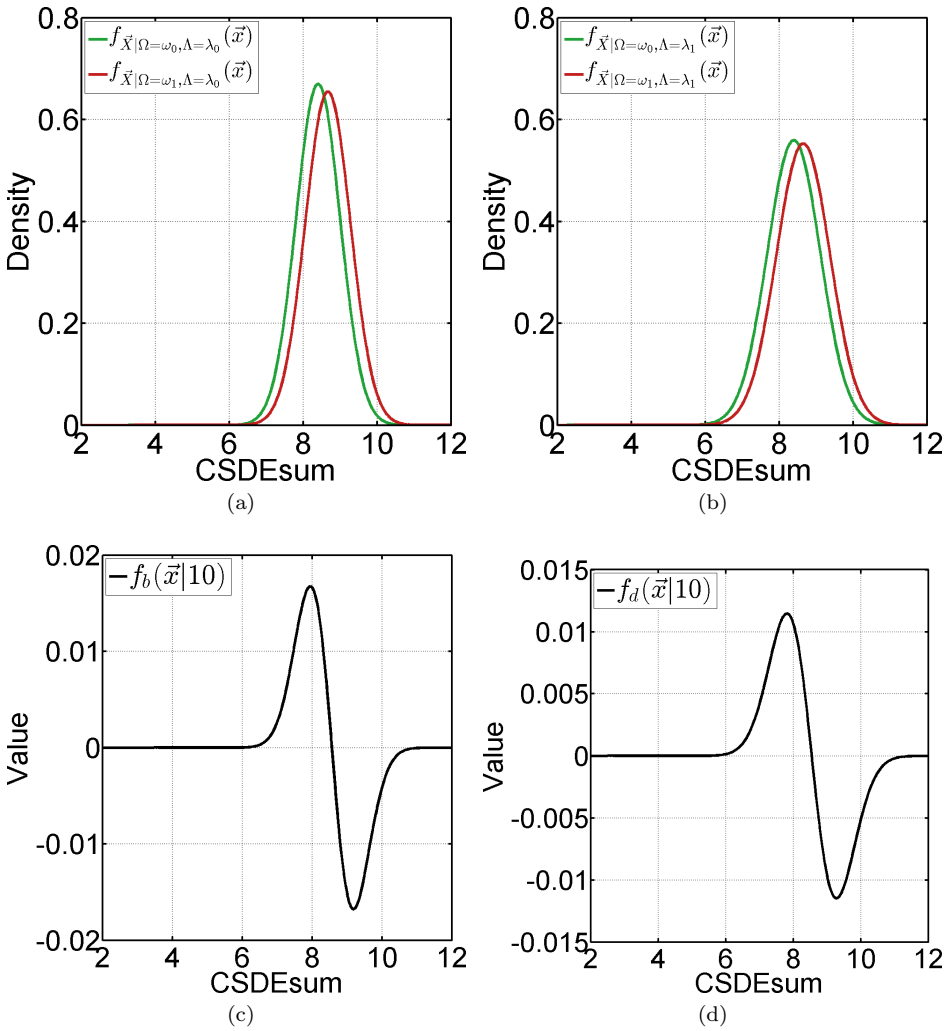


*Figure A.1.4: Parametric normal resubstitution plots. The green curve represents good prognosis, while red curve poor prognosis. The discrimination functions use* $c = 10$. ***a)*** *Bright segmentation class. Good prognosis parameters:* $\mu = 8.40$, $\sigma^2 = 0.355$. *Poor prognosis parameters:* $\mu = 8.66$, $\sigma^2 = 0.371$. ***b)*** *Dark segmentation class. Good prognosis parameters:* $\mu = 8.39$, $\sigma^2 = 0.509$. *Poor prognosis parameters:* $\mu = 8.64$, $\sigma^2 = 0.521$. ***c)*** *Bright discrimination function.* ***d)*** *Dark discrimination function.*

### A.1.5 Gaussian mixture model

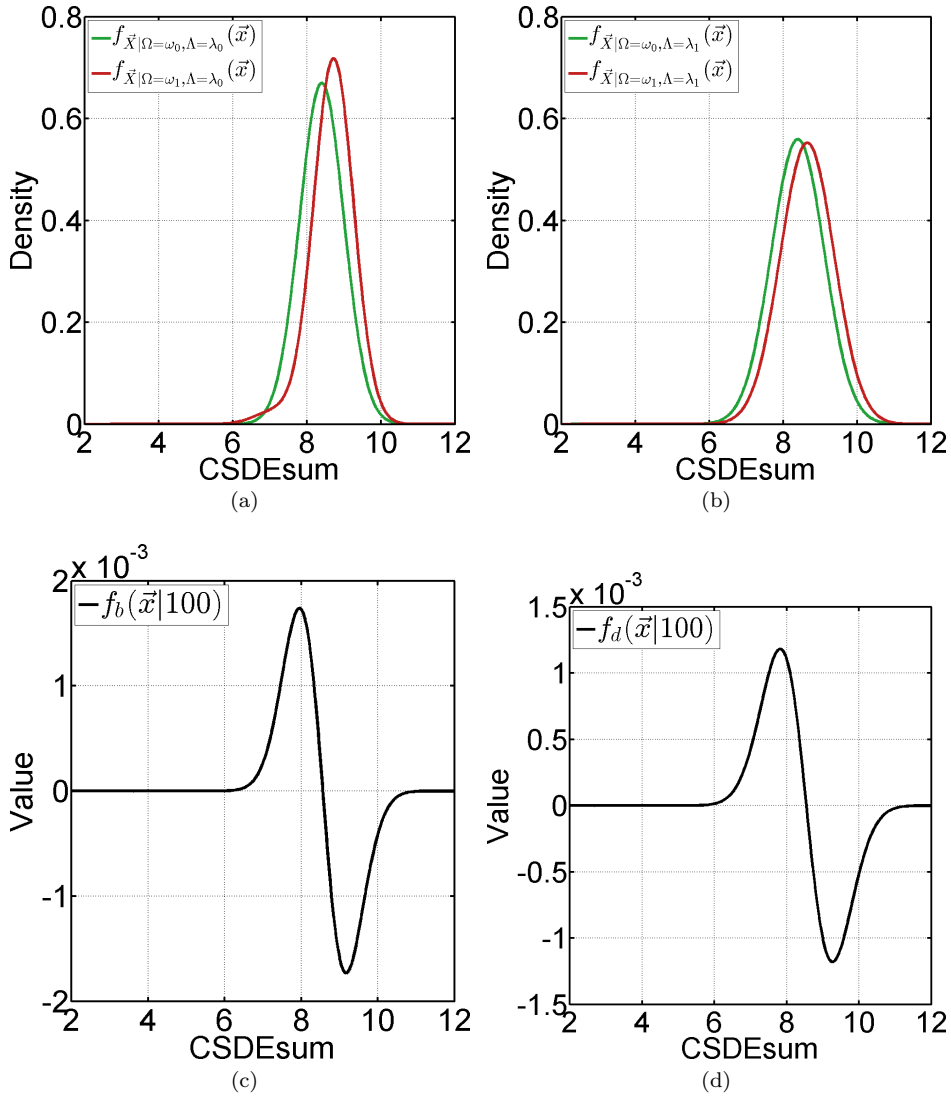


*Figure A.1.5: Parametric GMM resubstitution plots. Density estimates assume shared variance. The green curve represents good prognosis, while red curve poor prognosis. The discrimination functions use* $c = 100$. ***a)*** *Bright segmentation class. Good prognosis parameters:* $\mu = 8.40$, $\sigma^2 = 0.355$. *Poor prognosis parameters:* $\mu_1 = 8.71$, $\mu_2 = 7.12$ $\sigma^2 = 0.289$. ***b)*** *Dark segmentation class. Good prognosis parameters:* $\mu = 8.39$, $\sigma^2 = 0.509$. *Poor prognosis parameters:* $\mu_1 = 8.49$, $\mu_2 = 8.80$, $\sigma^2 = 0.497$. ***c)*** *Bright discrimination function.* ***d)*** *Dark discrimination function.*

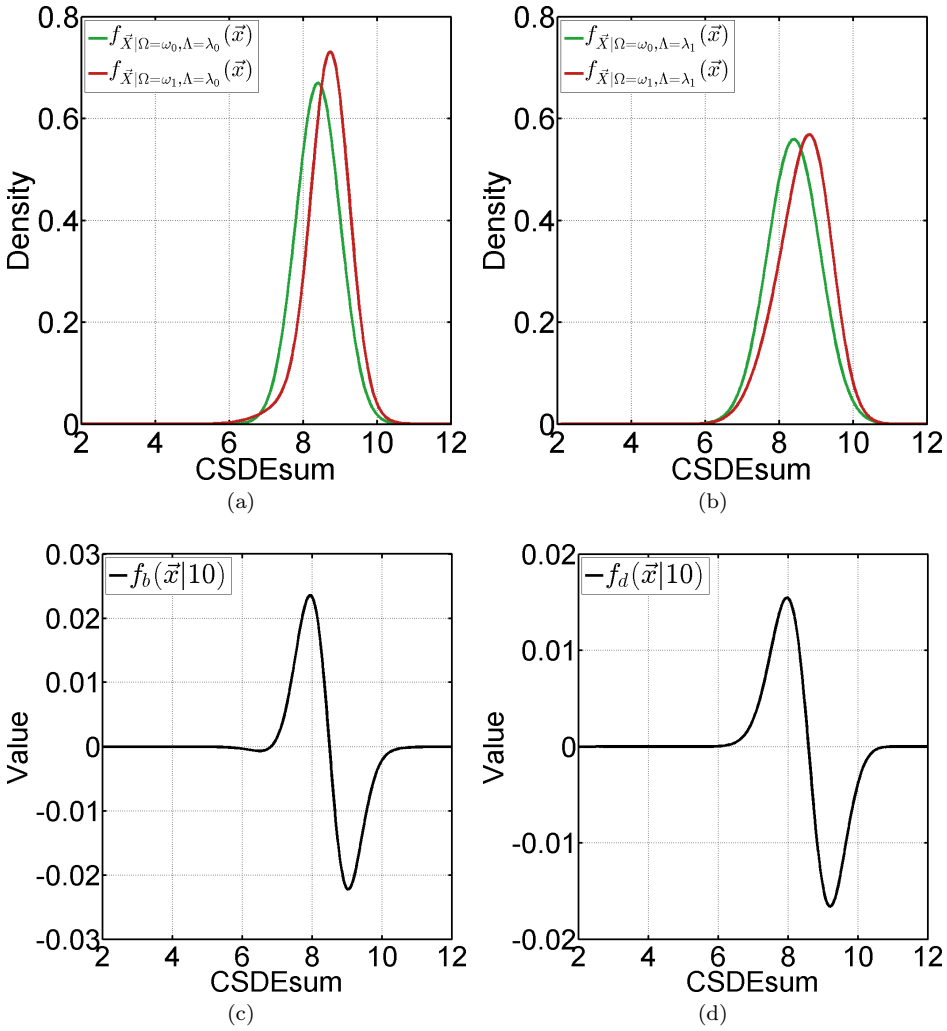


*Figure A.1.6: Parametric GMM resubstitution plots. Density estimates use individual variance estimates. The green curve rep resents good prognosis, while red curve poor prognosis. The discrimination functions use* $c = 10$. ***a)*** *Bright segmentation class. Good prognosis parameters:* $\mu = 8.40$, $\sigma^2 = 0.355$. *Poor prognosis parameters:* $\mu_1 = 8.21$, $\mu_2 = 8.73$ $\sigma_1^2 = 0.842$, $\sigma_2^2 = 0.255$ ***b)*** *Dark segmentation class. Good prognosis parameters:* $\mu = 8.39$, $\sigma^2 = 0.509$. *Poor prognosis parameters:* $\mu_1 = 8.98$, $\mu_2 = 8.36$ $\sigma_1^2 = 0.280$, $\sigma_2^2 = 0.543$. ***c)*** *Bright discrimination function.* ***d)*** *Dark discrimination function.*

## A.2 CSDE-space

### A.2.1 Parzen

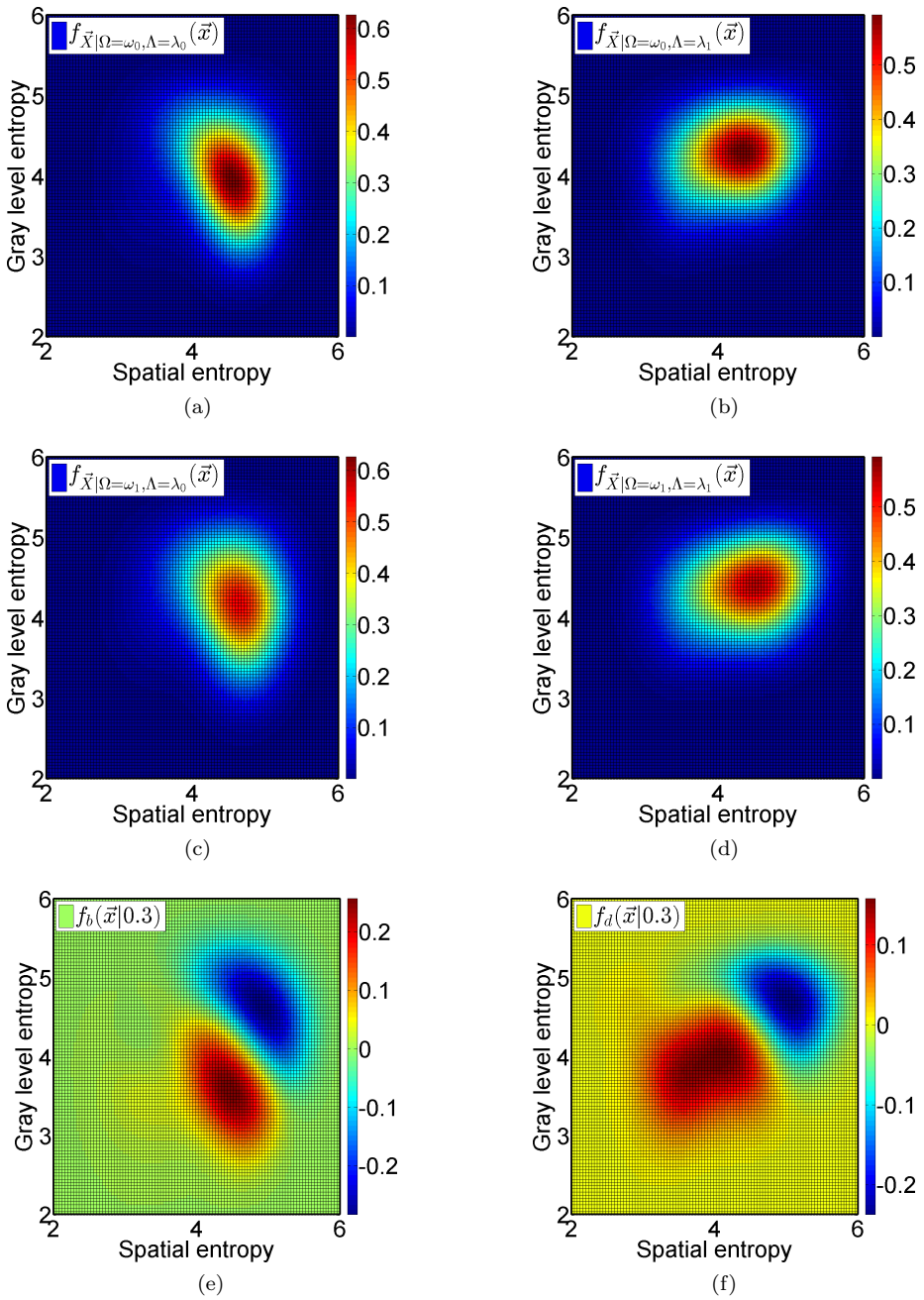


*Figure A.2.1: Parzen resubstitution plots. Density estimates use the optimal bandwidth when assuming normal distribution with diagonal bandwidth matrix, while the discrimination functions use $c = 0.3$.* ***a)*** *Bright segmentation class, good prognosis.* ***b)*** *Dark segmentation class, good prognosis.* ***c)*** *Bright segmentation class, poor prognosis.* ***d)*** *Dark segmentation class, poor prognosis.* ***e)*** *Bright discrimination function.* ***f)*** *Dark discrimination function.*

### A.2.2 kNN

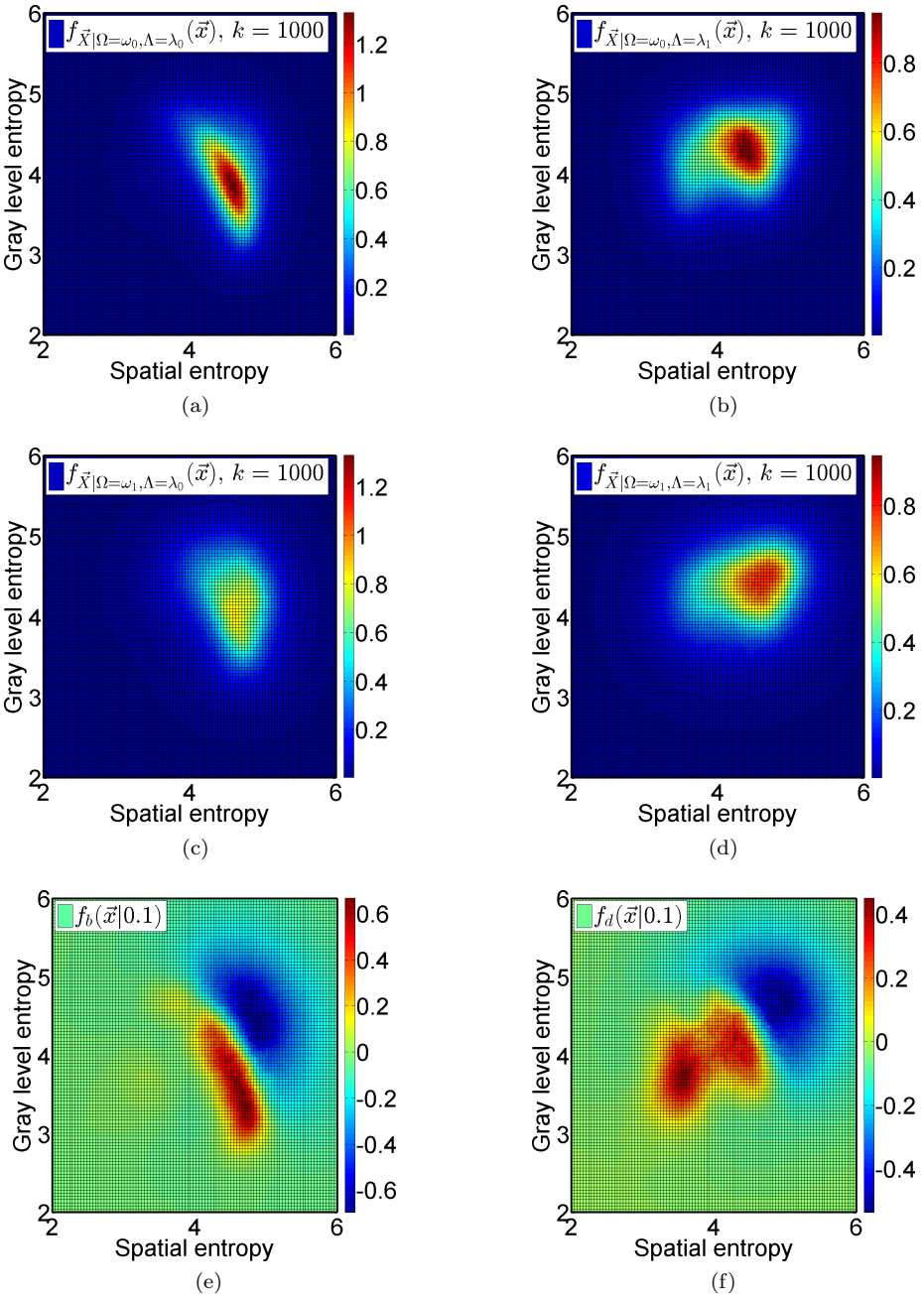


*Figure A.2.2: kNN resubstitution plots. Density estimates use $k = 1000$, while the discrimination functions use $c = 0.1$. **a)** Bright segmentation class, good prognosis. **b)** Dark segmentation class, good prognosis. **c)** Bright segmentation class, poor prognosis. **d)** Dark segmentation class, poor prognosis. **e)** Bright discrimination function. **f)** Dark discrimination function.*

### A.2.3 Parzen-kNN

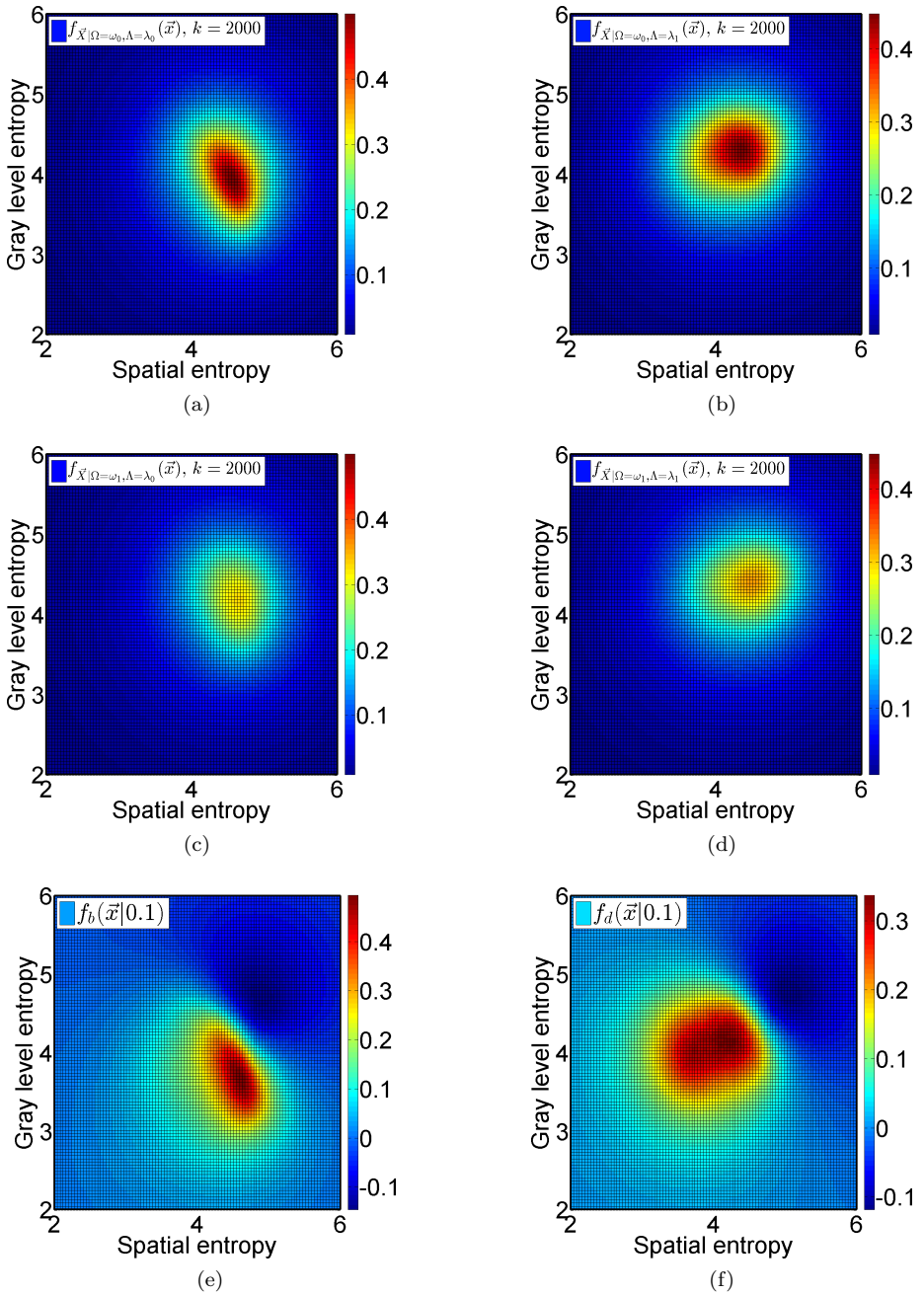


*Figure A.2.3: Parzen-kNN resubstitution plots. Density estimates assumes diagonal bandwidth matrices and $k = 2000$. The discrimination functions use $c = 0.1$. **a)** Bright segmentation class, good prognosis. **b)** Dark segmentation class, good prognosis. **c)** Bright segmentation class, poor prognosis. **d)** Dark segmentation class, poor prognosis. **e)** Bright discrimination function. **f)** Dark discrimination function.*

### A.2.4 Normal distribution

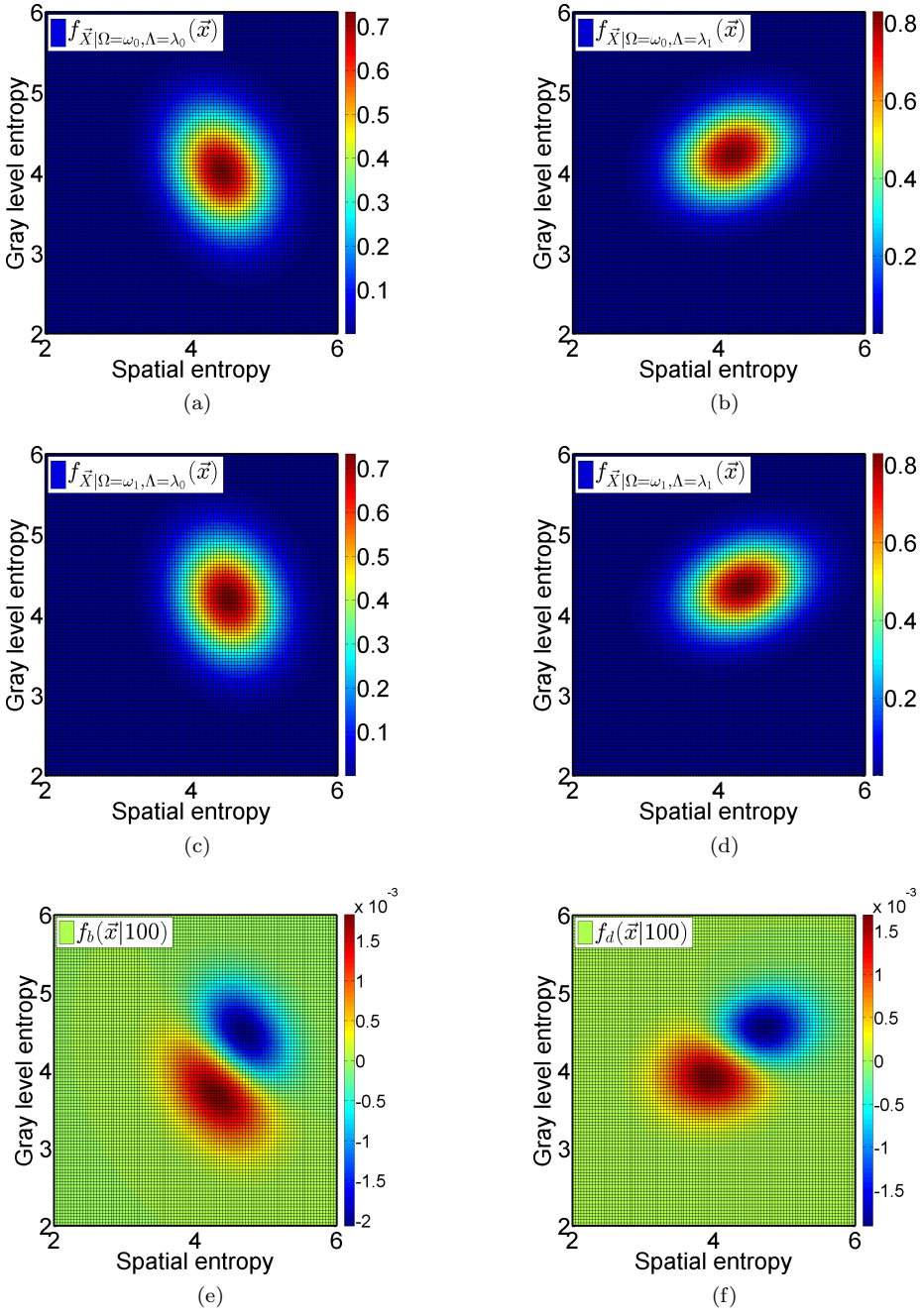


*Figure A.2.4: Parametric normal resubstitution plots. Density estimates use arbitrary covariance matrices. The discrimination functions use $c = 100$. **a)** Bright segmentation class, good prognosis. **b)** Dark segmentation class, good prognosis. **c)** Bright segmentation class, poor prognosis. **d)** Dark segmentation class, poor prognosis. **e)** Bright discrimination function. **f)** Dark discrimination function.*

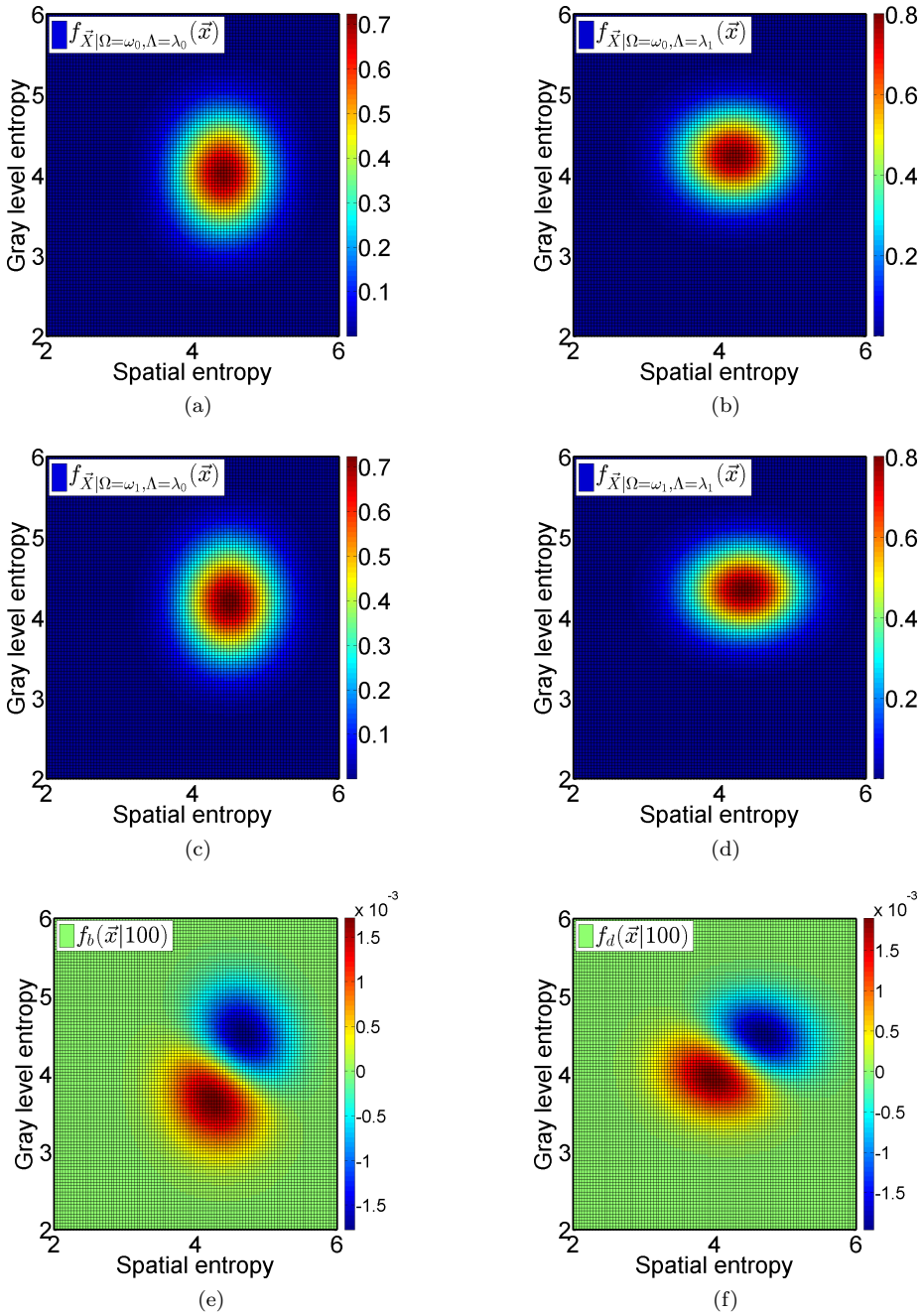


*Figure A.2.5: Parametric normal resubstitution plots. Density estimates assume diagonal covariance matrices. The discrimination functions use $c = 100$. **a)** Bright segmentation class, good prognosis. **b)** Dark segmentation class, good prognosis. **c)** Bright segmentation class, poor prognosis. **d)** Dark segmentation class, poor prognosis. **e)** Bright discrimination function. **f)** Dark discrimination function.*

### A.2.5 Gaussian mixture model

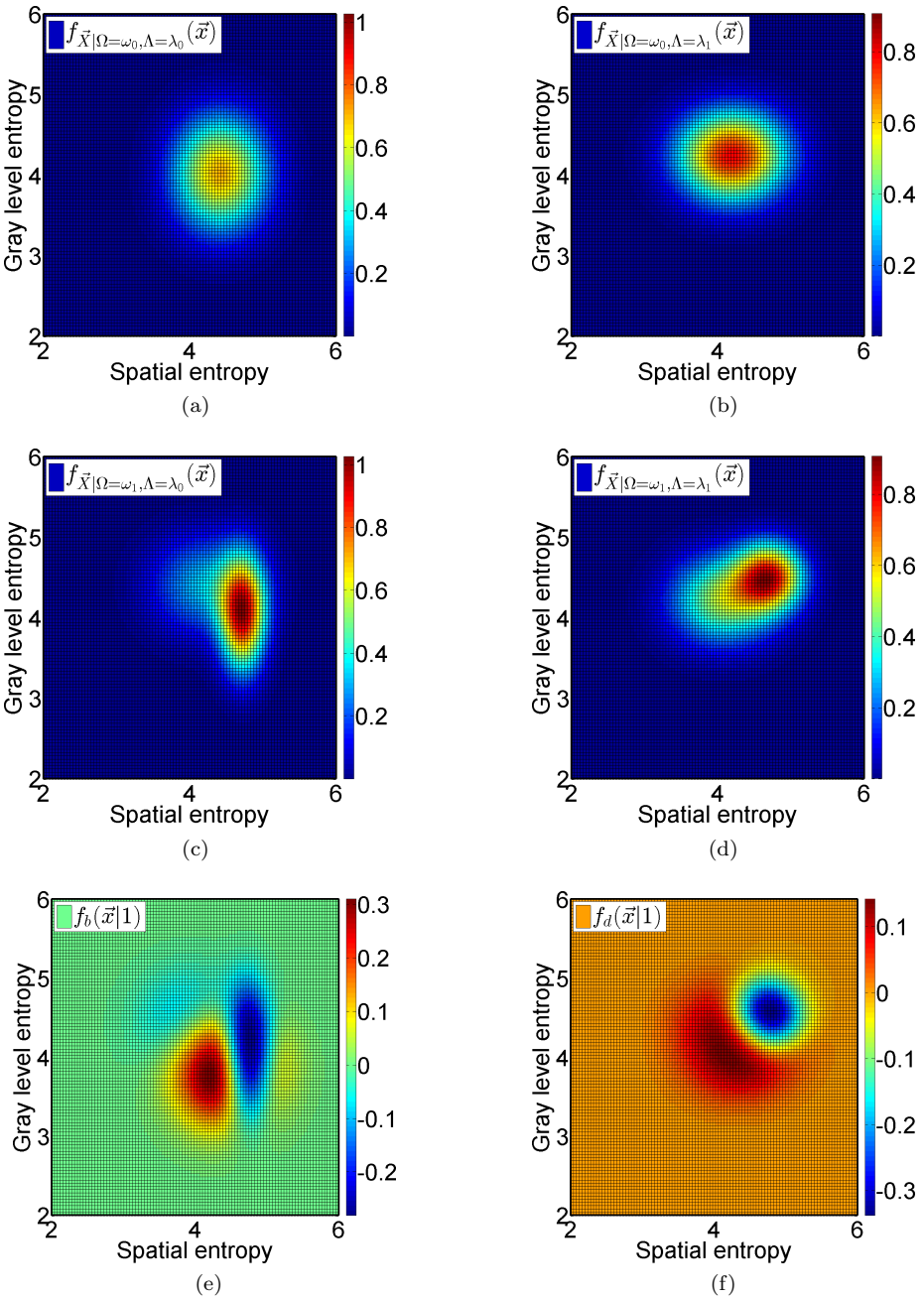


*Figure A.2.6: Parametric GMM resubstitution plots. Density estimates use individual covariance matrices which are assumed to be diagonal. The discrimination functions use $c = 1$. **a)** Bright segmentation class, good prognosis. **b)** Dark segmentation class, good prognosis. **c)** Bright segmentation class, poor prognosis. **d)** Dark segmentation class, poor prognosis. **e)** Bright discrimination function. **f)** Dark discrimination function.*

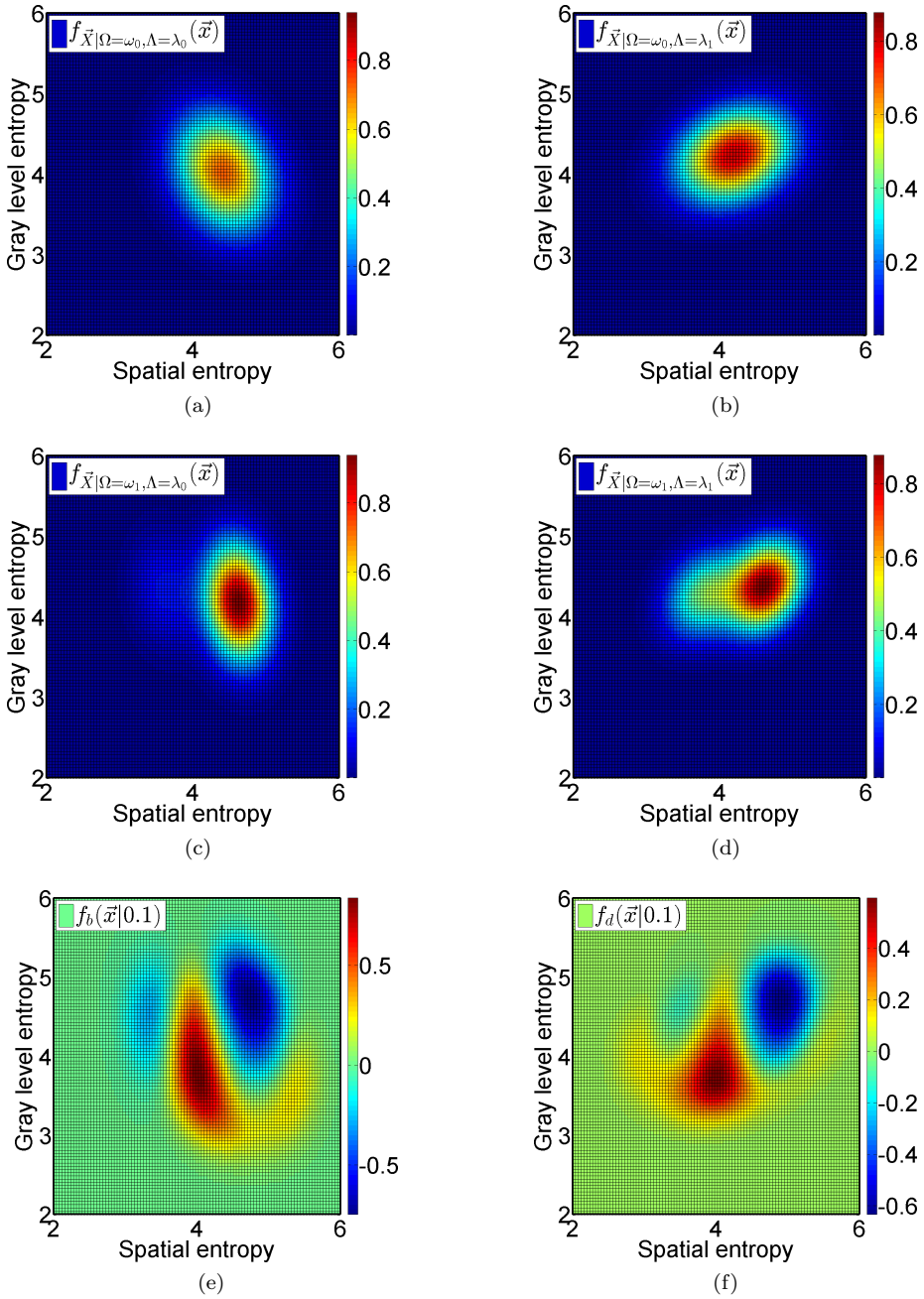


*Figure A.2.7: Parametric GMM resubstitution plots. Density estimates assume a shared covariance matrix, but this matrix may be arbitrary. The discrimination functions use $c = 0.1$. **a)** Bright segmentation class, good prognosis. **b)** Dark segmentation class, good prognosis. **c)** Bright segmentation class, poor prognosis. **d)** Dark segmentation class, poor prognosis. **e)** Bright discrimination function. **f)** Dark discrimination function.*

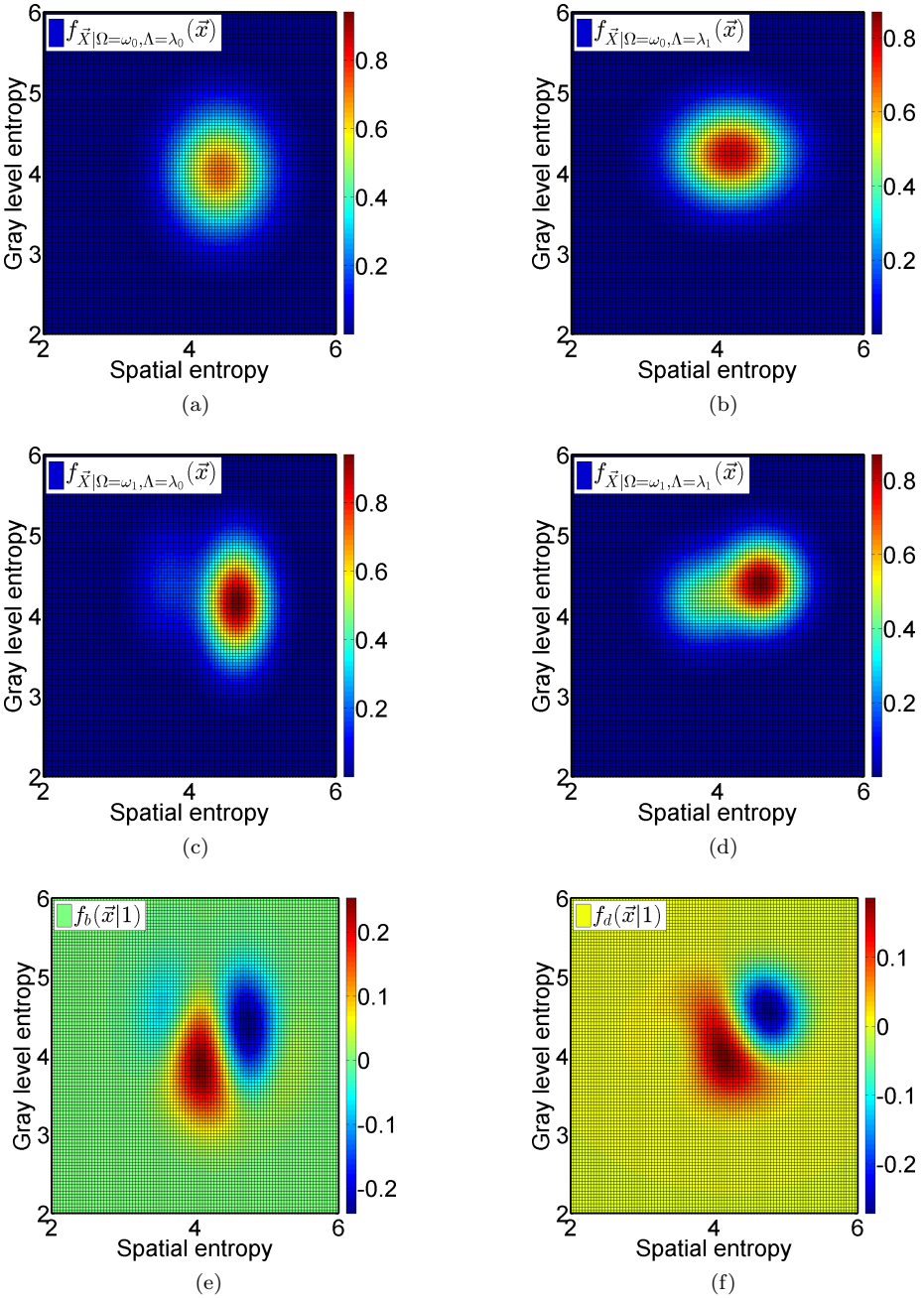


*Figure A.2.8: Parametric GMM resubstitution plots. Density estimates assume a shared and diagonal covariance matrix. The discrimination functions use $c = 1$.* ***a)*** *Bright segmentation class, good prognosis.* ***b)*** *Dark segmentation class, good prognosis.* ***c)*** *Bright segmentation class, poor prognosis.* ***d)*** *Dark segmentation class, poor prognosis.* ***e)*** *Bright discrimination function.* ***f)*** *Dark discrimination function.*

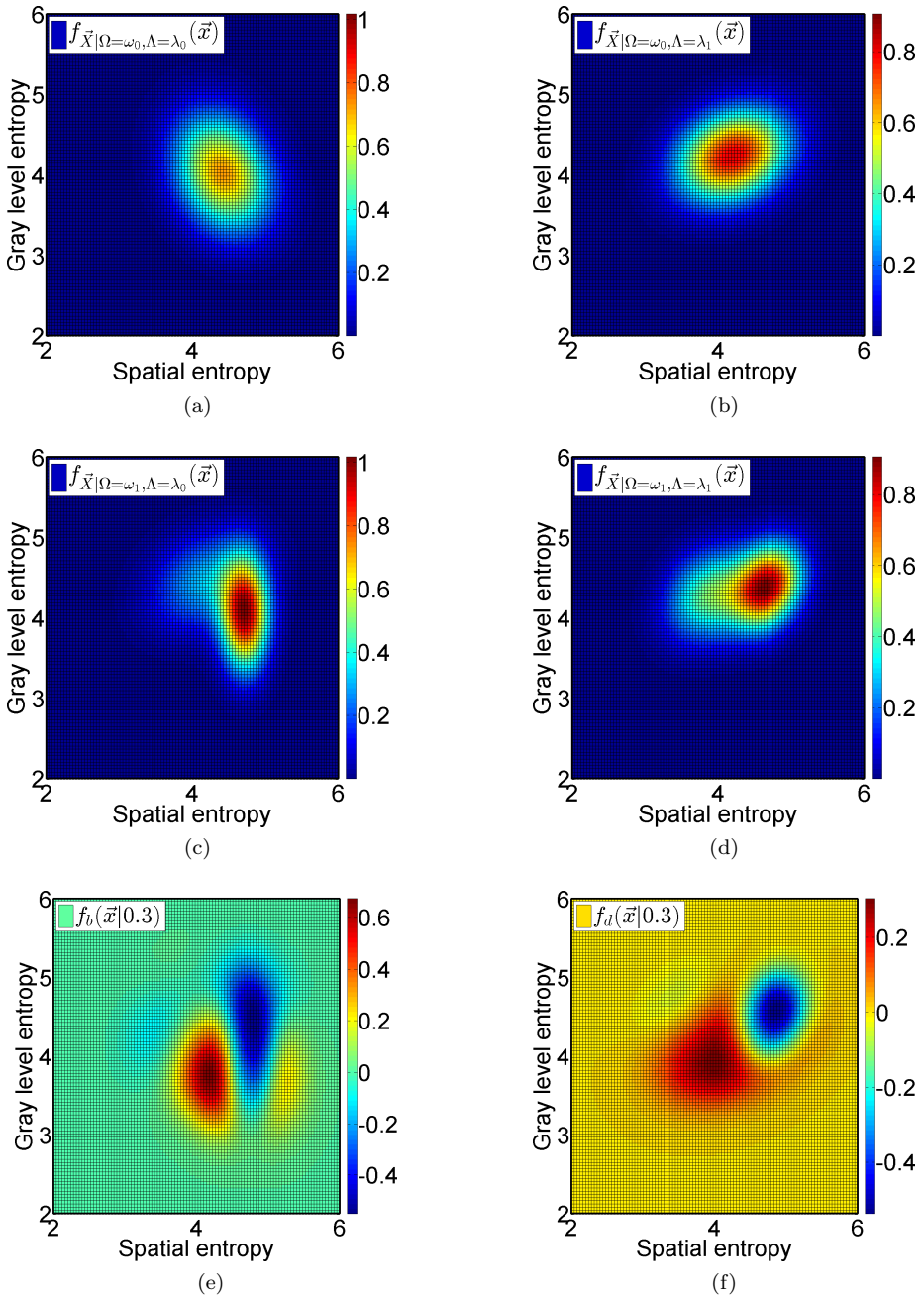


*Figure A.2.9: Parametric GMM resubstitution plots. Density estimates use individual and arbitrary covariance matrix. The discrimination functions use $c = 0.3$. **a)** Bright segmentation class, good prognosis. **b)** Dark segmentation class, good prognosis. **c)** Bright segmentation class, poor prognosis. **d)** Dark segmentation class, poor prognosis. **e)** Bright discrimination function. **f)** Dark discrimination function.*