Agnes LEITNER, BSc

Automatic Differentiation between Pseudoprogression and Tumor Recurrence by Applying an SVM to Multiparametric MRI-Data

Master Thesis

Institue of Medical Engineering
Graz University of Technology
Kronesgasse 5, A - 8010 Graz
Head: Univ.-Prof.Dipl.-Ing.Dr.techn. Rudolf Stollberger

Supervisor:
Univ.-Prof.Dipl.-Ing.Dr.techn. Rudolf Stollberger

Evaluator:
Univ.-Prof.Dipl.-Ing.Dr.techn. Rudolf Stollberger

Graz, July 2013

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....

date Agnes Leitner

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ....

Agnes Leitner

Abstract

Title: Automatic Differentiation between Pseudoprogression and Tumor Recurrence by Applying an SVM to Multiparametric MRI-Data

Objective: The aim of this retrospective study was to evaluate whether a differentiation of pseudoprogression and tumor recurrence in patients with resected glioblastoma multiforme was possible by applying a machine learning approach to data obtained by several different magnetic resonance imaging (MRI) sequences.

Methods: Data from 8 patients with pseudoprogression and 8 patients with tumor recurrence was used in this retrospective study. For each patient 8 images, obtained by multiple MRI sequences, were registered, such that each voxel was characterized by an 8-dimensional feature vector. A region of interest (ROI) was drawn over the contrast-enhanced lesion in the T1-weighted image. A one-class support vector machine (OC-SVM) was trained on the feature vectors of the voxels within the ROI of patients with pseudoprogression. The classifier was tested using cross-validation. The percentage of voxels within the ROI that were classified by the SVM to represent pseudoprogression was used to make a decision whether a patient suffered from recurrent tumor or showed pseudoprogression.

Results: The single voxels were classified with an area under the ROC-curve of 0.66. The percentages of voxels that were thought to represent pseudoprogression were significantly larger with a p-value of 0.0104 in patients with pseudoprogression compared to patients with tumor progression. The sensitivity and specificity with which the single patients were classified were 0.75 and 0.875 respectively.

Conclusion: The results showed that a differentiation based on MRI data using a machine learning approach is possible. However the excellent results of previous research by Hu et al. [1] could not be reproduced. Especially adjustments to the MRI protocols are expected to improve the method.

Keywords: Glioblastoma multiforme, Pseudoprogression, Support Vector Machine, Magnetic Resonance imaging

Kurzfassung

Titel: Automatische Unterscheidung von Pseudoprogression und Tumor Progression durch Anwendung einer SVM auf MRT-Daten

Ziel: Mit dieser Arbeit sollte retrospektiv überprüft werden, ob es möglich sei, aufgrund von Daten, die von verschiedenen Magnetresonanztomographiesequenzen (MRT-Sequenzen) stammten, Pseudoprogression von Tumor-Progression bei Glioblastom-Patienten mit Methoden des maschinellen Lernens zu unterscheiden.

Methoden: Die Bilder, die von unterschiedlichen MR-Sequenzen stammten, wurden registriert sodass jedes Voxel durch einen achtdimensionalen Feature-Vektor beschrieben wurde. Eine "Region of Interest" (ROI) wurde über die kontrastverstärkte Läsion im T1-gewichteten Bild eingezeichnet. Eine "one-class Support Vector Machine" (OC-SVM) wurde auf die Feature-Vektoren der Voxels innerhalb dieser ROIs durch Kreuzvalidierung trainiert und getestet. Der prozentuale Anteil von Voxels innerhalb der ROI, die durch die SVM als Pseudoprogression klassifiziert wurden, wurde für die Entscheidung, ob ein Patient ein Tumorrezidiv vorweist oder ob es sich um Pseudoprogression handelt, herangezogen.

Ergebnisse: Die einzelnen Voxels wurden mit einer Fläche unterhalb der ROC-Kurve ("area under curve", AUC) von 0.66 klassifiziert. Die prozentualen Anteile an Voxeln, die als Pseudoprogression klassifiziert wurden, waren in Patienten mit Pseudoprogression signifikant höher (p-Wert = 0.0104), als in Patienten mit Tumor-Progression. Die einzelnen Patienten wurden mit einer Sensitivität von 0.75 und einer Spezifität von 0.875 klassifiziert.

Schlussfolgerung: Die Resultate zeigen, dass eine Unterscheidung mithilfe von MR-Bilddaten und Anwendung von Methoden des maschinellen Lernens möglich ist. Die exzellenten Resultate früherer Untersuchungen durch Hu et al. [1] konnten jedoch nicht reproduziert werden. Es ist zu erwarten, dass besonders Anpassungen in den MR-Protokollen zu einer Verbesserung der Ergebnisse führt.

Schlüsselwörter: Glioblastoma multiforme, Pseudoprogression, Support Vector Machine, Magnetresonanztomographie

Contents

Abbreviations

ADC	apparent diffusion coefficient
AUC	area under the (ROC-)curve
CBF	cerebral blood flow
CBV	cerebral blood volume
DSC-MRI	dynamic susceptibility contrast-enhanced MRI
DWI	diffusion-weighted imaging
FLAIR	fluid attenuated inversion recovery
FOV	field of view
FN	number of false negatives
FP	number of false positives
GBM	glioblastoma mulitforme
KNN	k-nearest-neighbor
LDA	linear discriminant analysis
MRI	magnetic resonance imaging
MTT	mean transit time
nawm	normal appearing white matter
OC-SVM	one-class SVM
PDF	probability density function
RBF	radial-basis function
ROC	Receiver Operating Characteristic
ROI	region of interest
SN	sensitivity
SP	specificity
SV	support vector
SVM	support vector machine
T1WI	T1-weighted imaging
T2WI	T2-weighted imaging
TN	number of true negatives
TP	number of true positives
TTP	time to peak

Notation, Symbols and Definitions

$\mathcal{X}$	input space
$\mathcal{H}$	feature space
D	dimension of vector space
Φ	feature map $\mathcal{X} \rightarrow \mathcal{H}$
x	data point in input space
$\mathbf{w}$	weight vector in feature space
$\mathbf{x}$	a vector with entries x_i with $i = 1 \dots D$, usually a mapped data point in $\mathcal{H}$, $\mathbf{x} = \mathbf{\Phi}(x)$
$\mathbf{x}^T$	a transposed vector
$\mathbf{x}_1^T \cdot \mathbf{x}_2$	dot product between $\mathbf{x}_1$ and $\mathbf{x}_2$
$\mathbf{x}_n$	usually a vector of the training set with $n = 1 \dots N$
N	usually number of training samples
y_n	class label corresponding to the data point x_n
ν	regularization parameter in ν-SVMs
b	constant offset (or threshold)
ρ	margin parameter
α_n	Lagrangian multiplier corresponding to the data point x_n
ξ_n	slack variable
k	kernel
σ	standard deviation or parameter controlling width of gaussian kernel
γ	parameter controlling width of gaussian kernel, inverse proportional to σ
T	variable in one-class SVM, that allows to make the region bigger that is estimated to contain the class
c	parameter, based on which T is defined
h	activation function
$\mathrm{sgn}(x)$	$= \begin{cases} +1 & \text{for } x \geq 0 \\ -1 & \text{for } x < 0 \end{cases}$

1 Introduction

Glioblastoma multiforme is the most common malignant brain tumor. It is commonly treated with surgical resection and subsequent chemoradiotherapy. The response to the therapy is assessed using magnetic resonance imaging (MRI). However, treatment related changes mimic tumor progression. The so-called pseudoprogression cannot be distinguished from tumor progression in conventional MRI. The aim of this thesis was to find a way to distinguish those two entities using MRI. Perfusion imaging was most promising for the task to reveal differences between pseudoprogression and tumor progression [2].
Most of the recent studies, that evaluated the use of perfusion imaging for the differentiation of pseudoprogression from tumor progression, evaluated mean, minimal and/or maximal relative cerebral blood volume (CBV) within a region of interest (ROI) [2]. Currently most researchers try to find a threshold for one or a few of those parameters in order to make the differentiation. Although those studies yielded significant differences between patients with recurrent tumor and patients with pseudoprogression, there was still some degree of overlap.
In contrast to simple threshold-based approaches, in this study a machine learning approach was chosen for the decision making. Machine learning is found in a wide range of medical applications. For example machine learning approaches have been used to perform segmentation in medical images and to perform registration of medical images from different modalities or time series [3]. Machine learning is also found in computer-aided detection and diagnosis systems and in content-based image retrieval systems [3]. Numerous machine learning techniques that are used in clinical practice are classification techniques. For example in radiology computer-aided detection and diagnosis systems are frequently used for the classification of lesions [3].
For example support vector machines (SVMs) have been shown to be feasible to detect microcalcification from digital mammograms [4]. SVMs were also applied to detect structural patterns in cerebral MRI for an early diagnosis of Alzheimer's disease [5, 6]. SVMs as well as neural networks were used to identify the type or grade of brain tumors on MRI [7, 8, 9] and to detect multiple sclerosis [10]. Several different machine learning approaches were applied in colon cancer research. Detection systems for computed tomographic colonography have shown to be highly accurate in detecting colonic polyps [3]. Classifiers, such as SVMs, linear discriminant analysis (LDA) and neural networks, can

also be trained on data obtained by functional MRI (fMRI) in order to differentiate brain activity patterns [3].

In this study a machine learning approach was used, because that way CBV data can be combined with more quantitative parameters and data obtained by different MRI sequences. Furthermore it automatically finds patterns based on which the classification can be done. This study is strongly related to the work by Hu et al. [1] and the aim was to reproduce their excellent results.

Chapter 2 will give an overview over the medical background concerning the problem of diagnosing pseudoprogression. Furthermore an introduction to some machine learning approaches will be given. The focus will be laid on Support Vector Machines, because SVM was the algorithm that was chosen for the classification task. In chapter 3 the used data set and method will be described in detail. The results will be presented in chapter 4. Finally in chapter 5 the results will be discussed and compared to the results of Hu et al. [1].

2 Theoretical Background

2.1 Medical Background

In the USA 7.28 per 100 000 individuals are annually diagnosed with a malignant tumor in the brain or central nervous system [11]. In comparison, in Austria in 2009 the age-adjusted incidence rate (based on the WHO standard population) for brain cancer was 4.9 per 100 000 women and men [12]. In the USA for Glioblastoma an incidence rate of 3.19 per 100 000 is reported [11]. Therefore nearly 45% of all malignant brain tumors are classified as Glioblastoma in the USA. Thus Glioblastoma is the most common malignant brain tumor [11].

2.1.1 Glioblastoma Multiforme and its Therapy

Glioblastomas multiforme (GBM) arise from glial cells and grow by diffuse infiltration into the white matter of the brain. GBM are characterized by their distinctive neovascularization and the disruption of the blood-brain barrier (BBB) [13]. GBM are assigned a WHO-grade IV and patients diagnosed with GBM have a poor prognosis [14]. The mean survival time is only about 15 months [13].

Current standard of therapy is resection, followed by a chemotherapy combined with radiotherapy according to the Stupp Protocol [15, 16, 17]. A successful treatment leads to a decrease of the tumor volume. In contrast to that tumor progression or recurrent tumor are characterized by an increase of the tumor volume [16]. The response to the therapy is assessed using contrast-enhanced magnetic resonance imaging (MRI). Malignant tumors show faster and higher levels of enhancement than normal tissue. The increased enhancement generally results from stronger vascularization, which leads to bigger cerebral blood volume, and a disrupted blood-brain barrier (BBB) [18].

The BBB is formed by specialized endothelial cells around the capillaries in the central nervous system. It restricts or slows the diffusion of some drugs and other chemical compounds, radioactive ions and disease-causing organisms, such as viruses, from the blood into the extravascular space. In tissue with disrupted BBB contrast agents can pass into the extravascular space and cause contrast enhancement on MRI. Based on the size of the contrast-enhancing regions, the Macdonald criteria are used to assess the clinical response of high-grade gliomas to therapy [19]. However, the contrast enhanced regions only represent regions of disrupted BBB and not the whole tumor volume.

Furthermore a disrupted BBB and consequently contrast enhancement not only occurs in tumor tissue but can also be the result of other treatment-related effects. Especially reactions to radiotherapy/radio-chemotherapy like radiation necrosis or the so-called pseudoprogression lead to contrast enhancement as well [20]. Thus the evaluation of the size of contrast-enhancing regions is not sufficient to assess the success of the therapy.

2.1.2 Radiation Necrosis

Radiation necrosis is a severe local tissue reaction to radiotherapy. On MRI it shows signs of a disrupted BBB, edema and mass effect [20]. Radiation necrosis can occur 3 months to several years after radiotherapy [20, 16]. The incidence of radio necrosis after radiotherapy is estimated at 3-24% and is suspected to be higher when it is combined with chemotherapy [20, 21]. The occurrence of radiation necrosis seems to be directly related to the radiation dose and size of the brain volume to which it is applied during therapy [20, 21].

The clinical course and symptoms of radiation necrosis are highly variable. Some patients even remain asymptomatic [20]. Radiation necrosis is normally irreversible and likely to progress, may even be fatal.

2.1.3 Pseudoprogression

Pseudoprogression typically occurs up to 3 months after therapy [20]. It shows increased contrast enhancement on MRI and is likely to be misinterpreted as tumor recurrence. In a study of Chamberlain et al. [22] 15 patients underwent surgery in the first 6 months after radiochemotherapy because of a suspected recurrence. In 7 of these 15 patients (46%) no histopathological evidence for tumor was found.

Pseudoprogression is thought to be a local tissue-reaction to the radiotherapy with signs of inflammation, edema and increased vascular permeability [20]. It is assumed that it is a self-limiting process. Patients often recover or stabilize spontaneously. Pseudoprogression is related to a better outcome and overall survival [2]. Therefore it is important not to mistake pseudoprogression for tumor recurrence and treatment failure. Otherwise the treatment that is working might be changed to one that might not work. Additionally after a potential change of the therapy the spontaneous improvement of pseudoprogression might be interpreted as a result of the new treatment [2].

2.1.4 Imaging

Conventional contrast enhanced MRI

As stated before, conventional contrast enhanced MRI is not suitable for the differentiation between progression and pseudoprogression, because both lead to increased enhancement. Currently the only possibility to distinguish pseudoprogression from progression in contrast enhanced MRI is to do follow-up examinations. If the size of the contrast enhanced region stabilizes or decreases on follow-up studies, it can be assumed to represent pseudoprogression. If it worsens, it represents tumor progression [17, 2].

Perfusion Imaging

To measure the perfusion dynamic susceptibility contrast-enhanced MRI (DSC-MRI) is performed. For that a contrast agent is injected into a peripheral vein in a reproducible manner [18]. The first passage of the bolus of the contrast agent is detected and its concentration versus time for each voxel is measured [23]. Based on the measured curves the relative cerebral blood volume (CBV), relative cerebral blood flow (CBF), mean transit time (MTT) and time to peak concentration (TTP) in each voxel can be estimated [24, 25]. These are relative measures and do not allow absolute quantification. DSC-MRI assumes that the blood-brain barrier is intact and that the signal reduction observed in DSC-MRI results entirely from contrast agent within the blood vessels. However in regions of disrupted BBB that is generally not true. Contrast agent diffuses into extravascular space and causes signal changes that lead to an underestimation of CBV [2, 25, 26]. There are techniques to correct the effects of leaked contrast media. One possibility to reduce the leakage error is the injection of a small dose of contrast agent prior to the actual measurement. Another way to reduce the error is the reduction of the flip angle, which however results in lower signal-noise-ratio (SNR) [25, 2]. Reduction of flip angle is simple and is very effective with gradient-echo sequences [25, 2].

DSC-MRI has been shown to be useful in the differentiation of tumor recurrence and pseudoprogression [27, 28, 2, 21]. Generally tumor recurrence is associated with stronger perfusion due to increased metabolic activity and neoagiogenesis. Higher relative CBV and relative CBF has been seen in recurrent tumor, whereas radio necrosis and pseudoprogression show decreased relative CBV and relative CBF.

Diffusion-Weighted Imaging

All molecules undergo random, Brownian motion. Free particles, such as water molecules, are therefore subject to diffusion processes. The speed of the diffusion process is characterized by the diffusion coefficient also referred to as diffusivity. It is influenced by the characteristics and geometrical structure of the environment (e.g. viscosity, cell membranes, macromolecules). Diffusion-weighted imaging (DWI) detects differences in the diffusivity of water molecules in the different tissues. Based on DWI the apparent diffusion coefficient (ADC) is calculated for each voxel [29].

Tumor progression is associated with lower ADC values compared to the ADC values of necrotic tissue and pseudoprogression [30, 17, 26]. This can be explained by the increased cellular density, generally found in tumor tissue, which leads to lower diffusivity of water molecules [30].

2.2 Machine Learning

This thesis was about the question whether an automated distinction between tumor recurrence/progression and pseudoprogression based on a machine learning approach is possible. Basically the differentiation can be seen as a classification problem with two classes, progression and pseudoprogression. Both classes are not well distinguishable by physicians on the basis of the routinely acquired MRI data. However a machine learning algorithm possibly can learn a pattern in the MRI data based on which it can make a distinction between the two classes.

In the following, some machine learning methods with focus on supervised learning will be presented. The field of machine learning is a very wide field and it is impossible to discuss it to the whole extent in this thesis. However the aim of the following section is to give an overview of the most basic and important methods. To keep it simple only two-class classification methods will be considered.

2.2.1 Data Representation

A crucial task in machine learning is the choice of data representation. Each instance, also called example, sample, observation or data point, has to be described in some way so that it can be computationally processed. Generally the instances are described by an input vector, that is also often referred to as feature vector. The components of the input vector represent the so-called features, attributes or input variables [31]. The ideal choice of features makes the distinction between two instances of different categories an easy task.

2.2.2 Naïve Bayes

The naïve Bayes classifier is a classification method based on probability theory, more precisely it is based on the Bayes Theorem:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{2.1}$$

Despite its simplicity, naïve Bayes often outperforms more sophisticated classification methods [32, 33].

Given a vector $\mathbf{x}$ describing one instance with D elements, that each describe one attribute of the instance, and given one class y_k of the K classes then the following applies:

$$P(y_k|\mathbf{x}) = \frac{p(\mathbf{x}|y_k)P(y_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y_k)P(y_k)}{\sum\limits_{l=1}^{K} p(\mathbf{x}|y_l)P(y_l)}. \tag{2.2}$$

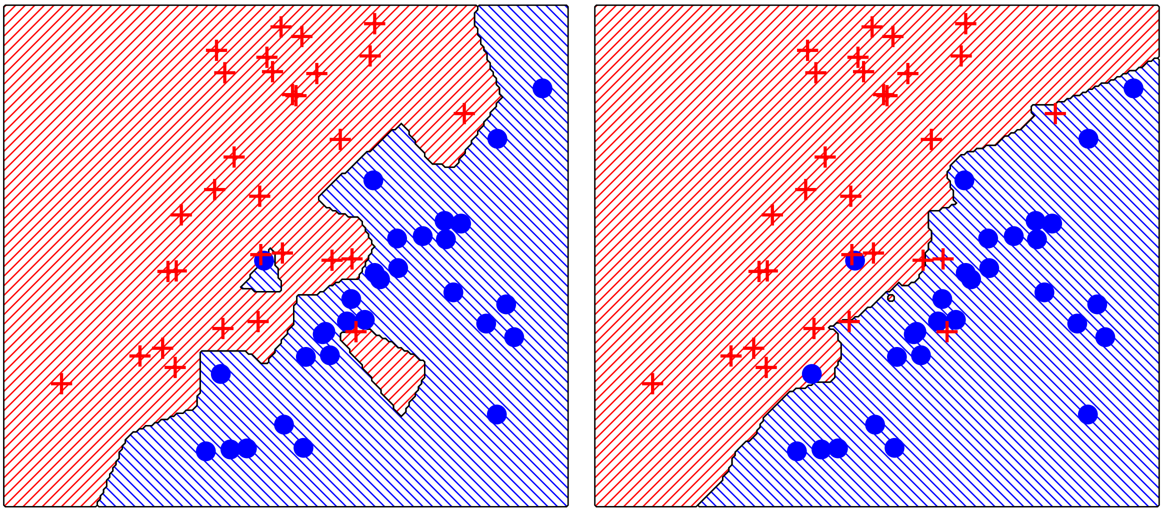
The naïve Bayes model assumes that all attributes are independent from each other and only depend on the output variable. This assumption is generally not true, but it simplifies the problem enormously [34]. Assuming stochastic independence the following applies:

$$p(\mathbf{x}|y_l) = \prod_{i=1}^{D} p(x_i|y_l). \tag{2.3}$$

Assuming that the probability distribution for each attribute $p(x_i|y_k)$ is known, the probability of an instance to be of the class y_k given a certain attribute vector $\mathbf{x}$ is easy to calculate. The substitution of equation 2.3 into equation 2.2 results in the rule of the naïve Bayes model:

$$P(y_k|\mathbf{x}) = \frac{P(y_k)\prod\limits_{i=1}^{D} p(x_i|y_k)}{\sum\limits_{l=1}^{K} P(y_l)\prod\limits_{i=1}^{D} p(x_i|y_l)} \tag{2.4}$$

Subsequently with equation 2.4 it is possible to calculate the probability that a given attribute vector $\mathbf{x}$ is of the class y_k. The naïve Bayes classifier predicts the class with the highest probability to be the class of the instance. The biggest advantage of the naïve Bayes classifier is its efficiency. That is, if you already have a good model, the prediction of the class for a new instance is fast. However, to get a good estimate of the distribution, you generally need a lot of data.



(a) 1-NN Classifier

(b) 5-NN Classifier

Figure 2.1: KNN-Classifier classifier applied to a toy example using Euclidean distances. (a) shows the resulting decision boundary using $k = 1$. (b) shows the resulting decision boundary using $k = 5$. $k = 5$ results in a more general decision boundary at the cost of more misclassified training data.

2.2.3 Nearest-Neighbor Classifier

A considerably more basic method is the nearest-neighbor classifier. It simply assigns to an instance with unknown class the class of the nearest, that is the most similar, instance of the training data. Hence to classify a new instance only the distances to all training data have to be assessed. Possible choices for the distance would be the Euclidean or the Mahalanobis distance [34]. A variation of the nearest-neighbor classifier is the k-nearest-neighbor (KNN) classifier. It assigns to an unclassified instance the class of the majority of the k nearest neighbors. k-nearest-neighbor works better on noisy data than the basic nearest-neighbor classifier. Figure 2.1 shows a toy example using two different values for k. The nearest neighbor algorithms show excellent performances when applied to suitable data and used with a good distance measure [35].

2.2.4 Linear Discriminant Analysis

The linear discriminant analysis (LDA) is a method that searches for a linear function (see equation 2.5) that separates the two classes optimally. In the 2-dimensional space ($D = 2$) the decision boundary would be a line. In a higher dimensional space ($D \geq 3$) the data would be separated by a hyperplane.

$$\mathbf{w}^T \cdot \mathbf{x} + b = 0, \quad \mathbf{w} \in \mathbb{R}^D, \quad b \in \mathbb{R}. \tag{2.5}$$

The decision function that decides whether a data point $\mathbf{x}_n$ lies in the one or in the other half-space is then defined as

$$f(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}^T \cdot \mathbf{x} + b). \tag{2.6}$$

To find the linear function a direction, defined by the vector $\mathbf{w}$, is calculated such that when the data points are projected onto $\mathbf{w}$, the classes are as well separated as possible [35]. The projection of a point $\mathbf{x}$ onto $\mathbf{w}$ is defined as $x = \mathbf{w}^T\mathbf{x}$. After the projection of the data points onto $\mathbf{w}$ the projected means m_k of the classes should be as far apart as possible and the data points of each class should be in regions as small as possible. Hence the difference between the two projected means $|m_1 - m_2|$ is to be maximized and the scatter s within each class is to be minimized. The scatter s represents an equivalent of the variance (σ^2) and is defined as [35]

$$s^2 = \sum_n (\mathbf{w}^T\mathbf{x}_n - m)^2. \tag{2.7}$$

Fisher's linear discriminant is defined as the linear function $\mathbf{w}^T\mathbf{x}$ that maximizes the criterion function $J(\mathbf{w})$ [35, 36]:

$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}. \tag{2.8}$$

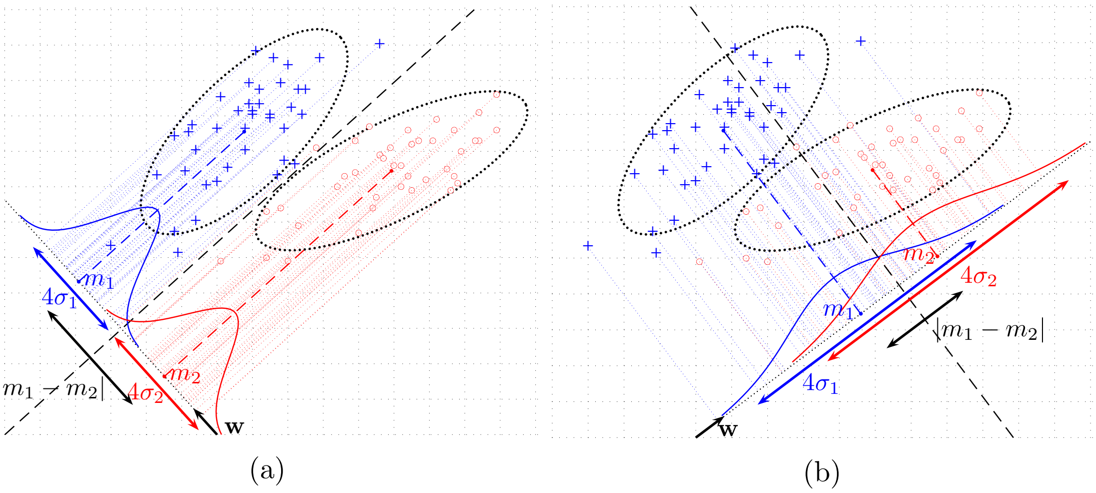


Figure 2.2: An example of a two-class classification problem. The groups follow bivariate normal distributions. The ellipses indicate the 2σ-distance to the mean of the Gaussian distributions. (a) shows the 2 groups divided by a linear decision boundary defined by the vector $\mathbf{w}$ obtained by a LDA. The groups are separated nearly optimally. (b) shows the same groups and a linear decision boundary resulting from a bad choice of $\mathbf{w}$.

Note that $(s_1^2 + s_2^2)$ is also called the total within-class scatter of the projected samples [36]. However Fisher's linear discriminant is not a discriminant but rather defining a direction for a projection to one dimension on which the data is separable. To make a separation of the projected data points a threshold b in that one-dimensional subspace must be found. b also refers to the distance of the decision boundary to the origin (see equation 2.5). For estimating an optimal b any classification method can be used [35]. One possible choice for b would be simply the mean of the two projected class means m_1 and m_2 [34].

$$b = \frac{m_1 + m_2}{2} \tag{2.9}$$

Using equation 2.9 for determining b works only well if the classes nearly follow Gaussian distributions. In some cases it may be reasonably to chose b by empirically minimizing the training error for a given dataset [32]. Figure 2.2 shows an example for a two-class classification problem. It illustrates how the two groups can be separated by a line defined by a vector $\mathbf{w}$ and a threshold b obtained by equation 2.9.

An advantage of the LDA is that it does not matter whether the attributes are correlated [34]. But LDA fails if the distributions of the classes are multimodal and highly overlapping. In such a case even the "best" $\mathbf{w}$ will not provide a useful separation [36].

2.2.5 Support Vector Machines

The support vector machine (SVM) is a classifier which, like the linear discriminant analysis (LDA), uses linear discriminant hyperplanes (see equation 2.5) to separate the data into two half-spaces. The hyperplane should be chosen in such a way that the class label becomes $y_n = +1$ for the one class in the one half-space $(\mathbf{w}^T \cdot \mathbf{x}_n + b \geq 0)$ and $y_n = -1$ for the second class in the other half-space $(\mathbf{w}^T \cdot \mathbf{x}_n + b < 0)$. The corresponding decision function equals to [37]:

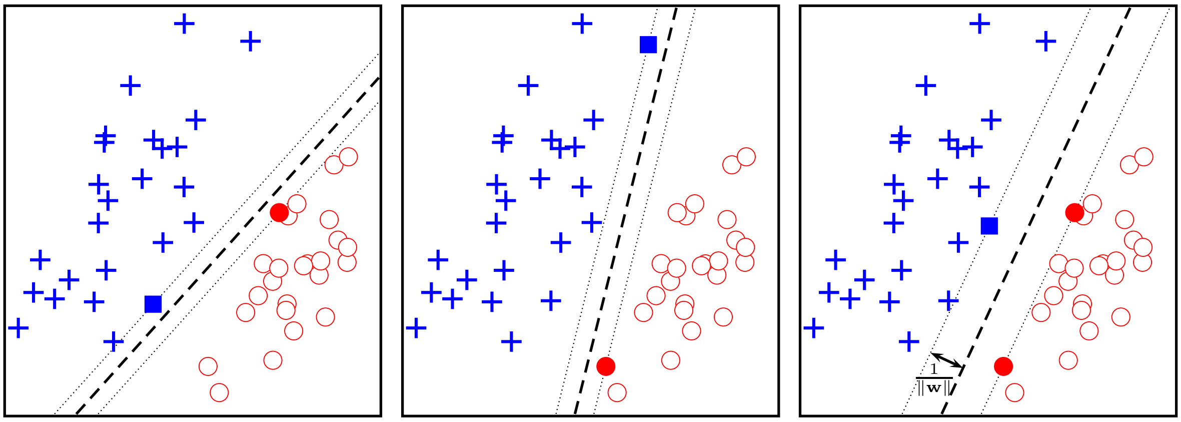
$$f(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}^T \cdot \mathbf{x} + b). \tag{2.10}$$

To find a hyperplane that correctly classifies all data points the data must be linearly separable.

In contrast to the LDA the SVM tries two find the hyperplane with maximal distance to the closest data point [37, 38]. That means the margin of the decision boundary is to be maximized. Figure 2.3 shows an example of a two-class classification problem and three possible linear decision boundaries. The optimal boundary is the one with maximal margin.

The distance of a point $\mathbf{x}_n$ to a hyperplane is given by $\left|\mathbf{w}^T \cdot \mathbf{x}_n + b\right| / \|\mathbf{w}\|$. Therefore the margin of the hyperplane is

$$\min_{n=1,\dots,N} \frac{|\mathbf{w} \cdot \mathbf{x}_n + b|}{\|\mathbf{w}\|}. \tag{2.11}$$



(a) (b) (c)

Figure 2.3: An example of a two-class classification problem. The two groups are linearly separable by various lines. (a) and (b) show lines that separate the data without classification errors. However they are not considered to be optimal, because there still exist decision boundaries with larger margins. The filled circles and rectangles indicate the data points that lie on the margin of the decision boundaries. (c) shows the optimal decision boundary with maximal margin. It is the line with maximal distance to the nearest data points. The points on the margin represent the support vectors. They define the line.

To simplify that statement it is reasonable to require $\min\limits_{n=1,\ldots,N} \left|\mathbf{w}^T \cdot \mathbf{x}_n + b\right| = 1$, which simply results in scaling $\mathbf{w}$ such that the margin equals to $\frac{1}{\|\mathbf{w}\|}$ [38]. That way the condition that the separating hyperplane has to comply can be written as

$$y_n(\mathbf{w}^T \cdot \mathbf{x}_n + b) \geq 1. \tag{2.12}$$

The problem of maximizing the margin therefore reduces to [38]

$$\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|} \qquad \text{s.t. } y_n(\mathbf{w}^T \cdot \mathbf{x}_n + b) \geq 1 \text{ for all } n \tag{2.13}$$

or equivalently

$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2 \qquad \text{s.t. } y_n(\mathbf{w}^T \cdot \mathbf{x}_n + b) \geq 1 \text{ for all } n. \tag{2.14}$$

$\min \frac{1}{2} \|\mathbf{w}\|^2$ is called the *objective function*. Together with the *inequality constraints* the objective function forms the so-called constrained optimization problem as written in equation 2.14. Such problems are solved by introducing Lagrange multipliers $\alpha_n \geq 0$ and optimizing a Lagrangian function:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^{N} \alpha_n \left(y_n(\mathbf{x}_n^T \cdot \mathbf{w} + b) - 1 \right). \tag{2.15}$$

For further information about the approach to solve the optimization problem using Lagrangian see [37, 35]. The following terms result from the statement that the derivatives of L with respect to b and $\mathbf{w}$ must be zero [37]:

$$\sum_{n=1}^{N} \alpha_n y_n = 0, \tag{2.16}$$

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n. \tag{2.17}$$

Hence $\mathbf{w}$ is defined by the subset of the training points $\mathbf{x}_n$, that have a non-zero α_n. Those are the so-called *support vectors* (SVs).

If the data is not linearly separable there is no solution for the optimization problem 2.14 [38, 34]. Such a classifier is called a hard margin classifier.

Soft Margin Classifier

For the case that a linear separation is not possible the non-negative slack variables ξ_n are introduced. ξ_n relax the constraints of equation 2.12 as follows:

$$y_n(\mathbf{w}^T \cdot \mathbf{x}_n + b) \geq 1 - \xi_n. \tag{2.18}$$

In a non-linearly–separable case, no matter which hyperplane is chosen, there always will be instances that lie on the wrong side of the hyperplane or lie on the right side but within the margin, i.e. too close to the hyperplane. After the introduction of ξ_n in equation 2.18 a solution is possible, even though some training instances may be misclassified. Given an arbitrary hyperplane with arbitrary $\mathbf{w}$ and b the constraints 2.18 can be satisfied just by making ξ_n large enough. Consequently large ξ_n have to be penalized [38, 34].

To achieve the penalization of large ξ_n the optimization problem in 2.14 is modified as follows [38]:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2} \left\| \mathbf{w} \right\|^2 + \frac{C}{N} \sum_{n=1}^{N} \xi_n \\ &\text{s.t. } y_n(\mathbf{w}^T \cdot \mathbf{x}_n + b) \geq 1 - \xi_n \quad \text{for all } n \\ &\quad\ \xi_n \geq 0, \end{aligned} \tag{2.19}$$

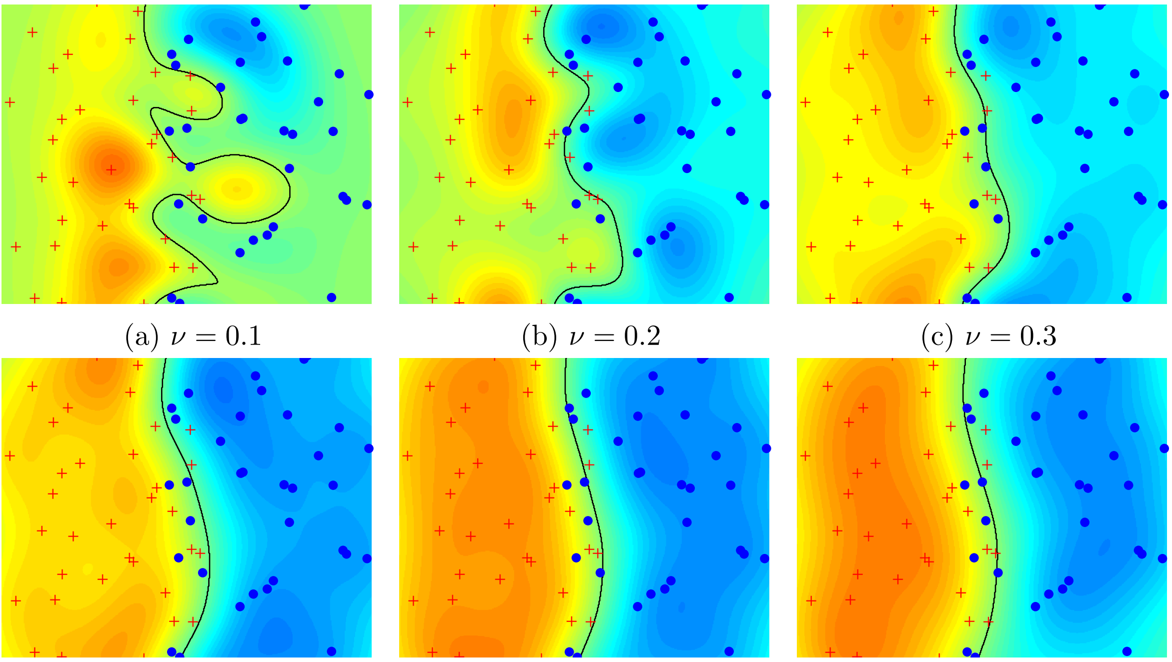
9 where $C > 0$ is a regularization parameter trading off generalizability, i.e a large margin, and training error, i.e. the number of nonseparable points. Note, that not only misclassified points are penalized but also points in the margin [38, 35]. To determine $\mathbf{w}$ and b equation 2.19 is minimized under the given constraints using a suitable optimiza-

tion method like quadratic programming [38, 35]. Solving the optimization problem 2.19 shows that only a subset of the data points, namely the points that lie on or within the margin or on the wrong side of the boundary, define $\mathbf{w}$. These are the so called support vectors.

ν-SVM

An equivalent to the soft margin SVM is the ν-SVM [37, 39, 38]. The primal problem for the ν-SVM can be written as [37]

$$\begin{aligned} &\min_{\mathbf{w},b,\xi,\rho} \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{N}\sum_{n=1}^{N}\xi_n \\ &\text{s.t. } y_n(\mathbf{w}^T \cdot \mathbf{x}_n + b) \geq \rho - \xi_n \quad \text{for all } i \\ &\qquad \xi_n \geq 0, \text{ and } \rho \geq 0. \end{aligned} \tag{2.20}$$


(a) $\nu = 0.1$

(b) $\nu = 0.2$

(c) $\nu = 0.3$

(d) $\nu = 0.5$

(e) $\nu = 0.7$

(f) $\nu = 0.9$

Figure 2.4: An example of two groups that are separated using a ν-SVM with a Gaussian kernel $k(x_n, x) = \exp(-\|x_n - x\|^2)$. The colored background indicates the value inside the sign-function of the decision function (see equation 2.22). In (a) all training data points but one are correctly classified. However the decision boundary seems to be quite overfitted and is likely to fail on new data. In (b) to (f) ν is more and more increased. More classification errors and more support vectors within the margin are allowed. This leads to a smoother and more general decision boundary.

ρ is a parameter that scales the margin. The margin now equals to $\rho / \|\mathbf{w}\|$. ν has been shown to be a lower bound on the fraction of support vectors and an upper bound on the fraction of instances having margin errors. Hence ν controls the number of support vectors and margin errors. When increasing ν more errors are allowed and the margin is increased [37]. Figure 2.4 shows the effects of various values for ν on the decision boundaries of a ν-SVM applied on a two-class classification problem. Compared to the parameter C in soft margin SVMs tuning the parameter ν is thought to be more intuitive and therefore the ν-SVM is often the preferred method [35].

Feature Space

Is the data not linearly separable you can try to find a nonlinear discriminant, which is generally not easy. Another possibility to separate the data is to map the data from the input space $\mathcal{X}$ to a higher dimensional space $\mathcal{H}$. The mapping to the so called feature space $\mathcal{H}$ is done by a nonlinear transformation $\mathbf{\Phi}$ [37]. In the feature space it may be possible to find a linear separating hyperplane. In the original input space such a hyperplane yields a nonlinear decision boundary [35]. Figure 2.5 shows an example of a non-separable problem that can be linearly separated when mapped with $\mathbf{\Phi}$ into a proper feature space.

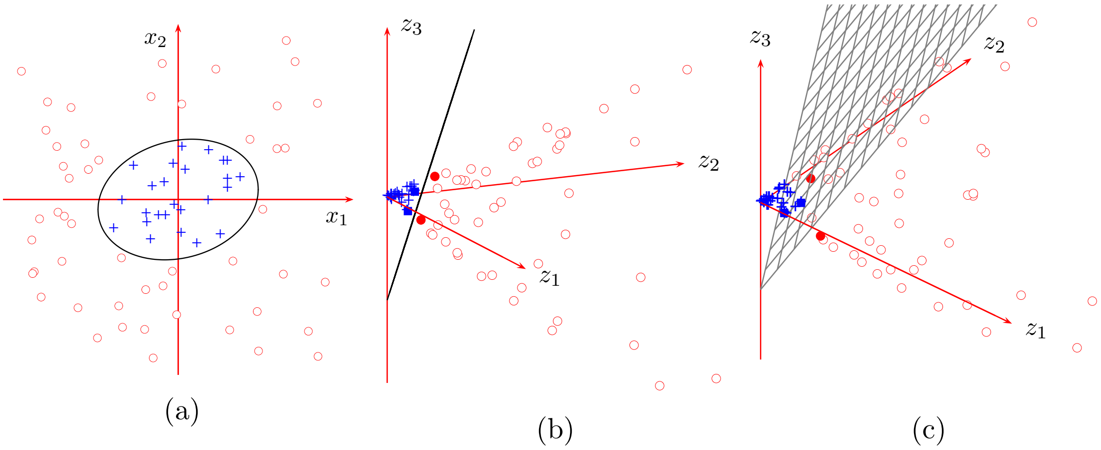


Figure 2.5: An example of a two-class classification problem mapped into feature space. The data points $x = (x_1, x_2)$ are mapped into feature space via the nonlinear map $\mathbf{\Phi}(x) = (z_1, z_2, z_3) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ (adopted from [37]). Afterwards the optimal separating plane was calculated. (a) shows the data in input space. (b) shows the data in feature space from an angle where the separating plane becomes a line. It is evident that the data is linearly separable in feature space. The filled circles and rectangles indicate the support vectors, that define the plane. (c) shows the separating plane from a different angle. When the plane is transformed back into the input space it becomes an ellipse as shown in (a).

Kernel Trick

The kernel trick refers to a technique that avoids the mapping of each data point via $\mathbf{\Phi}$ [37]. It has been shown that the dot product of the mapped data points $\mathbf{\Phi}(x_1) = \mathbf{x}_1$ and $\mathbf{\Phi}(x_2) = \mathbf{x}_2$ can be replaced by a kernel function $k(x_1, x_2)$ between the instances x_1 and x_2 in the input space [37]:

$$k(x_1, x_2) = \mathbf{\Phi}(x_1)^T \cdot \mathbf{\Phi}(x_2) \tag{2.21}$$

Substituting the dot products by the kernel simplifies the computation of the solution for the optimization problem and subsequently the computation of the decision boundary enormously. For further explanations about the application of the kernel trick see [37] or [35].

The kernel function can also be incorporated in the decision function $f(x)$ by plugging equation 2.17 into the equation of the decision function 2.10 and using equation 2.21 [37]:

$$\begin{aligned} f(x) &= \operatorname{sgn}\left(\mathbf{w}^T \cdot \mathbf{\Phi}(x) + b\right) = \operatorname{sgn}\left(\sum_{n=1}^{N} \alpha_n y_n \; \mathbf{\Phi}(x_n)^T \cdot \mathbf{\Phi}(x) + b\right) \\ &= \operatorname{sgn}\left(\sum_{n=1}^{N} \alpha_n y_n k(x_n, x) + b\right) \end{aligned} \tag{2.22}$$

where α_n are the Lagrange multipliers and y_n are the class labels. Hence the decision function $f(x)$ can be thought of as a function based on a linear combination of kernels, which are centered on the training points with non-zero α_n, the Support Vectors, and which are possibly negated by the class label y_n of those. A very popular choice for the kernel is the Gaussian radial-basis function (RBF) [37]:

$$k(x_n, x) = e^{-\frac{\|x_n - x\|^2}{2\sigma^2}}, \quad \sigma > 0 \tag{2.23}$$

The RBF defines a spherical kernel where x_n is the center and σ defines the radius. A big σ results in a smoother decision boundary and therefore in a more general solution. On the contrary a small σ allows to catch more of the training data in the right class.

2.2.6 One-class Classification and Novelty Detection

One-class classification can also be considered as novelty or outlier detection [35]. The basic task of novelty detection is to determine whether a new observation comes from the distribution on which the learner was trained or whether it shows a novel unknown pattern and therefore does probably not come from that initial distribution [40, 41, 42]. Those observations are considered as atypical and are classified as not being part of the initial class.

Parzen Window Estimator

For novelty detection a representation of normality is required. The probability density function $p(x)$ for example would be a good description for the normal data. Any data x for which $p(x)$ is below a specific threshold can then be considered as novel [40]. The estimation of the probability density function (PDF) is for example done by the Parzen window estimator. The Parzen window estimator estimates the PDF by the linear combination of N kernels $k(\mathbf{x}, \mathbf{x}_n)$ [37]:

$$p(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n), \tag{2.24}$$

where x_n are the known data points of the training set and N is the size of the training set. A common choice for the kernel is again the Gaussian kernel because it results in a smooth PDF estimate. This leads to the following density model [43]:

$$p(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \exp\left\{ -\frac{\|\mathbf{x} - \mathbf{x}_n\|^2}{2\sigma^2} \right\}, \tag{2.25}$$

where σ corresponds to the width or standard deviation of the Gaussian kernels and d is the dimension of the input space. In figure 2.6 some examples of Parzen-window estimates are shown. It shows the influence of the parameter σ and the number of samples N.

One-class SVM

The one-class SVM (OC-SVM) is another method for novelty detection [41]. It computes a region as small as possible that includes most of the data points of the training set. For all points in this region the decision function f (see equation 2.26) takes the value $+1$. Anywhere else f takes the value -1.

$$f(x) = \operatorname{sgn}\left(\mathbf{w}^T \cdot \mathbf{\Phi}(x) - \rho\right) \tag{2.26}$$

To find that optimal region the training data is mapped into the feature space $\mathcal{H}$. The region is then defined by a hyperplane that separates the data set from the origin in the feature space with maximal distance to the origin. The following quadratic program is solved to find the optimal separating hyperplane [41, 37]:

$$\begin{gathered}
\min_{\mathbf{w} \in \mathcal{H}, \xi \in \mathbb{R}^N, \rho \in \mathbb{R}} \frac{1}{2} \|\mathbf{w}\|^2 - \rho + \frac{1}{\nu N} \sum_{n=1}^{N} \xi_n, \\
\text{subject to } \; \mathbf{w}^T \cdot \mathbf{\Phi}(x_n) \geq \rho - \xi_n, \; \xi_n \geq 0.
\end{gathered} \tag{2.27}$$

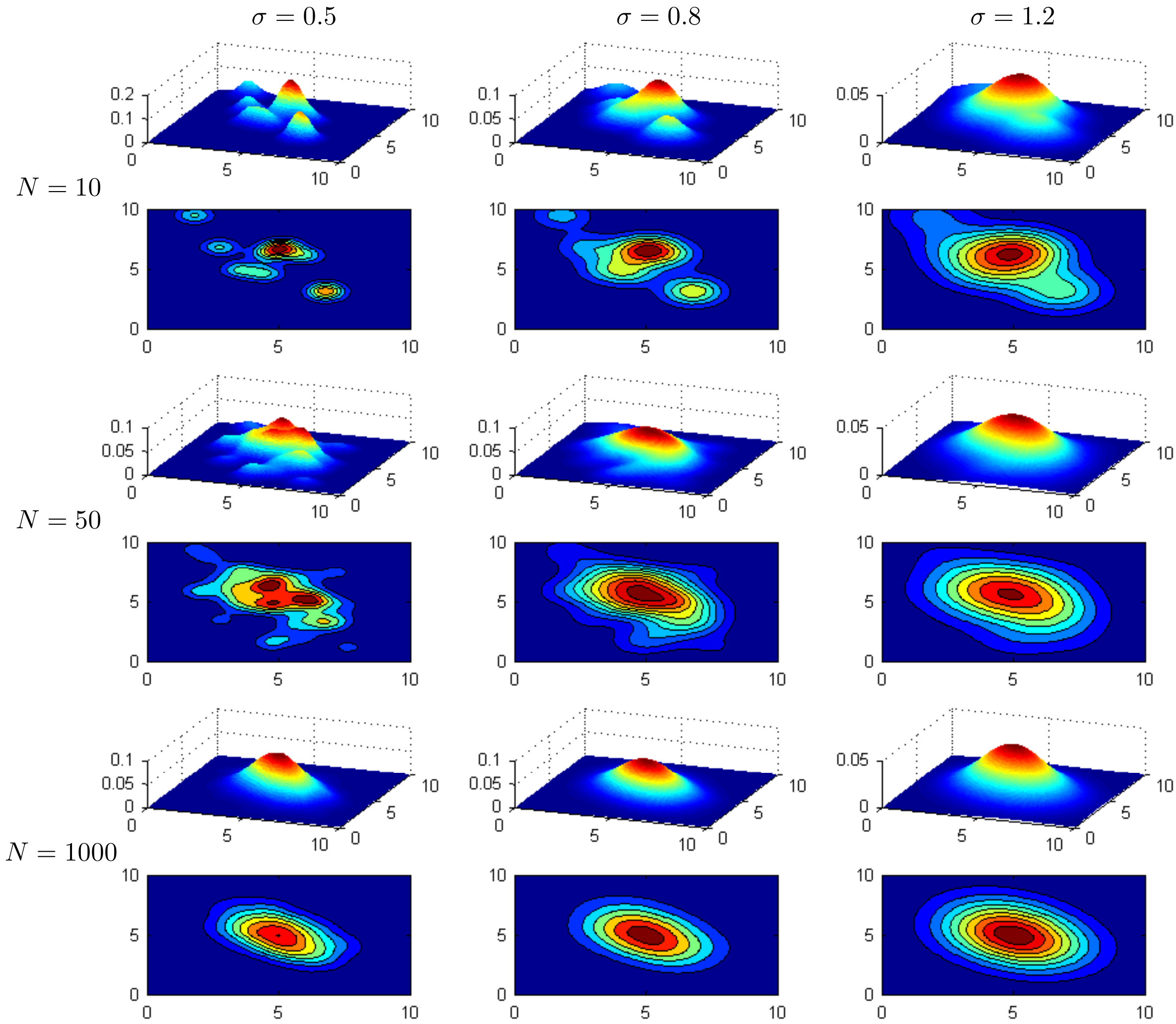


Figure 2.6: Parzen-window estimates of a bivariate normal density using different kernel widths σ and numbers of samples N. A large number of samples ($N = 1000$) leads to almost the same distribution, regardless of the kernel width, and matches the true distribution very well.

The distance of the hyperplane to the origin is $\rho / \|\mathbf{w}\|$. Consequently the optimal hyperplane is obtained by minimizing $\|\mathbf{w}\|$. Analogously to the ν-SVM (see section 2.2.5) the introduction of slack variables ξ_n allow that data points lie between the hyperplane and the origin. Such points have non-zero ξ_n and therefore non-zero ξ_n are penalized. The parameter ν controls to which extent ξ_n are penalized and consequently controls the fraction of the points fall on the wrong side of the hyperplane. The decision function (see equation 2.26) of the OC-SVM reduces to a thresholded Parzen window estimator, provided that the OC-SVM is applied using a kernel with integral 1, such as the Gaussian kernel, and that the parameter $\nu = 1$ [41, 37]. For $\nu < 1$ the decision function still is a thresholded density but it is obtained only by a subset of the training data, the support vectors (SVs) [37].

Safty Margin in OC-SVMs

In two-class ν-SVMs the parameter ν controls the fraction of SVs and data points with margin errors. A larger ν leads to a more generally shaped decision boundary and to a larger margin. However more data points with margin errors do not lead automatically to more training errors. Because in two-class ν-SVMs data points within the margin can still lie on the right side of the hyperplane and are therefore possibly classified correctly. On the contrary, in OC-SVMs this is not the case. The data points with margin errors are considered as outliers and fall on the wrong side of the hyperplane. They are therefore classified wrongly.

In OC-SVMs a larger ν does not lead to a larger margin between two classes, simply because there is only one class. Instead of maximizing the margin between two classes the distance of the hyperplane to the origin in feature space is maximized. A larger ν has not only the result of a more generally shaped decision boundary but also results in a larger distance of the hyperplane in feature space to the origin. Hence a larger ν always has the effect of reducing the size of the region, that is estimated to contain the training data.

Consequently it makes sense to introduce a "safety margin" in OC-SVMs. To increase the generalizability of the classifier Schölkopf et al. [44, 37] suggest to make the region, that is estimated to contain the training data, slightly larger than it is estimated by the algorithm. They do this by reducing ρ in the decision function 2.26 by some factor γ. To avoid confusion with the kernel parameter used by the LIBSVM package, which is introduced later, this parameter will here be called T, because it controls the threshold of the thresholded density, that is estimated by the OC-SVM. Hence the decision function becomes

$$f(x) = \operatorname{sgn}\left(\mathbf{w}^T \cdot \mathbf{\Phi}(x) - (\rho - T)\right), T > 0. \tag{2.28}$$

In the end the kernelized version of the decision function equals to

$$f(x) = \operatorname{sgn}\left(\sum_{n=1}^{N} \alpha_n k(x_n, x) - (\rho - T)\right), T > 0. \tag{2.29}$$

Reducing ρ has the effect of shifting the hyperplane in feature space towards the origin. The distance to the origin is now given by $\frac{\rho - T}{\|\mathbf{w}\|}$. Taken together, in OC-SVMs a larger ν results in a more general shaped decision boundary on the cost of a smaller region that is estimated to contain the training data. Therefore the region is made larger again by the parameter T.

2.2.7 Neural Networks

The term “neural networks” comes from the attempt to model the brain as an information processing system by means of mathematical models [43]. However the approach has shown to be also suitable for the application as a learning method. In the context of machine learning the most successful implementation of a neural network is the “feed-forward neural network”, also called the “multilayer perceptron” [43].

In comparison to the support vector machine the multilayer perceptron can be fast in processing new data, because of its possibly better compactness, although it provides the same generalization performance. The drawback is that the objective function, which is to be optimized in neural networks, is not a convex function of model parameters. Hence the training is computationally costly and it is not certain whether the optimal solution to the problem is found [43].

The SVM basically first defines basis functions that are centered on the training data points and then selects during training a subset of these. On the contrary neural networks start with a fixed number of basis functions, which are however adaptive.

Neural networks consist of several layers of nodes. The nodes of the network are often named after the basic information processing units of the brain, the neurons. Figure 2.7 shows a schematic of an example of a single neuron. The neurons “fire” when the signals that are passed to them are in sum larger then a certain threshold. In the context of machine learning instead of a step function the output signal of a neuron is usually controlled by a differentiable activation function.

The most basic implementation of a neural network has three layers of units: an input layer, a hidden layer and the output layer [32]. For a two-class classification problem there are 2 units in the output layer. The input layer consists of the input variables. The

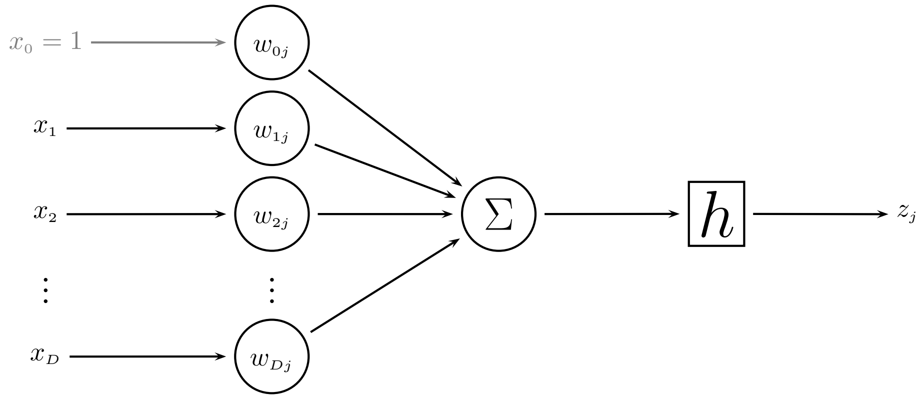


Figure 2.7: Schematic of a neuron. The input variables x_i are weighted with the parameters w_{ij}. Afterwards they are linearly combined and transformed by the nonlinear activation function h. z_j is the output of the neuron. x_0 is fixed to 1 because w_{0j} represents the bias.

hidden layer lies between the output and the input layer. The units of the hidden layer represent the adaptive basis functions. Figure 2.8 shows the schematic of a feed-forward neural network.

A unit of the hidden layer with subscript j computes a linear combination of the input variables x_i with $i = 1, \ldots, D$, where the input variables are weighted with the parameters $w_{ji}^{(1)}$ and a bias $w_{j0}^{(1)}$ is added. The superscript (1) denotes that the weight corresponds to the first layer. The result is transformed by a differentiable, nonlinear activation function h. So the output of a unit in the hidden layer is given by [43]:

$$z_j = h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right). \tag{2.30}$$

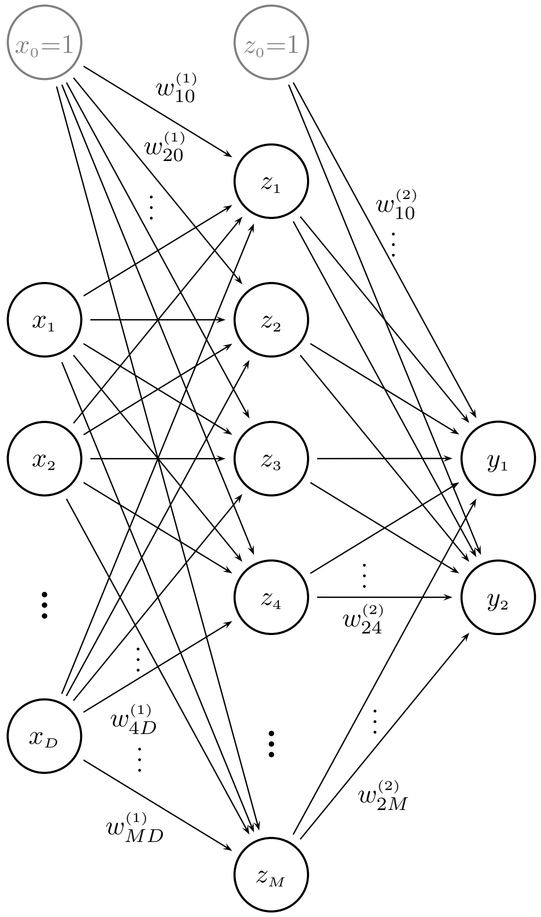


Figure 2.8: Schematic of a feed-forward neural network. The network has three layers of units. Hence, such a network structure is sometimes called a "three-layer network". Sometimes the same structure is also called "single-hidden-layer" network, because there is only one hidden layer. When referring to the number of layers of adaptive weights the structure can be called "two-layer network" as well [43]. x_i are the input variables, z_j are output signals of the hidden units and y_k are the output variables of the network. $w_{ji}^{(1)}$ and $w_{kl}^{(2)}$ are the weights of the first and the second layer respectively.

For the activation functions h generally sigmoidal functions are chosen [43]. Logistic sigmoid or tanh-function are a common choices. The outputs of the hidden layer are again linearly combined and transformed by an activation function σ to give the network outputs as follows [43]:

$$y_k = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}\right). \tag{2.31}$$

The bias variable $w_{j0}^{(1)}$ can be incorporated into the set of weight parameters when introducing an additional input variable x_0 and setting it to $x_0 = 1$. Similarly the bias variable $w_{k0}^{(2)}$ of the second layer can be absorbed into the set of second-layer weights $w_{kj}^{(2)}$. In the end the overall network function becomes

$$y_k(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{j=0}^{M} w_{kj}^{(2)} h\left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i\right)\right) \tag{2.32}$$

where σ is a logistic sigmoid function (illustrated in Figure 2.9):

$$\sigma(a) = \frac{1}{1 + \exp(-a)}. \tag{2.33}$$

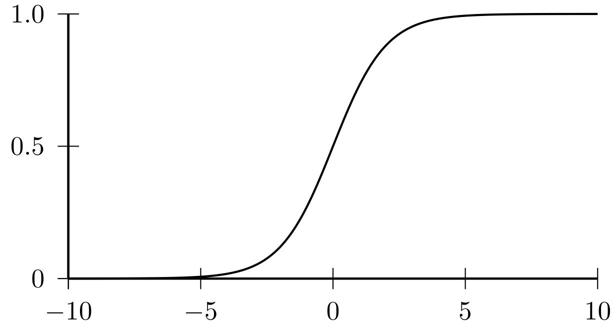


Figure 2.9: The curve shows the sigmoid function $\sigma(a)$ as in equation 2.33.

Network Training

For training a neural network two aspects have to be solved: identification of the structure of the network and learning the weight parameters. The problem of determining the weights of a fixed structure is solved. However the structure of the network must generally be found by experimentation and experience [33].

Given a fixed network structure the weights $\mathbf{w}$ have to be adjusted such that the predicted output variables $\mathbf{y} = (y_1, \dots, y_K)$ for an input vector $\mathbf{x}_n$ match the target values as close as possible. The target vector $\mathbf{t}_n$ could for example describe a class to which the instance $\mathbf{x}_n$ is known to belong. Hence, the error between output and target should be minimal, so an error function has to be minimized. For example, the sum-of-squares error function $E(\mathbf{w})$, as written in equation 2.34, can be minimized with respect to the weights $\mathbf{w}$ using gradient descent [43] to obtain optimal weights.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \|\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n\|^2. \tag{2.34}$$

A common algorithm to evaluate the gradient of the error function $E(\mathbf{w})$ for a feed-forward neural network is known as "error backpropagation". For further information about this algorithm see [43, 36].

2.2.8 Summary

The range of machine learning methods is a lot larger than presented here and most of the methods are a lot more sophisticated than the examples of this overview. The optimal choice concerning the learning algorithm is strongly dependent on the problem and the data that is available.

Naïve Bayes is simple, but a large number of data is required to estimate the probability distribution of the input variables. Furthermore independence of the input variables is required. Concerning the differentiation of progression and pseudoprogression both independence of the input variables and a large number of data is not given. Therefore Naïve Bayes is not an appropriate algorithm to solve this classification problem.

When the classes highly overlap simple algorithms like K-Nearest-Neighbor (KNN) cannot be expected to perform well. Linear Discriminant analysis (LDA) only considers linear decision boundaries in input space. It does not work well on data that is not normally distributed and is highly overlapping. Both can be expected from the MRI data concerning the assessment of treatment response of Glioblastoma. Consequently KNN and LDA are not appropriate choices for the differentiation of pseudoprogression and progression.

However, the Support Vector Machine (SVM) is a promising approach. Not much prior knowledge about the data is needed, to apply a SVM to a classification problem. Nevertheless the choice of the kernel and the regularization parameter ν or C is critical. There is no general guideline of how to find those parameters. The radial basis function (equation 2.23) is a reasonable first choice [45]. Advantage of the RBF is that only one kernel parameter, the kernel width, has to be selected. SVMs generalize well when appropriate parameters are found.

One-class classification makes sense when there is only one specific class of interest that is to be separated from several other classes. This is an advantage when the other classes all together are very heterogeneous. Using two-class SVM more data would be necessary to model the other classes accurately in the training process. One-class classification instead only needs data from the one class of interest for training.

Neural networks are prone to overfitting, especially when not much data is available. Since there is not much data available, concerning the problem of distinguishing pseudoprogession form tumor recurrence, the approach of neural networks was not the first choice. Besides, neural networks are computationally costly and without any experience it is difficult to find appropriate network structures.

3 Materials and Methods

3.1 Data Acquisition

The initial data set of this retrospective study consisted of the data from 55 follow-up studies from patients who were diagnosed with glioblastoma multiforme (GBM). All patients had undergone surgical resection followed by radiation therapy combined with chemotherapy with temozolomide according to the Stupp protocol [15] and a subsquent adjuvant chemotherapy with temozolomide [16]. Response to the therapy was controlled by MRI, including perfusion protocols, in intervals of 3 months after the surgery. The MRI data of these follow-up examinations were used to predict tumor recurrence or pseudoprogression. The true outcome was determined by further follow-up MRI scans.
The data came from 4 different MRI Scanners and each of them was using different protocols. To avoid biases in the data due to different hardware and protocols only data of the scanner providing the largest data set was used in the detailed analysis. Thus the number of data sets reduced to 20. For 4 patients data sets of two different follow-up studies, from two different dates were present. Only the data from the latest study was used. Hence 16 data sets were used for the further study. 8 patients were diagnosed with tumor recurrence and 8 patients showed pseudoprogression. The used MRI data was acquired with a 3-Tesla MRI scanner (MAGNETOM TimTrio, Siemens Healthcare, Erlangen, Germany) and the following protocols were applied:

- Contrast-enhanced T1 weighted imaging (T1WI):
  repetition time varied from 505 to 917 ms, echo time = 20 ms, flip angle = 90°, voxel size = $0.86 \times 0.86 \times 4$ mm^3, spacing between slices = 4.8 mm, Field of view (FOV) varied from 175 to 220×220 mm^2.

- T2 weighted imaging (T2WI):
  repetition time = 5000 ms, echo time = 102 ms, flip angle = 140°, voxel size = $0.69 \times 0.69 \times 4$ mm^3, spacing between slices = 4.8 mm, FOV = 220×220 mm^2.

- Fluid Attenuated Inversion Recovery (FLAIR):
  repetition time = 9240 ms, echo time = 106 ms, inversion time = 2500 ms, flip angle = 150°, voxel size = $0.86 \times 0.86 \times 4$ mm^3, spacing between slices = 4.8 mm, FOV = 199×220 mm^2.

- Dynamic susceptibility contrast-enhanced (DSC) perfusion weighted imaging: repetition time = 1250 ms, echo time = 28 ms, flip angle = 60°, voxel size = $1.8 \times 1.8 \times 5.4$ mm^3, spacing between slices = 6 mm, FOV = 230×230 mm^2.
  Based on perfusion imaging the following quantitative parameter maps were created:
  - relative cerebral blood volume (CBV)
  - relative cerebral blood flow (CBF)
  - Mean transit time (MTT)
  - Time to peak (TTP)
- Diffusion weighted imaging (DWI):
  repetition time = 2800 or 2900 ms, echo time = 106 or 109 ms, flip angle = 90°, voxel size = $1.72 \times 1.72 \times 5.4$ mm^3, spacing between slices = 6 mm, FOV = 220×220 mm^2.
  Apparent diffusion coefficient (ADC) map was created.

The calculation of CBV, CBF, MTT, TTP and ADC maps was performed by the built-in Software by Siemens.

3.2 Image Registration and Resampling

The 8 input images were registered and resampled using the SMP8 software package (Statistical Parameter Mapping, [46]) for MATLAB® (R2010a, The MathWorks, Inc., Natick, Massachusetts, United States). This was a crucial task because in the further analysis each volume element of the tissue was characterized by an 8-dimensional feature vector. The feature vector consisted of the intensities of the volume element in the 8 different images. Therefore it was necessary that the voxels at the same position in each image referred to the same volume element in the tissue. Figure 3.2 illustrates how the feature vector of one volume element was obtained.

All images of one patient were registered to the contrast enhanced T1-weighted image. Since the perfusion maps and the ADC-maps did not contain very much anatomical information, those maps were not directly registered to the T1-weighted image. Instead, the first image of the time series that was performed to acquire the perfusion data was registered to the T1-weighted image. The perfusion maps (CBV, CBF, MTT, TTP) were forced to remain in alignment. That means the transformation matrix that resulted from the transformation of the first time point of the perfusion acquisition series to the T1-weighted image was also applied to the perfusion maps. In the same way the first image of the diffusion acquisition series was registered to the T1-weighted image and the ADC-map was transformed by the resulting transformation matrix.

The registration algorithm of SPM8 is based on the work by Collignon et al. [47]. Collignon

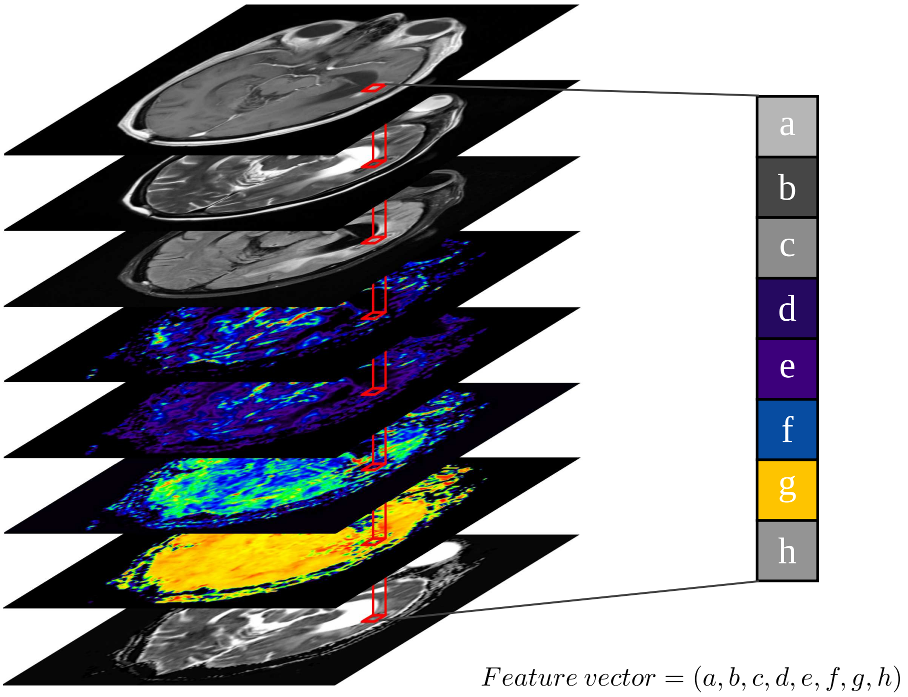


Figure 3.1: This graphic illustrates how the feature vector for one voxel was obtained from the registered images. For each image it shows the slice from the same position. The images were ordered from top to bottom as follows: T1WI, T2WI, FLAIR, CBV, CBF, MTT, TTP, ADC.

et al. proposed a rigid body registration for 3D multi-modality medical image data. They maximize the similarity of all possible pairs of voxel values by maximizing the mutual information of the joint probability distribution. The developers of SPM8 varied the algorithm such that the computations are accelerated and that getting stuck in local minima is better avoided [46].

The images were also resliced and resampled to provide that the slices and the resolution of each image matched. The CBV-map was used as reference image that defined the space to which the other images were resliced and resampled. A perfusion map was chosen as reference because all the perfusion images had already the same resolution and they had the lowest resolution. Thus only minimal interpolations were necessary in the process of reslicing and resampling. Trilinear interpolation was used as interpolation method.

In the end all images were resliced and resampled to a voxel size of $1.8 \times 1.8 \times 5.4$ mm^3 and a spacing between slices of 6 mm.

3.3 Selection of ROI and Normalization

The contrast enhanced lesions were manually segmented in the original contrast enhanced T1-weighted image by an experienced neuro-oncologist. To provide comparability within subjects the image intensities had to be normalized. Therefore a second region of interest

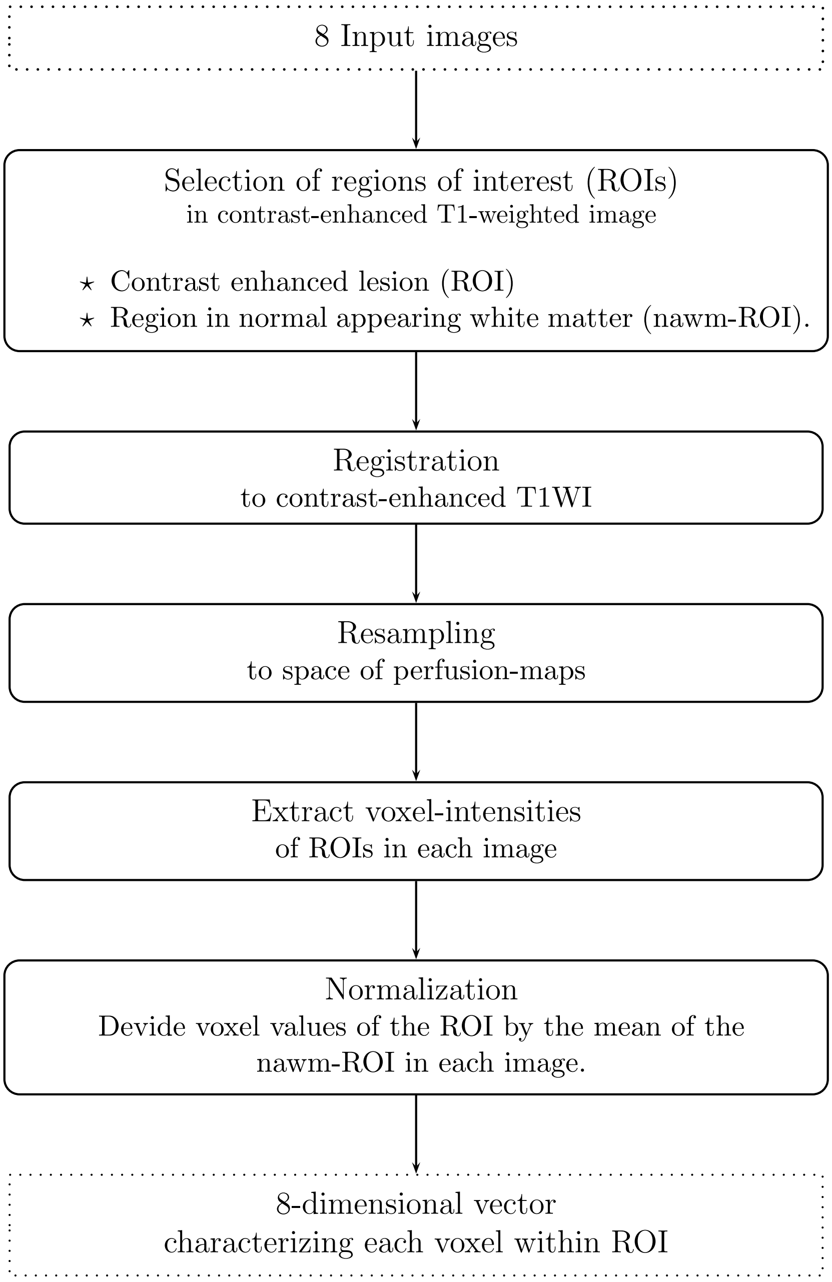

8 Input images
Selection of regions of interest (ROIs)
in contrast-enhanced T1-weighted image
⋆ Contrast enhanced lesion (ROI)
⋆ Region in normal appearing white matter (nawm-ROI).
Registration
to contrast-enhanced T1WI
Resampling
to space of perfusion-maps
Extract voxel-intensities
of ROIs in each image
Normalization
Devide voxel values of the ROI by the mean of the
nawm-ROI in each image.
8-dimensional vector
characterizing each voxel within ROI

Figure 3.2: Flow-chart showing the steps in image processing for feature extraction

(ROI) in the normal appearing white matter (nawm) was selected. Those 2 ROIs were preserved as two separate binary masks. Since the T1-weighted images were resliced and resampled the files containing the ROI-masks had to be resliced and resampled as well as described in section 3.2. Subsequently the ROIs could be transferred easily to all the other images.

T1WI, T2WI, FLAIR, CBV, CBF, MTT, TTP and ADC were normalized by dividing the voxel intensities by the mean intensity in the nawm-ROI of the corresponding image. Figure 3.2 summarizes all the previously described steps in image processing for feature extraction.

3.4 Scaling - Normalization of Intensity Range

When using SVMs it is advisable to have attributes, whose values have the same range. Otherwise attributes that have a greater numeric range might dominate those with smaller numeric ranges [45]. Moreover the kernel used in the SVM has the same width in all dimensions. That means that the distribution of all attributes is estimated with the same kernel width. Hence it makes sense that the attributes should have approximately the same maximum and minimum.

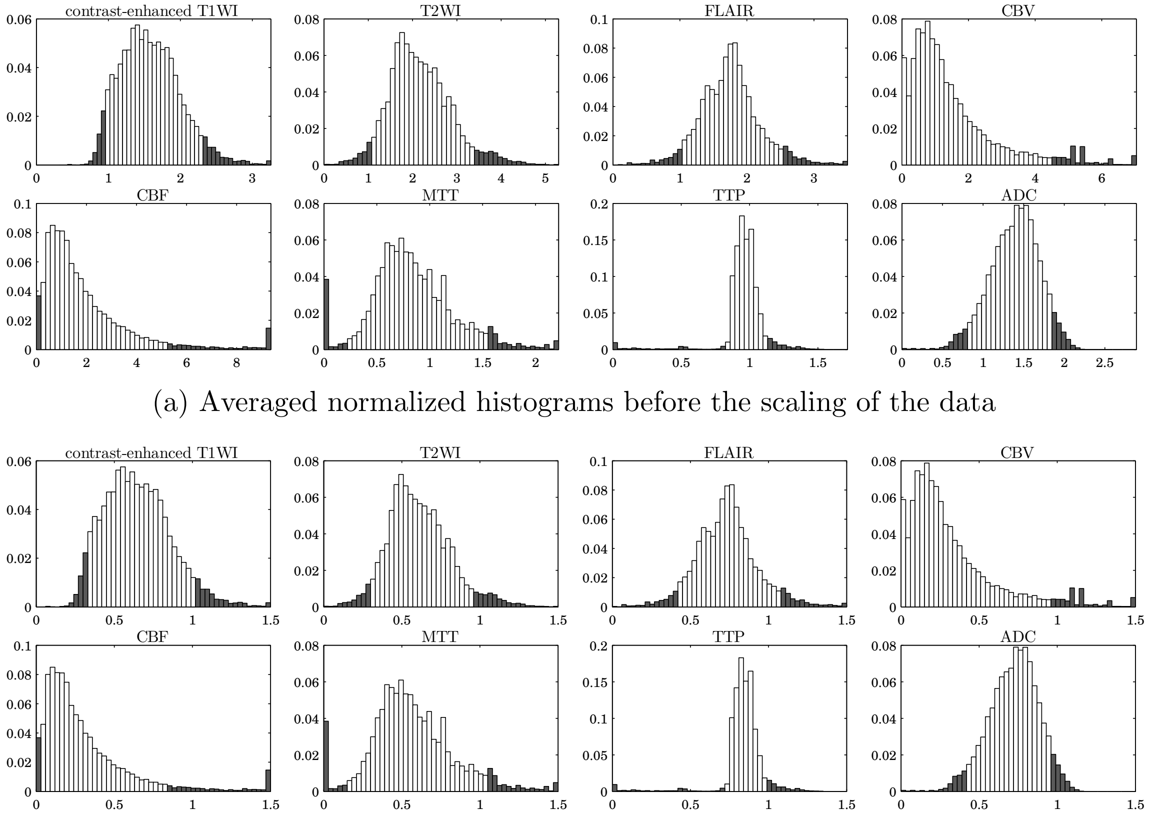
The data is for example often mapped to the range from 0 to 1 by linear scaling [45, 33]. Linear scaling is when the minimal value of the entire data set is subtracted of the data and afterwards the data is divided by the range, that is the difference between the minimal and the maximal value. Of course this has to be done for all attributes separately.

However, the result of this approach is highly influenced by outliers. Therefore the data was not divided by the difference between minimum and maximum, but instead of the maximum the 0.95-quantile of the entire data set for each sequence was used.

To illustrate the effect of scaling Figure 3.3 shows averaged normalized histograms for each sequence before and after the scaling of the data. To get those histograms the histograms for each patient and sequence were calculated and normalized by the number of voxels within the ROI of the corresponding patient. Those normalized histograms were averaged over all patients. Figure 3.3 shows those histograms for each sequence before and after the scaling. The shapes of the histograms have not changed, but it becomes apparent that after the scaling the values lie approximately in the same range, independent from the sequences.

3.5 Training and Classification

Analogously to Hu et al. [1] a one-class support vector machine (OC-SVM) was chosen as classifier. The implementation for the OC-SVM was adopted from the LIBSVM package



(a) Averaged normalized histograms before the scaling of the data

(b) Averaged normalized histograms after the scaling of the data

Figure 3.3: The histograms of the data of each patient and sequence were normalized by the number of voxels within the ROI of the corresponding patient. Those normalized histograms were averaged over all patients resulting in the histograms shown above. The dark bars denote the values below the 0.05-quantile and above the 0.95-quantile respectively. (a) shows the histograms before the scaling of the data. In each sequence the data between the 0.05-quantile and the 0.95-quantile lie in ranges of sizes of about 0.25 to 7 depending on the sequence. (b) shows the histograms after the scaling. The shapes of the histograms have not changed and the majority of the data in each sequence lies in ranges of less varying sizes (0.25 to 1).

[48] for MATLAB$^{®}$. As kernel function the radius basis function (RBF) ($k(x_n, x) = \exp(-\gamma \left\| x_n - x \right\|^2)$) was selected.

Each voxel within the ROI was considered as an spatially independent sample. However it was important to give each patient the same weight in the training processes. The data from patients with large ROIs should not dominate the training set. To meet this requirement from each patient with confirmed pseudoprogression the same number of voxels was randomly sampled. The number of sampled voxels in one patient was the number of voxels of the smallest ROI. The combination of these sampled data points resulted in the training set on which the OC-SVM was trained.

Hu et al. [1] hypothesized that the trained OC-SVM classifies significantly more voxels as positives in subjects showing pseudoprogression than in subjects suffering from recurrent tumor. Hence, by evaluating the percentage of positive voxels within the ROI the patients can be classified into pseudoprogression or progression.

3.6 Validation

To evaluate the classifier several performance measures were computed. To increase the generalizability cross-validation was performed. Using receiver operating characteristic (ROC) an optimal threshold for the fraction of positively classified voxels was determined.

3.6.1 Leave-One-Out Cross-Validation

Due to the small number of patients the performance of the classifier was evaluated by leave-one-out cross-validation. The SVM was trained using data of all patients with confirmed pseudoprogression but one. This was repeated until each patient was left out from the training once, which resulted in 8 different models. Afterwards those models were tested with the data from the patients which were left out at the training and with data from randomly sampled patients showing progression. As a result each performance measure was computed 8-times. In the end those measures where averaged:

$$averaged\ measure = \frac{\sum_{i=k}^{K} measure_i}{K}, \tag{3.1}$$

where K is the number of reruns for the cross-validation. The averaged performance measures were the basis for the evaluation and optimization of the classifier.

3.6.2 Performance Measures

The classification by a two-class classifier can result in four cases. The first one is when a sample that truly belongs to the "positive" class is classified as being a positive. That would be a true positive (TP). The second case is when a sample believed by the classifier to be a positive, but in truth belongs to the "negative" class. Such a case is called a false positive (FP). Accordingly the case of a negative sample that is rightly classified as such is called true negative (TN). A sample that is wrongly predicted to be a negative but in truth is a positive is a false negative (FN). From the numbers of TP, FP, TN and FN the confusion matrix is built (table 3.1) and several performance measures for the classifier can be computed.

		actual class		
		positive	negative	
predicted class	positive	TP	FP	TP+FP
	negative	FN	TN	FN+TN
		TP+FN	FP+TN	TP+FP+FN+TN

Table 3.1: The confusion matrix, also sometimes referred to as truth table, gives an overview over the performance of a classifier. From its elements several performance measures like sensitivity and specificity can be computed.

Sensitivity and Specificity

Sensitivity (SN) corresponds to the probability that a patient who in truth should be a positive actually is classified as positive. It is the ability of the classifier to recognize positive results.

$$Sensitivity = \frac{TP}{TP + FN} = True\text{-}Positive\text{-}Rate \tag{3.2}$$

Specificity (SP) corresponds to the probability that a patient who should be classified as negative is truly identified as a negative by the classifier. It is the ability to recognize negative results.

$$Specificity = \frac{TN}{TN + FP} = 1 - False\text{-}Positive\text{-}Rate \tag{3.3}$$

Both sensitivity and specificity should be large. Assuming that a classifier classified all samples as positive, then clearly all truly positive examples would be recognized correctly. The sensitivity would amount to the optimal value of 1. However, the specificity would be as bad as it can get, namely 0, because none of the in truth negative samples would be identified as such. In short, a high sensitivity leads in most cases to a reduced specificity and vice versa. It is essential to find an acceptable trade-off between those two measures depending on the application of the classifier. The product of sensitivity and specificity can be calculated as an overall measure for the performance of the classifier [33]:

$$Sensitivity \cdot Specificity = SN \cdot SP = \frac{TP \cdot TN}{(TP + FN) \cdot (TN + FP)} \tag{3.4}$$

ROC-Curve

The receiver operating characteristic (ROC) curve plots the sensitivity against the false-positive-rate (1 - specificity). It illustrates the tradeoff between sensitivity and specificity

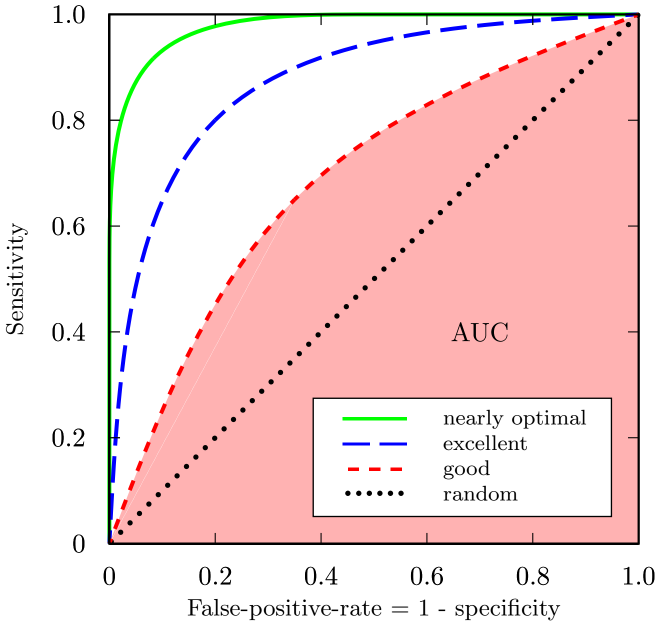


Figure 3.4: 4 examples for ROC-Curves. The dotted line is the result of a completely random classifier. This is the worst classifier possible, because any classifier that is below the diagonal can be improved by inverting its decision. The more the curves converge the upper left corner the better. The AUC for the red dashed line is indicated by the red shaded area.

at different values of one parameter of the classifier, such as a threshold that separates the data into two classes. The more the ROC-curve converges the upper left corner the better. The area under the curve (AUC) is a measure for the quality of a classifier. The bigger the area the better. The AUC can be interpreted as the probability that a positive instance is ranked over another negative one [33]. A completely random classifier has an AUC of 0.5. Any other classifier can only do equally well or better. The AUC can only be ≥ 0.5 because any classifier whose ROC-curve is below the diagonal can be improved by inverting its decision [35]. Figure 3.4 shows some examples for ROC-curves.

3.6.3 Parameter Optimization

To train the OC-SVM two parameters had to be selected: γ and ν. In the LIBSVM package γ controls the width of the kernel function, providing that the radial basis function is used as kernel function. ν controls the fraction of the training data that is allowed to be misclassified by the OC-SVM.

In the LIBSVM package the radial basis function is defined as $f(x_n, x) = \exp(-\gamma \cdot \|x_n - x\|^2)$. In comparison to the the Gaussian function $f(x_n, x) = a \cdot \exp(-\|x_n - x\|^2 / 2\sigma^2)$ it becomes clear that γ is inverse propotional to the squared standard deviation of the kernel: $\gamma = \frac{1}{2\sigma^2}$. Therefore a larger γ leads to a smaller kernel width. Because σ provides a better impression of the size of the kernel,

instead of tuning γ directly, σ of the expression $\gamma = \frac{1}{2\sigma^2}$ was used as parameter in the further optimization process.
To find the optimal value the OC-SVM was trained with the following values for σ: 0.05, 0.1, 0.15, 0.2, 0.225, 0.25, 0.275, 0.3, 0.325, 0.35, 0.375, 0.4, 0.425, 0.45, 0.5, 0.75, 1, 1.5 and 2.
Additionally the parameter ν was varied. The classifier was trained with the following values for ν: 0.1, 0.3, 0.5, 0.65, 0.8, 0.9, 0.95, 0.99.
However, in OC-SVMs a larger ν also results in a smaller region that is estimated to contain the training data. Therefore the region defining hyperplane was shifted towards the origin in feature space, which is equivalent to making the region bigger by a "safty margin". The probability that a new point falls into the region and is classified increases to be positive. The distance of the hyperplane to the origin is given by $\frac{\rho}{\|\mathbf{w}\|}$. By reducing ρ by a factor $T > 0$ the distance between the region defining hyperplane to the origin is reduced to $\frac{\rho - T}{\|\mathbf{w}\|}$. T was determined with respect to the SVs that were considered as outliers. More precisely, T was chosen such that a defined fraction c of the SVs, that originally fell on the wrong side of the hyperplane, was classified correctly after the modification of the margin by T. For example, $c = 0.3$ resulted in a more "liberal" classifier. 30% of the SVs that at first represented training errors were classifed correctly with $c = 0.3$. c was varied from 0 to 1. $c = 1$ means, that the region was modified such that all outliers fell into it. $c = 0$ means that the region remains as it was estimated by the OC-SVM.
For each training run different training sets were randomly sampled, which led to different results in each training run. Therefore the training process with a specific parameter set was repeated 40 times. Each time the training set was randomly resampled. The performance measures obtained by leave-one-out cross-validation (each time one patients was left out) were averaged over the 40 runs. The parameters that resulted in classifiers that on average classified the single voxels with the largest product of sensitivity and specificity ($SN \cdot SP$) were considered to be optimal.

4 Results

4.1 Parameter Selection

The largest product of sensitivity and specificity ($SN \cdot SP$), averaged in cross validation and averaged over 40 training runs, that could be achieved in the parameter optimization process was 0.4000. It was obtained with the following parameters: $\sigma = 0.425$ (equivalent to $\gamma = 2.77$), $\nu = 0.90$ and $c = 0.65$.

Furthermore, the parameter combinations of σ, ν and c presented in table 4.1 all yielded values for $SN \cdot SP$ of larger than 0.3990. This suggests that parameters of $\sigma = 0.30$ to 0.45, $\nu = 0.9$ to 0.99 and $c = 0.65$ to 0.70 are a robust choice to obtain a good classifier.

Table 4.1: Parameter combinations of σ, ν and c, that all showed almost the same good result of $SN \cdot SP > 0.3990$.

σ	0.300	0.300	0.350	0.400	0.400	0.425	0.425	0.450
ν	0.900	0.990	0.950	0.900	0.950	0.900	0.990	0.900
c	0.650	0.700	0.650	0.650	0.650	0.650	0.700	0.650

In the following the results of a classifier trained with one specific training data set are presented as an example. The OC-SVM was trained using the parameters considered as optimal: $\sigma = 0.425$ (equivalent to $\gamma = 2.77$), $\nu = 0.90$ and $c = 0.65$. Again, due to random sampling of the training data the classifier would yield different results even when the same paramerters are used. The results presented here are obtained from a averagely well performing classifier with $SN \cdot SP = 0.4001$, obtained by cross-validation, for the voxelwise classification.

Figure 4.1 shows the ROC curve obtained by the voxel-for-voxel classification by the example classifier mentioned above. More precisely, in each training run of the leave-one-out cross-validation (each run one patient was left out) sensitivity and specificity, with respect to the correctly classified single voxels, were calculated for various values of c. For each c the mean of sensitivity and the mean of specificity over all folds in the cross-validation process was built. Plotted as a ROC-curve this yielded an area under the curve (AUC) of 0.66639. The largest $SN \cdot SP$ was, as expected, obtained with a value $c = 0.65$.

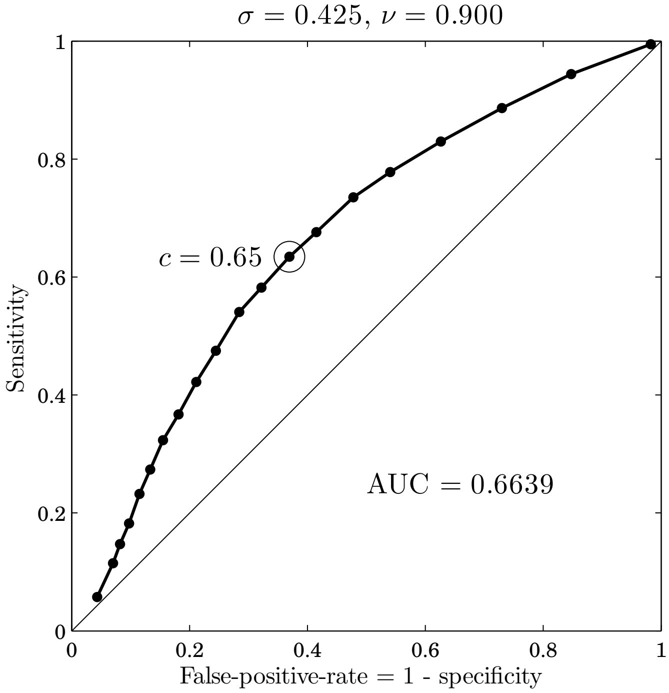


Figure 4.1: Averaged ROC-curve of the voxelwise classification obtained by a averagely well performing training run. c controlled the trade-off between sensitivity and specificity. $c = 0.65$ resulted in the largest product of sensitivity and specificity ($SN \cdot SP = 0.4001$).

4.2 Classification Performance

Table 4.2 lists the results from cross-validation for each patient. It also lists the fraction of voxels within the ROI of a patients that are thought by the classifier to represent pseudoprogression (positive voxels).

In the group of patients showing pseudoprogression the median percentage of voxels classified as positive was 61.01%. In the group of patients suffering from recurrent tumor the median was 40.34%. A Wilcoxon rank-sum test was performed, yielding that the two groups can be distinguished with a p-value of 0.0104.

Figure 4.2 shows the ROC-curve that was obtain by classifying the patients according to the percentage of voxels that were classifed as positives within their ROI. The ROC-curve yielded an AUC of 0.8750. A threshold for the percentage of positive voxels between 53.96% and 51.07% resulted in the largest product of sensitvity and specificity ($SN \cdot SP = 0.6563$) concerning the classification of the single patients. Table 4.3 shows the confusion matrix for a classifier with threshold of $(53.96\% + 51.07\%)/2 = 52.52\%$. The correspondig sensitivity and specificity amounted to $SN = 0.75$ and $SP = 0.875$ respectively.

Figure 4.3 shows two examples of patients, one with recurrent tumor and one with pseudoprogression. It shows the T1-weighted image of the two patients and how the voxels within the ROI were classified.

Table 4.2: Results from leave-one-out cross-validation. Each time one patient showing pseudoprogression was left out in the training process. The resulting classifier was tested with the data of the left out patient. It was also tested with data from one randomly selected patient showing recurrent tumor. The results of the testing is shown in this table. The group denoted as positive refers to patients showing pseudoprogression. The negative group refers to patients suffering from recurrent tumor.

Since there cannot be true positive and false negative voxels within the ROI of a patient showing recurrent tumor, no sensitivity is specified. The same is true for specificity in patients showing pseudoprogression. There cannot be true negative and false positive voxels. Positive voxels are the voxels that are believed by the OC-SVM to represent pseudoprogression.

	actual group	sensitivity	specificity	fraction of positive voxels within ROI
Patient 1	negative	-	0.5926	40.74%
Patient 2	negative	-	0.5345	46.55%
Patient 3	negative	-	0.4202	57.98%
Patient 4	negative	-	0.7651	23.49%
Patient 5	negative	-	0.4893	51.07%
Patient 6	negative	-	0.6921	30.79%
Patient 7	negative	-	0.9509	4.91%
Patient 8	negative	-	0.6005	39.95%
Patient 9	positive	0.4205	-	42.05%
Patient 10	positive	0.9518	-	95.18%
Patient 11	positive	0.4369	-	43.69%
Patient 12	positive	0.7723	-	77.23%
Patient 13	positive	0.7340	-	73.40%
Patient 14	positive	0.5396	-	53.96%
Patient 15	positive	0.5558	-	55.58%
Patient 16	positive	0.6644	-	66.44%
Mean		0.6344	0.6306	

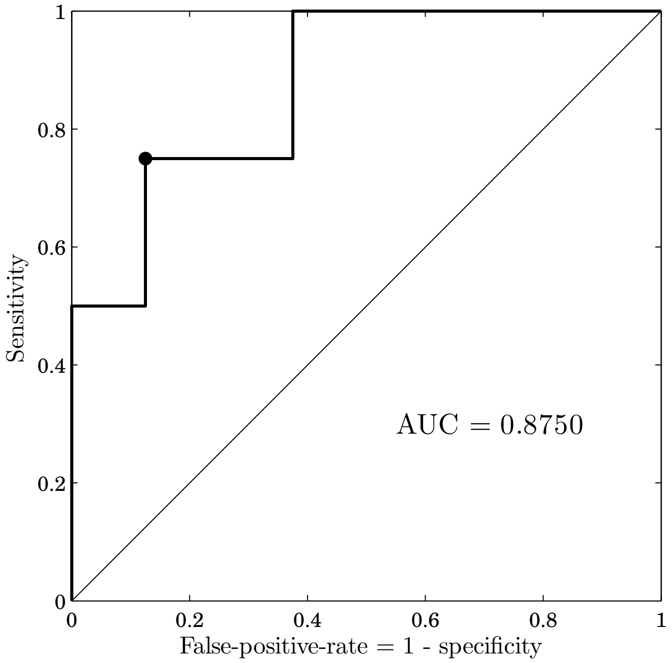
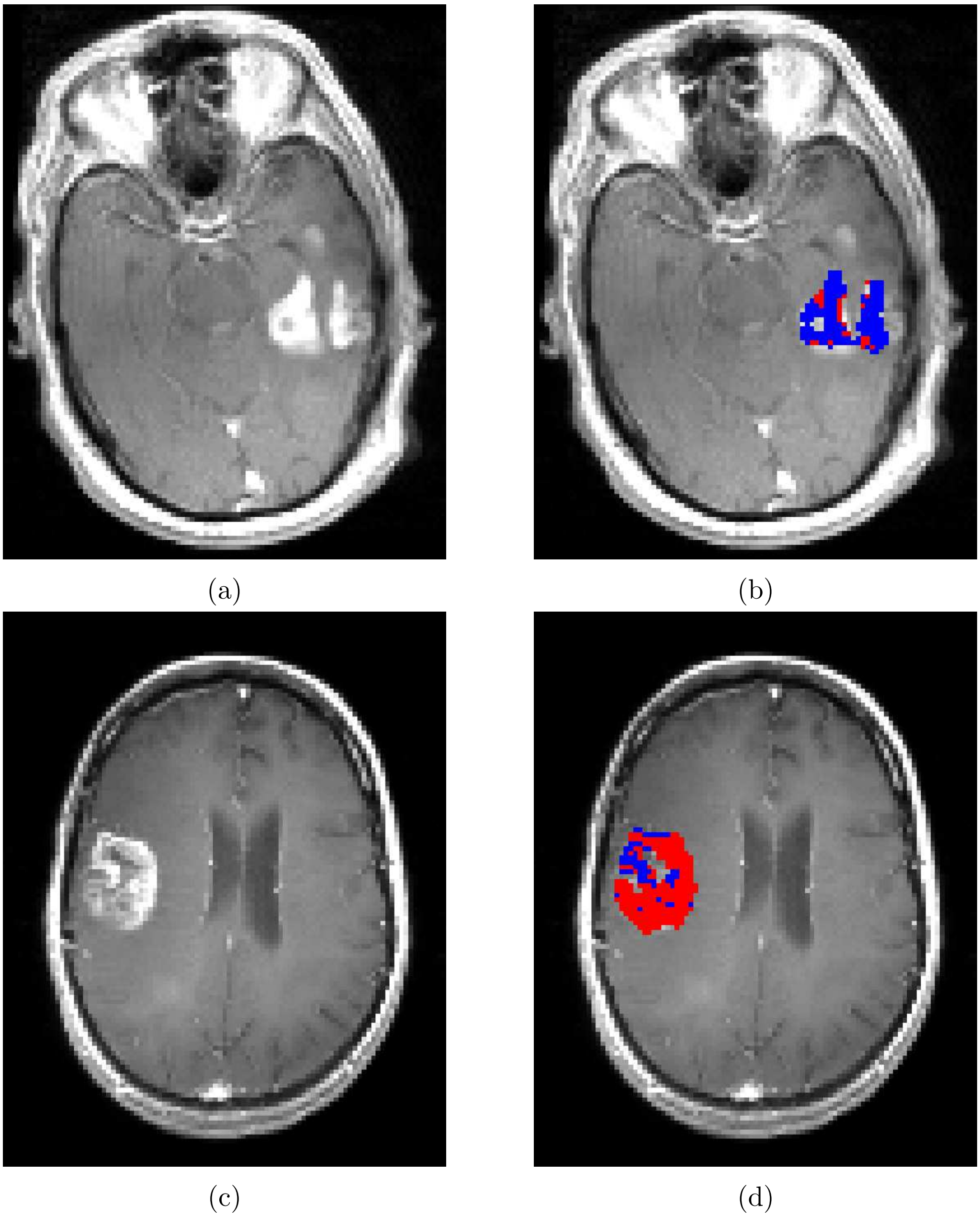


Figure 4.2: The ROC-curve was obtained by classifying the single patients according to the percentage of voxels that were classified as positves within the ROI. The black dot indicates the point were the patients were classified with the maximal product of sensitivity and specificity ($SN \cdot SP = 0.6563$). It corresponds to a threshold between 53.96% and 51.07%.

Table 4.3: The confusion matrix resulting from classifying the patients according to the fraction of voxels that were classified as positives within the ROI. All patients with a percentage of positive voxels larger than 52.52% were predicted to show pseudoprogression. All patients with a percentage of positive voxels lower than 52.52% were predicted to show recurrent tumor.

		actual class		
		pseudoprogression	recurrent tumor	
predicted class	pseudoprogression	6	1	7
	recurrent tumor	2	7	9
		8	8	16



(a)

(b)

(c)

(d)

Figure 4.3: The color of the voxels within the ROI indicates the class to which they were thought to belong. Red regions indicate voxels that are classified as positives by the OC-SVM. Blue regions indicate voxels that are classified as negatives. (a) shows the contrast-enhanced T1-weighted image of a patient with tumor recurrence. (b) shows the same image as in (a) but with the colored ROI. The majority of the voxels within the ROI are classified to represent the negative class (blue). (c) shows the contrast-enhanced T1-weighted image of a patient that is thought to show pseudoprogression. (d) shows the same image as in (c) but with the colored ROI. The majority of the voxels are classified to represent the positive class (red).

5 Discussion

5.1 Parameter Selection

There is no general guideline or theory about how to find the best parameter set σ (or γ), ν and c. The optimal parameters depend on the training data.
Hsu et al. [45] recommend performing a grid search using cross-validation to find the best parameters. Grid search simply means that various combinations of the parameters are tried and the one with the best performance is picked. Although this approach is time consuming it is most reliable and in contrast to some other more sophisticated, but iterative methods it can be parallelized.

5.1.1 Regularization Parameter ν

In general larger ν lead to smoother decision boundaries, because more outliers are allowed and more instances of the training data are forced to be used as support vectors (SVs), which define the boundary. That is why classifiers trained with large ν will probably perform better, when the optimal decision boundary can be expected to be not convoluted very much and when only a small training set is available. However large ν also lead to longer training time [37] and due to the larger number of SVs caused by the larger ν the evaluation of the classifier on new data also takes longer. Therefore large ν are not favorable. When the data of one class is very compact and well separable from other classes, only a small subset of the training data has to be used as SVs and few outliers have to be allowed to estimate a well fitting region that contains most of the class members. A low ν will perform well and the model will be more compact, which leads to short training and evaluation time. However, when the class is highly overlapping with other classes a large ν will most likely perfom best.
The tests with cross-validation revealed that values of $\nu = 0.90$ to $\nu = 0.99$ performed best on the available data. $\nu = 1$ was not allowed by the training function of the LIBSVM package. However it can be assumed that an OC-SVM using $\nu = 1$, would perform well, too, providing that the originally estimated region is made bigger by the margin parameter T. An OC-SVM that was trained with $\nu = 1$ is equivalent to a thresholded Parzen-windows estimate of the underlying density [37]. Therefore using a Parzen window

estimator instead of an OC-SVM is worth to be considered.
Similar to the K-nearest-neighbor classifier, the Parzen window estimator takes no time to be trained. The training just involves storing the entire training set. However the evaluation of the decision value can be time consuming when the training set is very large. On the contrary, the OC-SVM only stores a subset of training data, the support vectors. The support vectors are the instances that are sufficient to define the optimal decision boundary. Hence, when the training data set is very large it makes sense to use a OC-SVM, because in comparison to the Parzen window estimator the model is more compact and the evaluation of new data is faster.
When the best performing ν approximates 1, it means that almost all training instances have to be used as SVs to obtain a good decision boundary. When so many training instances are necessary to obtain good results, this suggests that training with more data, i.e. with a larger training data set, would perform considerably better.

5.1.2 Kernel Parameter σ

To get a first estimate of the expected optimal value for σ the histograms in figure 3.3b were considered. The majorities of the values (values between the 0.05-quantile and 0.95-quantile) of the single features lie in ranges with sizes of about 0.25 to 1. Thus, kernels with standard deviations of at least $\sigma = 0.1$ to $\sigma = 0.5$ could be expected to estimate the distribution of the data well.
Figure 5.1 shows the product of sensitivity and specificity ($SN \cdot SP$) with respect to

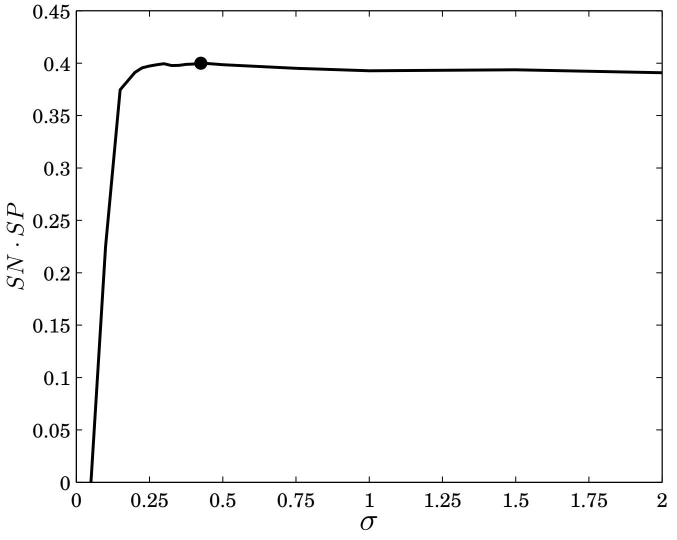


Figure 5.1: The product of sensitivity and specificity ($SN \cdot SP$), obtained by cross-validation, with respect to various values of σ, averaged over 40 training runs, using the parameters $\nu = 0.90$ and $c = 0.65$. The black dot indicates the maximal $SN \cdot SP$ at $\sigma = 0.425$.

.

various values of σ, when ν and c are fixed to $\nu = 0.90$ and $c = 0.65$. It shows that very small kernels perform very badly. With kernels of standard deviations lower than 0.15 the classifier is obviously likely to be overfitted. That means, although it classifies the training data well, it is not able to classify new test data correctly. Hence, a good estimation of the distribution of the data is not possible with such small kernels. At least not when the training data set is small.

As shown in figure 5.1 the largest $SN \cdot SP$ is achieved with the kernel width of $\sigma = 0.425$. Larger σ do not perform significantly worse, but it seems that the more σ is increased, the more the performance is reduced. These results suggest, that $\sigma = 0.0425$ is a robust choice and that slightly larger or slightly lower σ would not perform significantly worse.

5.1.3 Margin Parameters c and T

c controls the size of the region that is estimated to contain the class on which the OC-SVM was trained. This is done by reducing the distance of the region defining hyperplane to the origin in feature space by the margin T (Originally Schölkopf et al. [37] named the parameter γ. To avoid confusion with the kernel parameter here it is referred to as T.). c controls the size of T in feature space. Increasing c make the region bigger, such that more SVs that were originally considered as outliers fall in the region and do not represent training errors any more. c defines the fraction of outliers that fall into the class defining region after the modification of the distance of the hyperplane to the origin in feature space by T.

Figure 5.2 plots for a fixed $\sigma = 0.425$ and various ν, which value of c yielded in the maximal product of sensitivity and specificity. It is apparent that larger ν require larger c. This supports the theory that larger ν result in smaller regions, that are estimated to contain the training data, because more outliers are allowed. In order that the classifiers shows good results, the region has to be made bigger by a larger c. The dependence of the optimal value for c on ν suggests that a better definition of T should be developed. A definition that somehow takes the value of ν into account would be probably better.

However, there is very little literature on the role of the margin parameter T and there is no standard definition how to find the optimal T.

The need that c and accordingly T has to be wisely selected in order that the class can be separated from other classes, raises the question why not to use a two-class SVM. Since in this case there is a defined second class a two class SVM would be the intuitive choice for a classifier. Using a two-class SVM no such parameter as c or T would have to be chosen. This question will be discussed later in section 5.6.

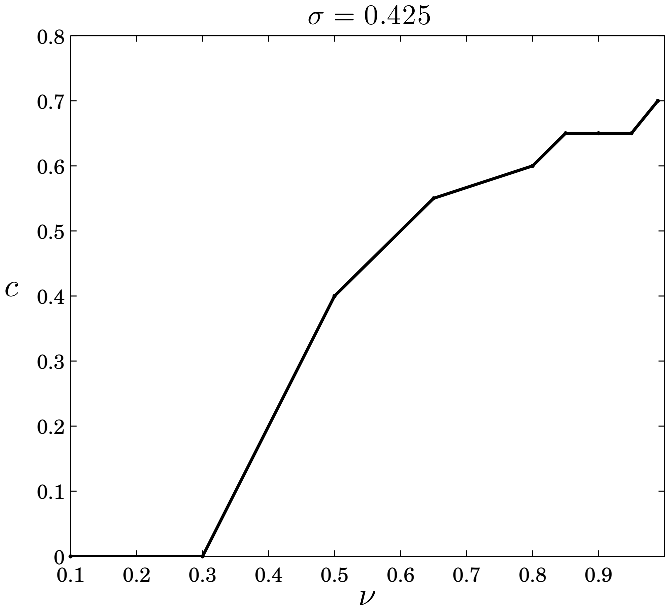


Figure 5.2: The parameter c that yielded in the best classifier (largest $SN \cdot SP$) for various ν. The kernel width was fixed to $\sigma = 0.425$.

.

5.1.4 Limitation

A limitation of this study was that only a small data set was available. The classifier could not be tested independently from the optimization of parameters of the OC-SVM (σ or γ, ν and c). Ideally the SVM trained with the parameter set, which was considered to be optimal, would be tested with an independent test set. That way it would be ensured that the parameters not only are optimal on the data set that was used for optimization but also perform well on new data. However, the optimal parameters were determined using cross-validation which still provides that they represent a robust choice and that they probably will perform well on completely new data, too.

5.2 The Importance of the Different Features

To evaluate which of the 8 features had the most influence on the result, the SVM was also trained using only one feature and using all features but one. Table 5.1 lists the p-values, that indicate how well the two groups of patients could be distinguished by the percentage of voxels that were classified as positives by a classifier that was trained with only one feature or all but one feature. It shows that when the OC-SVM is trained only with data from the CBV-map the two groups can be distinguished with the same p-value (0.0104) as when all features were used. This suggests that the CBV data is most important for the classification. However, when all features but CBV are used, the performance is not heavily reduced. It seems that the other features can compensate the absence of the CBV data. This is not surprising since the perfusion maps correlate, because CBF is defined as [25]

$$CBF = \frac{CBV}{MTT}. \tag{5.1}$$

Furthermore, a classifier that is trained only with the two features CBF and MTT performs just as well (p-value = 0.0104) as the classifier trained only with CBV, although the single features CBF and MTT do not perform well at all.

The FLAIR data seems also to contribute to the decision. When only FLAIR data is used the p-value is relatively low, although not significant. Additionally, when FLAIR data is excluded the performance is slightly reduced. These findings are to some extent consistent with the results of Hu et al. [1]. They also identified the perfusion data as most important for the classification. Additionally they found that there was a difference in FLAIR data between the two groups, but it was not significant. However, they stated that the ADC data showed significant difference between the two groups, but in this study ADC did not seem to contribute much to the decision. However the protocols of diffusion imaging possibly could be improved in order to improve the classification by the ADC values.

Table 5.1: The p-values indicate how well the two groups can be distinguished. The p-values are listed for classifiers that were trained only with the one feature listed in the same row and for classifiers that were trained with all features except for the one of the same row. The p-values were obtained by a Wilcoxon rank sum test applied to the percentages of as positive classified voxels within the ROI of each patients.

Feature	p-value	
	only one feature	all but one feature
contrast-enhanced T1WI	0.3823	0.0104
T2WI	0.7984	0.0104
FLAIR	0.1049	0.0148
CBV	0.0104	0.0148
CBF	0.5053	0.0207
MTT	0.5053	0.0104
TTP	0.6453	0.0104
ADC	0.5054	0.0104

5.3 Suggestions for Improvements of MRI Protocols

Since this was a retrospective study it was not possible to actively influence the settings of the MRI protocols that were used to acquire the data. In future the method should be evaluated in a prospective study, in which the MRI sequences can be properly adjusted in order to increase the performance. In the following some suggestions for adjustments of the MRI protocols will be given.

5.3.1 Flip Angle

The original data set comprised data from 47 patients from different MR-scanners. For some patients the data from multiple MRI studies was available. However, only the data from the latest examination was considered. To avoid bias in the analysis of the data, data from only one scanner was used in the entire study.

However, the data of the other scanners was also shortly analyzed. A little worse results could be expected, because the data sets of those scanners were considerably smaller. Surprisingly the previously used method did not perform well at all on the data of the other scanners. The results of the classifier were not significantly better than random classification.

Hence the MRI protocols used on the other scanners were analyzed and compared to the protocols of the well performing scanner. Since the perfusion data seemed to be most important for the decision, especially the protocols for the perfusion imaging were compared. The most striking difference was the difference in flip angle in the perfusion weighted imaging protocols. The flip angle of the perfusion protocol from the scanner that yielded good results (Siemens TimTrio) was set to 60°, whereas in the other scanners the flip angle was set to 90°.

However, a flip angle of 90° is is not ideal, when perfusion imaging is applied to areas of disrupted blood-brain barrier. The calculation of CBV can be distorted by the so called leakage error, which leads to an underestimation of the CBV. Reducing the flip angle has shown to reduce the error. Fatterpekar et al. suggest to use a flip angle of 35° when using perfusion imaging for the assessment of the treatment response concerning brain tumors [2]. Hence, by adjusting the flip angle in the protocols of perfusion imaging the performance of the classifier could be possibly considerably improved and the method would probably work on data from the other scanners as well.

5.3.2 Spatial Resolution

Even when slice thickness, spacing between slices and voxel size are in all sequences the same, it cannot be expected that slices and voxels of the resulting images match exactly, due to potential motion of the patient during the acquisition. Therefore the images were registered and spatially transformed (rotated, translated, scaled), such that they matched as well as possible.

However, in this retrospective study the slices of the different sequences did not match by definition, because of differing settings concerning slice thickness, spacing between slices and voxel size. Consequently, after registration, in the reslicing process the intensities partly had to be interpolated between the slices. The use of interpolated intensities is not optimal and should be avoided.

Optimally the spacing between slices would be zero. If that is not possible, then the slice thickness and spacing between slices should at least match in those sequences where zero spacing is not possible, such that the amount of interpolation is reduced as far as possible. Since the ROIs are transferred from the T1-weighted images to the perfusion maps they should ideally not be modified very much by the resampling process. However in this study the ROIs were resliced and resampled like the T1-weighted image and therefore they were partly the result of interpolations. Uncertainties due to the interpolations had to be accepted. On the contrary, if the slices matched right from the beginning, the ROIs would actually mark the region they were intended to mark.

Alternatively the ROIs could be marked not before the images were registered and resliced, but afterwards. However, due to organizational reasons the ROIs were drawn in the original (not resampled) contrast enhanced T1-weighted image.

5.4 Sampling of Training and Test Data Set

In order to build a good model, as much from the available data as possible should be used for training. However there were some restrictions for the selection of the training data.

Care was taken that the data was sampled in a way such that the classifier was not corrupted by any bias within the data. Such a bias could be characteristics of the hardware and different MRI protocols. Therefore only data acquired with the same protocols from one scanner was used. Voxels within the ROI of one patient are likely to show similar characteristics. To avoid a bias in the training data set the exact same number of voxels was randomly sampled from each patient that was included in the training of the OC-SVM. That way it was ensured that the SVM was not misled to take characteristics of patients with bigger ROIs more into account than the ones from patients with smaller ROIs.

The data set had to be divided in a training and a test set. The test set should come from different patients than the training set. Simply dividing the whole data set would result in a even smaller number of patients available for training. Additionally the test results would be less representative when the classifier was tested with only a fraction of the data set. Therefore cross-validation was performed to evaluate the performance of the classifier.

K-fold cross-validation divides the data set in K groups. The classifier is trained with data from $K-1$ groups and evaluated on the remaining group. This is repeated K-times. Each time a different group is held out. The performance measures of the K runs are averaged. That way, a larger number of data can be used in each training run and in the end all data was used for testing. When K is the total number of samples it is called leave-one-out cross-validation. It this study each group of voxels came from only one patient. Hence concerning the patients leave-one-out cross-validation was performed. Leave-one-out cross-validation is computationally costly. Therefore it is only recommended when only a small data set is available.

5.5 Comparison to Previous Research

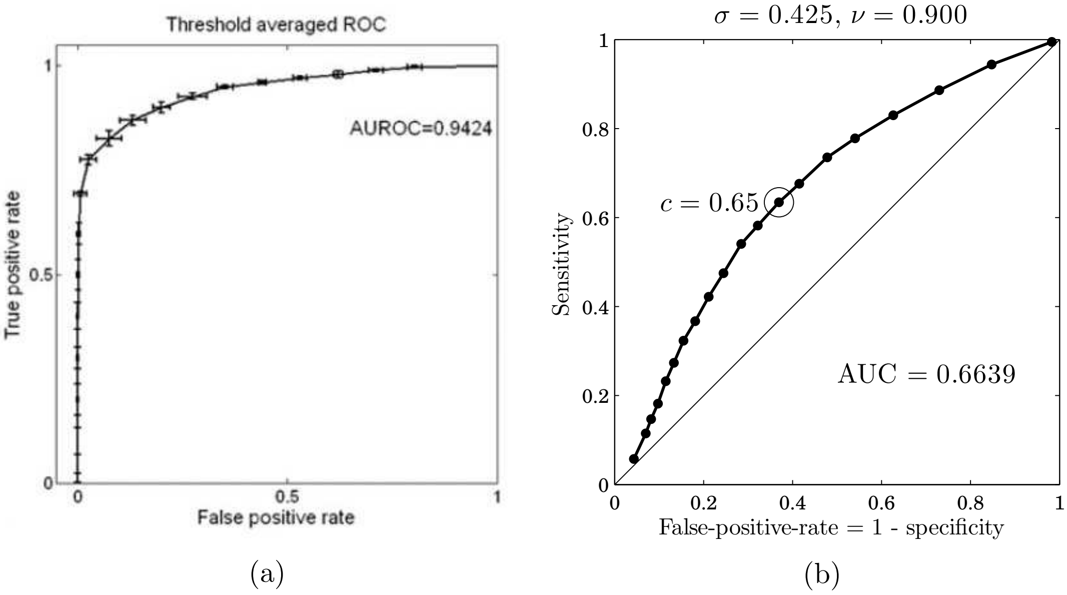
This study is strongly related to the work by Hu et al. [1]. They evaluated if it was possible to differentiate pseudoprogression from tumor recurrence using a one-class SVM applied to multiparametric MRI data. They used a voxel-based approach similar to the method as it was described in chapter 3. The data came from 31 patients. The classifier was trained on data of 8 subjects of the group showing pseudoprogression and the parameters of the SVM were optimized with data from the same 8 subjects and 7 subjects showing tumor recurrence. The data of the remaining 16 patients (8 pseudoprogression, 8 recurrent tumor) was used to test the classifier.

When classifying the single voxels of the test set they achieved a area under the ROC-curve (AUC) of about 0.94. The corresponding ROC-curve was derived for a fixed ν and γ pair by varying a threshold T. However, they did not specify how they defined T. They just said that "SVM requires selection of a single threshold to generate a discrete classifier". It just can be assumed, that they referred to something like the margin parameters ρ or T in the decision functions $f(x) = sgn\left(\sum_n \alpha_n k(x_n, x) - \rho\right)$ or $f(x) = sgn\left(\sum_n \alpha_n k(x_n, x) - (\rho - T)\right)$, respectively.

The ROC-curve in figure 4.1 is somewhat equivalent to the ROC-curve published by Hu et al. mentioned above. Both ROC-curves are shown for comparison in figure 5.3. It shows, that the excellent results of Hu et al. were not reproducable. An AUC of 0.94 could not be reproduced. The AUC obtained in this study was only about 0.66.

Hu et al. [1] identified as optimal parameters for the training of the classifier $\gamma = 5$ and $\nu = 0.06$. They identified those parameters by testing the classifier with a data set obtained from the same patients, that were used for training the classiefer. However, they avoided to use data from voxels that were used for training. Furthermore, for parameter optimization they did not sample voxels from the entire data of the patients showing recurrent tumor, but they filtered the data of those patients first with a "liberal" OC-SVM classifier. That is, they excluded all voxels that showed similarity with the data on which the classifier was trained on.

Under this circumstances it is evident that a low value for the parameter ν will perform best, because in the data set they used for parameters optimization it can be expected that the data of patients showing pseudoprogression is well separable from the "filtered" data of the patients suffering from recurrent tumor. The region containing the training



(a)

(b)

Figure 5.3: (a) shows the ROC-curve obtained by Hu et al. [1] by varying a threshold of the OC-SVM trained with the parameters $\gamma = 5$ and $\nu = 0.06$. (b) shows the equivalent ROC-curve obtained in this study, however different parameters γ and ν were used ($\gamma = 2.77$ or $\sigma = 0.425$, $\nu = 0.9$).

data can be made quite large and still show a low rate of outliers in the training data, i.e. a low value for ν.

The good results of Hu et al. [1] could not be reproduced with the low value of $\nu = 0.06$. On the contrary, the best results were achieved with large ν ($\nu = 0.9$). This discrepancy is explainable by the different way of how the data for the parameter optimization was selected. Hu et al. "filtered" the data of the progression group and they used data from the same patients, whose data was also used for training. Such an approach is prone to overfitting. Overfitting describes the effect that a classifier is overly well fitted to the training data, such that it performs very well on the training data but poorly on new data. It is surprising that Hu et al. still obtained an AUC of 0.94, when testing the classifier with the "unfiltered" test data set, which was completely excluded in the training and optimization process.

In contrast to that, in this study all data from the progression group, without filtering, was used and the performance was tested by cross-validation. That means that no data at all of patients that were involved in the training was used for parameter optimization. The optimal value for the parameter γ was with $\gamma = 2.77$ (equivalent to $\sigma = 0.425$) in the same range like the optimal value determined by Hu et al. ($\gamma = 5$). The difference of the value can be explained by different scaling methods applied to the data.

Hu et al. showed that in pseudoprogression patients 91.2% ± 6.3% ($n = 8$) and in progression patients 31.6% ± 27.4% ($n = 8$) of all voxels within the ROI were classified as positive voxels [1]. They stated, that this is a significant difference with a p-value lower than 0.01. Unfortunately the did not declare the exact p-value and which statistical test they used. To compare the exact p-value a two-sample t-test (also known as Welch-test) was performed, assuming that the values presented by Hu et al. are the means and standard deviations of the percentages. However, it is questionable whether the two-sample t-test is really the appropriate test, because it is not known whether the percentages really were normally distributed. Additionally the standard deviations are extremely different which rather suggests to use a rank-sum test [49]. Since no detailed results were presented by Hu et al. it was not possible to do a rank-sum test. The p-value determined by the two-sample t-test was 0.0002.

In this study the two groups showed a significant difference concerning the percentages of positive voxels with a p-value of 0.0104. Wilcoxon rank-sum test (equivalent to Mann-Whitney U-test) was used as statistical test, because it could not be assumed that the percentages followed a normal distribution, due to the small number of patients [49]. A p-value of 0.0104 suggests that the difference in percentage of positive voxels between the two groups was less significant compared to the results of Hu et al.

Taken together the excellent results of Hu et al. could not be reproduced. The reason could be the fact that different MRI protocols were used. Especially improvements in the protocol of perfusion imaging may lead to better results. Furthermore, a more well-considered choice of patients may also yield in better results. For example data from very small lesions could be excluded.

5.6 Alternative Methods

Commonly SVMs are used as binary classifiers trained on both positive and negative samples. For the problem of assessing the treatment response of glioblastoma multiforme the two classes were defined to be voxels of pseudoprogression and voxels of recurrent tumor, respectively. However, the one-class SVM was chosen as classifier, because it was impossible to identify the single voxels of recurrent tumor, that should represent the second (negative) class, within the contrast-enhanced lesion. Volumes of pseudoprogression were thought to be more homogeneous, which makes the voxels representing pseudoprogression more definable. Therefore the data from patients with recurrent tumor was entirely excluded from the training. Using a one-class SVM it was possible to train only on data from patients with pseudoprogression, which represented the (positive) class.

5.6.1 Two-class SVM

Nevertheless, using the conventional two-class SVM still should be considered. Therefore two-class SVM was also shortly tried to test its use as classifier. After a rough search for the best parameters ν and σ the training and testing of the two-class SVM did not reveal considerably worse results compared to the results presented in chapter 4. Interestingly the best parameters for the two-class SVM were quite the same as used for the training of the one-class SVM previously. The fact that the parameters $\nu = 0.9$ and $\sigma = 0.5$ yielded the best results using the two-class SVM supports the choice of optimal parameters for the one-class SVM.

Using a two-class SVM instead of a OC-SVM would be advantageous considering that only two parameters had to be optimized. It would not be necessary to determine the margin parameter T. A two-class SVM determines automatically the hyperplane in feature space that optimally lies between the two groups. Hence a parameter such as T that shifts the hyperplane is not needed.

5.6.2 Histogram-based Method

Emblem et al. [8] proposed a histogram based approach using an SVM for glioma grading. Using histogram "signatures" as feature vectors would be a promising approach for the problem of differentiating pseudoprogression and tumor progression, too. Especially multidimensional histograms would allow to combine different MRI sequences and parameter maps and also to take the distribution of those combined intensities into account. Similar approaches already were used in image classification [50]. However multidimensional histograms can become very large. When n is the number of bins of the histogram of one parameter alone, then the size of a multidimensional histogram is n^p, where p is the number of parameters that are combined in the multidimensional histogram. Hence, a feature vector representing such a multidimensional histogram would have the length of n^p. When the feature vector is big the curse of dimensionality takes effect. The curse of dimensionality is the phenomenon that, when the input space of the feature vector becomes larger, a larger number of training samples are needed to achieve significance [43, 32]. Since the number of patients is limited using multidimensional histograms as feature vectors will not be feasible.

However, since CBV seems to be most important for distinguishing pseudoprogression from tumor progression, using only the histograms of normalized CBV intensities within a ROI as proposed by Emblem et al. [8] may also work well for treatment assessment of glioblastoma multiforme.

5.7 Conclusion

A drawback of this study was the small number of subjects. With a bigger number of subjects the results could be generalized with a higher significance. Furthermore, shortcomings of the MRI protocols could not be corrected because it was a retrospective study. In a prospective study the MRI protocols could be improved in order to increase the performance of the here presented method.

The selection of the ROIs in the contrast-enhanced T1-weighted image is highly operator-dependent. Therefore the influence of the selection of the ROIs should be investigated. However, the classification of the single patients is not dependent on single voxels, which suggests that variations of the ROIs do not have very heavy effects on the decision. Nevertheless the issue should be of concern and an automatic segmentation should be considered.

Taken together, the here presented approach to assess the treatment response of glioblastoma multiforme shows potential. SVMs have proven to be suitable for the classification of voxels characterized by multiparametric MRI data.

Although the results of this study were not as good as the results presented by Hu et al. [1] they were still convincing. The two groups of patients were well distinguishable, which suggests that the approach may be of value in clinical practice. Improvements of the MRI protocols of perfusion imaging, may lead to a more accurate estimation of CBV, which can be expected to improve the performance of the classifier considerably.

6 References

[1] Hu X, Wong KK, Young GS, Guo L, Wong ST: Support vector machine multiparametric MRI identification of pseudoprogression from tumor recurrence in patients with resected glioblastoma. *Journal of Magnetic Resonance Imaging* 33: 296–305 (2011)

[2] Fatterpekar GM, Galheigo D, narayana A, Johnson C, Knopp E: Treatment-related change versus tumor recurrence in high-grade gliomas: A diagnostic conundrum – use of dynamic susceptibility contrast-enhanced (DSC) perfusion MRI. *American Journal of Roentgenology* 198(1): 19–26 (2012)

[3] Wang S, Summers RM: Machine learning and radiology. *Medical Image Analysis* 16(5): 933–951 (2012)

[4] El-Naqa I, Yang YY, Wernick MN, Galatsanos NP, Nishikawa RM: A support vector machine approach for detection of microcalcifications. *IEEE Transactions on Medical Imaging* 21: 1552–1563 (2002)

[5] Davatzikos C, Fan Y, Wu X, Shen D, Resnick SM: Detection of prodromal alzheimer's disease via pattern classification of magnetic resonance imaging. *Neurobiology of aging* 29(4): 514–523 (2008)

[6] Klöppel S, Stonnington CM, Chu C, Draganski B, Scahill RI, Rohrer JD, Fox NC, Jr CRJ, Ashburner J, Frackowiak RSJ: Automatic classification of mr scans in alzheimer's disease. *Brain* 131(3): 681–689 (2008)

[7] Zacharaki EI, Wang S, Chawla S, Yoo DS, Wolf R, Melhem ER, Davatzikos C: Classification of brain tumor type and grade using mri texture and shape in a machine learning scheme. *Magnetic Resonance in Medicine* 62: 1609–1618 (2009)

[8] Emblem KE, Zoellner FG, Tennoe B, Nedregaard B, Nome T, Due-Tonnessen P, Hald JK, Schele D, Bjornerud A: Predictive modeling in glioma grading from mr perfusion images using support vector machines. *Magnetic Resonance in Medicine* 60: 945–952 (2008)

[9] nd Vinod Kumar JS, Gupta I, Khandelwal N, kamal Ahuja C: A dual neural network ensemble approach for multiclass brain tumor classification. *International Journal for Numerical Methods in Biomedical Engineering* 28: 1107–1120 (2012)

[10] Bendfeldt K, Klöppel S, Nichols TE, Smieskova R, Kuster P, Traud S, adn Yvonne Naegelin NML, Kappos L, Radue EW, Borgwardt SJ: Multivariate pattern classification of gray matter pathology in mulitple sclerosis. *NeuroImage* 60: 400–408 (2012)

[11] Dolecek TA, Jennifer M Propp NES, Kruchko C: CBTRUS statistical report: Primary brain and central nervous system tumors diagnosed in the united states in 2005-2009. *Neuro-Oncology* 14(5): v1–v29 (2012)

[12] Zielonke N: *Krebsinzidenz und Krebsmortalität in Österreich.* Statistik Austria, Wien (2012). Available at `http://www.statistik.at/web_de/statistiken/gesundheit/krebserkrankungen/`

[13] Grunwald I, Dillmann K, Roth C, Backens M, Reith W: Supratentorielle Tumoren. *Radiologe* 47: 471–485 (2007)

[14] Schneider T, Mawrin C, Scherlach C, Skalej M, Firsching R: Gliomas in adults. *Deutsches Ärzteblatt International* 107(45): 799–808 (2010)

[15] Stupp R, Mason WP, van den Bent MJ, et al.: Radiotherapy plus concomitant and adjuvant temozolomide for glioblastoma. *The New England journal of medicine* 352(10): 987–96 (2005)

[16] Payer F: Pseudoprogression oder Pseudorespons: Herausforderung an die Bildgebung des Glioblastoma multiforme. *Wiener Medizinische Wochenschrift* 13–19 (2011)

[17] Hygino da Cruz LC, Rodriguez I, Domingues RC, Gasparetto EL, Sorensen AG: Pseudoprogression and pseudoresponse: imaging challenges in the assessment of post-treatment glioma. *American journal of neuroradiology* 32(11): 1978–85 (2011)

[18] Gribbestad IS, Gjesdal KI, Nilsen G, Lundgren S, Hjelstuen MH, Jackson A: An introduction to dynamic contrast-enhanced MRI in oncology. In: A Jackson, DL Buckley, GJM Parker (eds.): *Dynamic Contrast-Enhanced Magnetic Resonance Imaging in Oncology.* Berlin Heidelberg New York, Springer: – (2005)

[19] Macdonald DR, Cascino TL, Schold SC, Cairncross JG: Response criteria for phase II studies of supratentorial malignant glioma. *Journal of Clinical Oncology* 8(7): 1277–1280 (1990)

[20] Brandsma D, Stalpers L, Taal W, Sminia P, van den Bent MJ: Clinical features, mechanisms, and management of pseudoprogression in malignant gliomas. *The Lancet Oncology* 9(5): 453–461 (2008)

[21] Siu A, Wind JJ, Iorgulescu JB, Chan TA, Yamada Y, Sherman J: Radiation necrosis following treatment of high grade glioma - a review of the literature and current understanding. *Acta neurochirurgica* 154(2): 191–201 (2012)

[22] Chamberlain MC, Glantz MJ, Chalmers L, Horn AV, Sloan AE: Early necrosis following concurrent temodar and radiotherapy in patients with glioblastoma. *Journal of neuro-oncology* 82(1): 81–83 (2007)

[23] Pedersen M, van Gelderen P, Moonen CT: Imaging techniques for dynamic susceptibility contrast-enhanced MRI. In: A Jackson, DL Buckley, GJM Parker (eds.): *Dynamic Contrast-Enhanced Magnetic Resonance Imaging in Oncology.* Berlin Heidelberg New York, Springer: – (2005)

[24] Wu O, Østergaard L, Koroshetz WJ, Schwamm LH, O'Donnell J, Schaefer PW, Rosen BR, Weisskoff RM, Sorensen AG: Effects of tracer arrival time on flow estimates in MR perfusion-weighted imaging. *Magnetic Resonance in Medicine* 50: 856–864 (2003)

[25] Keston P, Murray AD, Jackson A: Cerebral perfusion imaging using contrast-enhanced MRI. *Clinical Radiology* 58: 505–513 (2003)

[26] Provenzale JM, Mukundan S, Barboriak DP: Diffusion-weighted and perfusion MR imaging for brain tumor characterization and assessment of treatment response. *Radiology* 239(3): 632–649 (2006)

[27] Sugahara T, Korogi Y, Tomiguchi S, Shigematsu Y, Ikushima I, Kira T, Liang L, Ushio Y, Takahashi M: Posttherapeutic intraaxial brain tumor: the value of perfusion-sensitive contrast-enhanced MR imaging for differentiating tumor recurrence from nonneoplastic contrast-enhancing tissue. *American Journal of Neuroradiology* 21: 901–909 (2000)

[28] Hu LS, Baxter LC, Smithd KA, Feuersteine BG, Karisb JP, Eschbacherf JM, Coonsf SW, Nakajid P, Yehh RF, Debbinsg J, Heisermanb JE: Relative cerebral blood volume values to differentiate high-grade glioma recurrence from posttreatment radiation effect: Direct correlation between image-guided tissue histopathology and localized dynamic susceptibility-weighted contrast-enhanced perfusion MR imaging measurements. *American Journal of Neuroradiology* 30: 552–558 (2009)

[29] Basser PJ, Özarslan E: Introduction to diffusion MR. In: H Johansen-Berg, TEJ Behrens (eds.): *Diffusion MRI - from quantitative measurement to in-vivo neuroanatomy.* Amsterdam, Academic Press: 3–10 (2009)

[30] Caroline I, Rosenthal MA: Imaging modalities in high-grade gliomas: Pseudoprogression, recurrence, or necrosis? *Journal of clinical neuroscience* 19(5): 633–637 (2012)

[31] Nilsson NJ: *Introduction to Machine Learning: An early dratft of a proposed textbook.* Stanford, Stanford University (1998). Available at `http://robotics.stanford.edu/~nilsson/mlbook.html`

[32] Hastie T, Tibshirani R, Friedman J: *The Elements of Statistical Learning.* Springer, 2nd ed. (2009)

[33] Witten IH, Frank E: *Data Mining - Practical Machine Learning Tools and Techniques.* San Francisco, Elsevier, 2nd ed. (2005)

[34] Runkler TA: *Data Mining - Methoden und Algorithmen intelligenter Datenanalyse.* Wiesbaden, Vieweg+Teubner, 1st ed. (2010)

[35] Alpaydin E: *Introduction to Machine Learning.* Cambridge, Massachusetts, The MIT Press, 2nd ed. (2010)

[36] Duda RO, Hart PE, Stork DG: *Pattern Classification.* New York, John Wiley & Sons, Inc., 2nd ed. (2001)

[37] Schölkopf B, Smola AJ: *Learning with Kernels.* Cambridge, Massachusetts, The MIT Press (2002)

[38] Smola A, Vishwanathan SVN: *Introduction to Machine Learning.* Cambridge, UK, Cambridge University Press (2008)

[39] Schölkopf B, Smola AJ, Williamson RC, Bartlett PL: New support vector algorithms. *Neural Computation* 12: 1207–1245 (2000)

[40] Tarassenko L, Hayton P, Cerneaz N, Brady M: Novelty detection for the identification of masses in mammograms. In: *Proceedings Fourth IEE International Conference on Artificial Neural Networks.* IEE (1995), vol. 4, 442–447

[41] Schölkokpf B, Williamson R, Smola A, Shawe-Taylor J, Platt J: Support vector method for novelty detection. In: SA Solla, TK Leen, KR Müller (eds.): *Advances in Neural Information Processing Systems 12.* Cambridge, MA, MIT Press (2000), 582–588

[42] Campbell C, Ying Y: *Learning with Support Vector Machines.* Morgan & Claypool (2011)

[43] Bishop CM: *Pattern recognition and machine learning.* New York, Springer (2006)

[44] Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC: Estimating the support of a high-dimensional distribution. *Neural Computation* 13: 1443–1471 (2001)

[45] Hsu CW, Chang CC, Lin CJ: *A Practical Guide to Support Vector Classification.* Department of Computer Science, National Taiwan University, Taipei 106, Taiwan (2010). `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`, downloaded in May 2013

[46] Ashburner J, Barnes G, et al: *SPM8 Manual.* Institute of Neurology, UCL, London, UK (2012). Software available at `http://www.fil.ion.ucl.ac.uk/spm/`

[47] Collignon A, Maes F, Delaere D, Vandermeulen D, Suetens P, Marchal G: Automated multi-modality image registration based on information theory. *Information Processing in Medical Imaging* 263–274 (1995)

[48] Chang CC, Lin CJ: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2: 27 (2011). Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[49] Riffenburgh RH: *Statistics in Medicine.* London, UK, Academic Press, Elsevier, 3rd ed. (2012)

[50] Chapelle O, Haffner P, Vapnik VN: Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks* 10(5): 1055–1064 (1999)

[51] Rorden C: *MRIcron software package.* (Version 12 12 2012), `http://www.mccauslandcenter.sc.edu/mricro/mricron`

7 Appendix

Attached to this thesis a package is provided, that includes the code of all scripts, functions and all software packages that were used for this study. To run the scripts and functions MATLAB® is required. The following MATLAB toolboxes are needed: Statistics Toolbox and Bioinformatics Toolbox. The MRIcron converting tool dcm2nii, the spm8 package and the LIBSVM package for MATLAB® are provided with the attached package.
The script *run_all* runs all functions and scripts in the order as they were intended to be run. Table 7.2 lists all functions and scripts that are included in the package. It also lists the dependencies of the functions to each other and which software packages and MATLAB toolboxes are required to run those functions.
In the following the most important functions and scripts and how the data should be provided to the functions are described.

7.1 Converting DICOM Files

The DICOM files were converted to NIfTI using the converting tool dcm2nii from MRIcron, which is part of the MRIcron software package [51]. The MATLAB function *convertIMA2NII*(`input_dir`, `output_dir`) creates batch-files to run dcm2nii for converting all DICOM files in a given directory, providing that the files are organized in a defined manner.
All DICOM files have to end on the extension '`ima`'. The DICOM files of one series have to be in one folder with a name that describes the series. All series folders that correspond to one patient or examination have to be in one folder with a folder name that is unique and is of the form '`Patient_`*ID*', where *ID* is a unique identifier, consisting of numbers or letters but no special characters.
`input_dir` is the directory containing all those patient folders. In the directory `output_dir` new patient folders with the same names as in the input directory are created. All converted NIfTI images of one patient/examination are written to the corresponding patient folder in the output directory. The NIfTI files get the names '`Patient_`*ID_seriesfoldername*', where *seriesfoldername* is the name of the folder where the DICOM files of the series were stored.
VOI-files (MRIcron format that defines volumes of interest, has the extension `voi`) that

Table 7.1: The here listed 10 series have to be in the folder of each patient. The corresponding NIfTI files of the series are identified by their file names. The below listed strings must be or must not be in the name, in oder that the files are identified correctly.

	Series Description	must be in name	must not be in name
1	contrast-enhanced T1WI	`T1`	
2	T2WI	`T2_TSE`	
3	FLAIR	`T2_TIRM`	
4	CBV-map	`CBV`	
5	CBF-map	`CBF`	
6	MTT-map	`MTT`	
7	TTP-map	`TTP`	
8	ADC-map	`ADC`	
9	1st time point of perfusion imaging	`PERF`	`CBV`, `CBF`, `MTT` or `TTP`
10	1st image of diffusion imaging	`DIFF` or `TRACE`	`ADC`

were saved with the DICOM files are copied to the corresponding patient folder as well. While converting the data from DICOM to NIfTI the data is anonymized.

7.2 Registration

The MATLAB function *registerEstimate*(`input_dir`) does the registration estimation using the spm8 package. The registration is done in the given input directory `input_dir`. `input_dir` should contain folders that each correspond to one patient/examination like they are created by the function *convertIMA2NII*. For each folder it does the registration of the NIfTI images that are stored in it.

The function searches for a NIfTI-file, which name starts with `Patient` and contains the string `T1`. This file is set to be the reference file in the registration process. Furthermore in table 7.1 all series are listed that should be stored in the folder as NIfTI files. The series are identified by the name of the NIfTI files. Table 7.1 also lists the strings that should be part of the file name such that the NIfTI files can be assigned to the corresponding series. All relevant NIfTI files have to start with the string `Patient`. The function *registerEstimate* modifies the transformation matrices of the NIfTI files such that they are all registered to the T1-weighted image.

7.3 Reslicing and Resampling

The function *resliceResample*(`input_dir`, `interpolation`) does the reslicing and resampling according to the transformation matrices stored in the NIfTI headers. `input_dir` is the input directory in which the folders for each patient/examination are located, which

contain the NIfTI files that are to be resliced and resampled. In each folder it searches for the NIfTI file which name starts with `Patient` and contains the string `CBV`. It takes this file as reference image, which defines the space the other images are resampled to.
Furthermore the function converts all VOI-files that are located in the folder to NIfTI format (*.nii) using the function *mricronVoi2Nii*. Subsequently all NIfTI files whose names start with the string `Patient` are resliced and resampled to the space of the reference image using the spm8 package [46]. Excluded from this resampling process are the first time point of the perfusion imaging (containing the string `PERF` but not `CBV`, `CBF`, `MTT` or `TTP` in its name) and the first image of the diffusion imaging (containing the strings `DIFF` or `TRACE` but not `ADC` in its name). Note that all images, including the VOI-files, should be aligned to the T1-weighted image in order that the resampling shows meaningful results, like it is done by the function *registerEstimate*.

The resliced and resampled images are written to new files in the same folder. A prefix 'r' is added at the beginning of the name.

`interpolation` defines the interpolation method by which the images are sampled. When `interpolation` is set to `0` nearest neighbor interpolation is used. `1` is set for trilinear interpolation and `2` is set for 2nd degree b-spline interpolation.

In the end all images that are not resliced and resampled can be deleted with the function *clearNoneResampled*(`input_dir`). `input_dir` is the same directory as used in the function *reslice*. It deletes all NIfTI files that start with the string `Patient`, without the prefix `r`, except for the file containing the string `CBV` in its name. Since the reference image was not resliced it does not have the prefix `r` it its name.

7.4 Loading the Data into MATLAB®

The function `data` = *loadData*(`input_dir`) loads the data of the images within the ROI into MATLAB. `input_dir` is again the directory containing folders that each correspond to one patient/examination and contain all relevant NIfTI files.

The function masks the images with the masks stored in the ROI defining NIfTI files, that were converted from the VOI-files and resampled in the previous step by the function *resliceResample*. The name of those ROI defining NIfTI files should be of the form `r*_voi.nii`, where `*` denotes an arbitrary string (The string `_voi` is added to the original name by the function *mricronVoi2Nii* when converting a VOI-file to NIfTI.). There should be exactly two of such files and one of them should contain additionally the string `nawm`. The mask of this file is assumed to mark the ROI in the normal appearing white matter. The other file is thought to mark the contrast-enhanced lesion.

Again, it is important that the file names contain the strings as described in table 7.1. Additionally the name of the CBV-file has to start with the string `Patient` and all other relevant files have to start with the string `rPatient`. That way the files should be uniquely identifiable.

The output `data` is a cell array. Each cell of the array contains the data of one patient/examination. It contains an $N \times 8$ matrix, where N is the number of voxels within the ROI marking the contrast-enhanced lesion and each of the 8 columns corresponds to one image/series. The columns correspond to the series 1-8 listed in table 7.1 and are organized in the same order as in table 7.1. The values of the ouput variable `data` are normalized by the mean of the intensities within the nawm-ROI.

7.5 Train and Test the OC-SVM

The MATLAB script *train_and_test_oc_svm* trains and tests the OC-SVM. The training is done using the LIBSVM-package [48]. First the data has to be loaded with the function *loadData*. The data of the group on which the OC-SVM is trained has to be loaded to the variable `data_a` and the data of the other group has to be loaded to the variable `data_b`. Several parameters (σ, ν, etc.) are set within in the script. For further information look at the script.

The script writes the averaged perfomance measures obtain by cross-validation to a csv-file. The columns of the csv-file correspond to the following parameters and performance measures: σ, γ, ν, c, voxel-wise sensitivity, voxel-wise specificity, voxel-wise $SN \cdot SP$, p-value.

The classification results of the last run, that is the last parameter set and the last repeat, can be observed in the variable `performance_measures`. In columns 1-3 of the matrix `performance_measures` the voxelwise accuracy, sensitivity and specificity of the single patients are stored. The fourth column stores the fraction of the as positives classified voxels within the ROI of each patient.

Table 7.2: MATLAB files that are included in the provided software package and dependencies to each other.

MATLAB Files	called functions / required packages
clearNoneResampled	
convertIMA2NII	dcm2nii (MRIcron software package)
crossValidation	getConfusionMat myClassifyLibSVM
getConfusionMat	
getMasks	spm8 software package
loadData	getMasks loadMaskedData normalization sortSequencesByNames
loadMaskedData	spm8 package
mricronVoi2Nii	
myClassifyLibSVM	Statistics toolbox normMatInDim
normMatInDim	
normalization	
registerEstimate	spm8 software package registerEstimateCreateJobfile
registerEstimateCreateJobfile	
resliceResample	spm8 software package reslicingCreateJobfile mricronVoi2Nii
reslicingCreateJobfile	
run_all	convertIMA2NII registerEstimate resliceResample clearNoneResampled loadData train_and_test_oc_svm
sampleTrainingSet	Bioinformatics toolbox Statistics toolbox
scaleFeatures	Statistics toolbox
sortSequencesByNames	
trainOcSvm	LIBSVM package (Version 3.17)
train_and_test_oc_svm	Statistics toolbox
	scaleFeatures sampleTrainingSet trainOcSvm crossValidation